IT Agility through Service-Oriented Architecture

Author:
Sadr Dadras, Arash

Publication Date:
2017

DOI:
https://doi.org/10.26190/unsworks/3302

IT Agility through Service-Oriented Architecture

by

Arash Sadr Dadras

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Information Systems & Technology Management
Australian Business School

Supervisors:

Dr. Farhad Daneshgar

Prof. Walter Fernández

August 2016

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Sadr Dadras

First name: Arash

Other name/s:

Abbreviation for degree as given in the University calendar: PhD

School: Information System and Technology Management

Faculty: Business school

Title: IT Agility through Service-Oriented Architecture

Abstract 350 words maximum:

Firms recognise agility as a crucial capability to survive and thrive in uncertain and turbulent markets. One of the reported enablers of agility, from an IT perspective, is the Service-Oriented Architecture (SOA). SOA with reliance on its characteristics has provided faster time to market and reduced system complexity. Review of SOA empirical results, however, suggested an increased process and system complexities which conversely inhibit achieving the goal of agility. The contradictory effect of SOA on complexity and lack of insight on IT sensing capability motivated this research to study the effects of SOA on the two-underlying sensing and responding capabilities of the IT agility. Due to the current limited theoretical perspectives in the extant SOA studies, the current study undertakes a theory-building research. With reliance on the real options theory, complemented by the dynamic capabilities, the current study develops an initial theoretical framework and argues that SOA characteristics, when embedded in a system at the design time, will provide future knowledge and process options. The embedded future options, when executed, will facilitate the IT sensing and responding capabilities. By collecting and analysing data from twenty-two in-depth interviews as well as project documents across multiple cases in a Bank, an Airline company, and an Airport, this study extends its initial conceptual framework to a mid-range theory that explains the interaction between SOA and IT agility. Results of the study contribute to the SOA literature by conceptualising the SOA in three sets of characteristics including: ‘information-centric’, flexibility-centric’ and ‘structure-centric’, as well as theorising the effects of each set on the effectiveness of process options and ‘change detection’ and ‘shared insight’ knowledge options. Furthermore, the study took steps to extend the options theory by developing a new concept called options depreciators, which represents the negative effect of options when there are multiple competing options. Finally, this dissertation contributes to the SOA governance literature by providing a novel view on the required governance structure for achieving agility. Under this perspective, this study presents an adaptive governance structure that is consistent with the SOA hierarchical layering and also appropriate to foster business autonomy and to create sustainable assets for future use.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed ....

Date ....

COPYRIGHT STATEMENT

‘I hereby grant the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral theses only). I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.'

Signed ....

Date ....

AUTHENTICITY STATEMENT

‘I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.’

Signed ....

Date ....

To

My parents Morteza and Esmat,

My wife Gelareh and

My sons Avesta and Ahoora

Acknowledgements

What a life changing journey! After eight years of part-time PhD, it feels great to reach this point: writing the very last page! And reaching this point was only possible with help and support from a number of special individuals.

To my supervisor, Dr Farhad Daneshgar, who was always available to assist me during the course of my research. I am very grateful that you accepted my supervision. Your positive and supportive attitude helped me to remain on track and just work harder.

To my co-supervisor, Prof Walter Fernandez. Although you joined my supervision team just a few months ago, your advice and insights were invaluable. Thank you so much for your help and support.

The feedback from my review committee members and the fellow PhD students at the school were fantastic, and I am grateful for their contribution to my research.

And taking this journey was only possible with motivation from my parents, particularly my mum's regular questioning on how I was progressing with my research! I love you both, and I hope I can pass on the love and the attention you gave me to my children.

Obviously, such a long-term commitment was not possible without a total support, patience and encouragement from my life companion and love, my wife, Gelareh. We had to sacrifice a lot of our moments to help me realise this dream; you embraced the challenge, and supported me. Thank you for your endless love and support. I could not have done it without you.

Abstract

Firms recognise agility as a crucial capability to survive and thrive in uncertain and turbulent markets. One of the reported enablers of agility, from an IT perspective, is the Service-Oriented Architecture (SOA). SOA with reliance on its characteristics has provided faster time to market and reduced system complexity. Review of SOA empirical results, however, suggested an increased process and system complexities which conversely inhibit achieving the goal of agility. The contradictory effect of SOA on complexity and lack of insight on IT sensing capability motivated this research to study the effects of SOA on the two-underlying sensing and responding capabilities of the IT agility. Due to the current limited theoretical perspectives in the extant SOA studies, the current study undertakes a theory-building research. With reliance on the real options theory, complemented by the dynamic capabilities, the current study develops an initial theoretical framework and argues that SOA characteristics, when embedded in a system at the design time, will provide future knowledge and process options. The embedded future options, when executed, will facilitate the IT sensing and responding capabilities. By collecting and analysing data from twenty-two in-depth interviews as well as project documents across multiple cases in a Bank, an Airline company, and an Airport, this study extends its initial conceptual framework to a mid-range theory that explains the interaction between SOA and IT agility. Results of the study contribute to the SOA literature by conceptualising the SOA in three sets of characteristics including: 'information-centric', flexibility-centric' and 'structure-centric', as well as theorising the effects of each set on the effectiveness of process options and 'change detection' and 'shared insight' knowledge options. Furthermore, the study took steps to extend the options theory by developing a new concept called options depreciators, which represents the negative effect of options when there are multiple competing options. Finally, this dissertation contributes to the SOA governance literature by providing a novel view on the required governance structure for achieving agility. Under this perspective, this study presents an adaptive governance structure that is consistent with the SOA hierarchical layering and also appropriate to foster business autonomy and to create sustainable assets for future use.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Background to the research

The competitive environment of contemporary business has become more intensive with a higher rate of change. Organisations that possess the capability to respond to environmental dynamics, known as business agility, are likely to produce better outcomes (Sambamurthy, Bharadwaj, & Grover, 2003). Business agility enables firms to sense events such as competitive rivalry and customer demand shifts and respond quickly by performing competitive actions (Dove, 2001; Sambamurthy et al., 2003).

With the heavy reliance of organisations on information systems, the agility of IT, which is a collection of IT resources and processes in an organisation, has been recognised as an important enabler of business agility (Fink & Neumann, 2007; Yousif & Pessi, 2016). Such capability, known as IT agility, provides the IT with the ability to swiftly detect and implement changes in their information systems to support the business to survive and thrive in an uncertain environment (Lui & Piccoli, 2007; Tiwana & Konsynski, 2010). Agility of the IT improves the business agility (Fink & Neumann, 2007) which in turn can lead to higher competitive actions and higher organisation performance (Overby, Bharadwaj, & Sambamurthy, 2006; Roberts & Grover, 2012; Sambamurthy et al., 2003).

The extant agility literature provides a mixed view on the role of information systems in enabling agility. On the negative front, factors such as rigid IT architecture (Oosterhout, Waarts, & Hillegersberg, 2006; Rettig, 2007), inflexibility (Allen & Boynton, 1991; Oosterhout et al., 2006; Rettig, 2007) and complexity of information systems (Oosterhout et al., 2006; Rettig, 2007) have positioned the information systems as agility inhibitors.

Service-Oriented Architecture (SOA), as an alternative style of designing information systems, has promised to overcome many of the system constraints such as inflexibility and complexity of information systems. SOA, relying on its characteristics such as standardisation, interoperability and modularity, has provided further IT flexibility, faster time to market and reduced complexity (Abelein, Habryn, & Becker, 2009; Baskerville et al., 2005; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c), which resulted in agility being suggested as the main SOA business benefit in the SOA literature (Hagel & Brown, 2001; Huang & Hu, 2004; J. Schelp & Aier, 2009; Yoon & Carter, 2007). Contrary to the above results, some existing SOA field studies have also indicated that SOA increases the system complexity, which is an agility inhibitor (Baskerville et al., 2005; Kokko, Antikainen, & Systa, 2009; J. Schelp & Aier, 2009).

Considering the conflicting reports concerning the SOA impact on agility and lack of theorisation to explain how SOA creates agility, the current research investigates the effect of SOA on IT agility as a novel contribution to the SOA and agility literature.

1.2 Research Objectives and Questions

As highlighted above, the objective of this research is to study the impact of *SOA as a style of architecture on IT agility* and to build a theoretical explanation of the SOA and IT agility interaction to uncover the complexity paradox reported in the SOA literature. This objective is consistent with studies that have assessed the business value of IT (G.D. Bhatt & Grover, 2005; Kohli & Grover, 2008; Nevo & Wade, 2010; Rahrovani & Pinsonneault, 2012; Tallon, 2007). The studies of IT and its business values have typically explained how IT resources including IT assets and capabilities can create business value in the form of improved organisation performance, competitive advantage or operational efficiency and effectiveness.

The current research investigates how SOA as an IT intangible asset (Feeny & Willcocks, 1998; Ross, Beath, & Goodhue, 1996) impacts IT agility. The above enquiry can be encapsulated in the following main research question for the current study:

Main RQ: How does SOA as an intangible asset impact the IT agility?

To explore the question above, the SOA and IT agility concepts need to be further developed.

On the conceptualisation of SOA, the current study considers SOA as an intangible IT asset (Ross et al., 1996). As a style of architecture (Sprott, 2004), SOA is concerned with rules governing the overarching structure and properties of systems and their relationships (Tiwana & Konsynski, 2010). Such rules are presented through characteristics such as system scalability, connectivity and system modularity, which are *embedded* in a system upfront during the system design (Tiwana & Konsynski, 2010). On this basis, the current study presents the SOA through its characteristics.

Turning from the SOA to the IT agility, the current research follows the scholars (Overby et al., 2006; Sambamurthy et al., 2003; Teece, Pisano, & Shuen, 1997) who consider agility as a dynamic capability in the organisation. Dynamic capability is the capacity of an organisation to purposefully create, extend, or modify its resource base (Helfat et al., 2007). Such capabilities enable the firm to address rapidly changing environments to achieve competitive advantage (Teece et al., 1997). The extant agility literature (Overby et al., 2006; Roberts & Grover, 2012) has recognised *IT sensing* and *responding* capabilities as two underlying capabilities of the IT agility. The current empirical studies reveal that agile organisations have both strong sensing capability to detect opportunities and threats, as well as strong responding capability to seize the opportunity promptly (Overby et al., 2006; Roberts & Grover, 2012).

Finally, focussing on 'how' in the above research question brings the attention to the mechanisms involved in the interaction between SOA and IT agility, that in turn, can shed light on the conflicting reports concerning the SOA complexity (Baskerville et al., 2005; Kokko et al., 2009; J. Schelp & Aier, 2009).

Given the two underlying components of IT agility and the above SOA conceptualisation, the first step in achieving the current research objective is to uncover the SOA characteristics that affect the IT sensing and IT responding capabilities. Therefore, two further sub-questions for the current study are developed:

> RQ1: What are the SOA characteristics that affect the IT sensing capability?
> RQ2: What are the SOA characteristics that affect the IT responding capability?

Answers to the above questions are covered in the future chapters. In short, this research identifies three themes of SOA characteristics that when they are embedded in the information systems they will impact the IT sensing and responding capabilities. The SOA information-centric characteristics contribute to the IT sensing capability, and the SOA flexibility-centric and SOA structure-centric characteristics are the main contributor to IT responding capability.

The second step in the current research is to uncover the interactions between the embedded SOA characteristics and the IT sensing and responding capabilities. The links between the embedded SOA characteristics and IT agility will provide further insight on any interplay between the factors that impact on IT agility. On this basis, the following two additional research questions are developed:

> RQ3: How do the SOA embedded characteristics facilitate the sensing component

of IT agility?

RQ4: How do the SOA embedded characteristics facilitate the responding component of IT agility?

To answer the above two questions, the current study takes a theory building route (Gregor, 2006) to explain the interactions between SOA and IT agility. The outcome is presented in the form of a set of theoretical propositions and a conceptual model showing interactions between the three identified SOA themes and the IT sensing and IT responding capabilities through several knowledge and process options. Additionally, the study discovers a few non-SOA factors that are relevant to the SOA interaction with the IT agility.

On the theoretical front, the current study adopts the 'real options theory' (Bowman & Hurry, 1993; Copeland, Koller, & Murrin, 1994; Trigeorgis, 1996) and 'dynamic capabilities' (Amit & Schoemaker, 1993; Makadok, 2001; Teece et al., 1997) as the theoretical foundations of the study to better understand and explain the link between SOA and IT agility. The real options, which is based on the financial call option (Amram & Kulatilaka, 1999), considers investments in organisational resources in terms of the investment's ability to generate future strategic choices with the assumption that there is an underlying source of uncertainty (Bowman & Hurry, 1993). Application of real options theory in the current study is inspired by the Sambamurthy et al. (2003)'s concept of 'digital options'. The current study argues that the SOA characteristics, when embedded in a system at the design time, provide future knowledge and process options. The embedded future options, when executed, facilitate the IT sensing and responding capabilities to create, extend or modify the information systems as the organisational resources. Architectural characteristics of SOA such as connectivity and loosely coupling are examples of such characteristics that can be *embedded* in a system upfront to provide future options such as system extension or reuse to system development processes.

The adoption of options theory in the current study is opportunistic. While many studies adopted the options theory as a way of thinking (Fichman, 2004; Fichman, Keil, & Tiwana, 2005), they typically focus on an individual set of options related to one investment. However, the current study extends this view by studying how the options theory can be applied in an environment with multiple investments and sets of options, which, in turn, can be embedded and executed by different parties. The current study evaluates whether there is an interaction between these options with an intention to contribute to the base theory.

To answer the above research questions and to extend the initial base theory in the context of SOA, the current study conducted a post-positivist case study research in a theory building study. By analysing multiple case studies, the study uncovers SOA characteristics that contribute to the IT sensing and responding capabilities and identify the mechanisms involved in the interaction between the SOA characteristics and the IT agility.

In brief, the objective of the current study is to assess the interactions between SOA and IT agility through a theoretical stance. The first two research questions of the study focus on identification of the SOA characteristics that affect the IT sensing and IT responding capabilities. Furthermore, to uncover the relationship between SOA and the IT agility, the last two research questions address how the SOA characteristics and the IT sensing and responding capabilities interact. By adopting the real-options theory and dynamic capabilities as theoretical perspectives, the current study positions the SOA as an option generator, which embed process and knowledge options in the system for future use. Such options enhance the IT sensing and responding capabilities by providing them with the flexibility to create, extend or modify the existing systems.

Not only the current study provides an opportunity to investigate and conceptualise the SOA characteristics that play a role in enhancing IT agility, but it also sheds light on the options theory in the context of multiple investments and sets of options.

1.3 Rationale and Motivation

The current study focuses on the theoretical explanation on how SOA as an intangible asset affects sensing and responding capabilities of the IT. This section provides justifications for the current study on multiple fronts including the significance of agility and SOA, consideration of both sensing and responding capabilities, a gap in SOA theorisation, and focus on architecture in theory building. These are explained below.

Significance of agility and SOA. Due to business uncertainty and unpredictable environment, organisations have high expectation of their IT being able to respond to the required changes in their information systems. The above is in addition to the ongoing challenges that IT currently faces in protecting systems from threats such as cyber-attacks. In such an environment, IT’s ability to sense and respond to these changes and threats becomes a critical enabler of business agility (Overby et al., 2006; Sambamurthy et al., 2003).

On the SOA front, Service-Oriented Architecture (SOA) as an emerged approach in the modern software engineering (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2010; Papazoglou, 2008), has affected different areas of IT within organisations. Areas such as IT infrastructure through the introduction of new middleware and technologies, application development by introducing services, application architecture by introducing distributed and loosely coupled systems and, finally, business process modelling and automation, are examples of SOA impact on the IT landscape (Zimmermann, Krogdahl, & Gee, 2004). Furthermore, the exposure of SOA is not limited to the IT services within an organisation. The application of services can be seen in other emerging concepts such as *microservices* (Balalaie, Heydarnoori, & Jamshidi, 2016) adopted by many organisation such as Google, Amazon and Facebook to expose their services externally. Similarly, the Cloud computing and services heavily rely on the SOA concepts (Hoberg, Wollersheim, & Krcmar, 2012; Stieninger & Nedbal, 2014).

The significance of both IT agility and SOA as underlying concepts applicable in today's IT world motivated the current study. Such motivation became stronger with the mixed view reported in the SOA literature concerning the SOA impact on agility. As found in broader agility literature, business agility can be impacted upon by many non-IT factors such as the business network structure, organisation structure and governance structure (Yang & Liu, 2012; Yusuf, Sarhadi, & Gunasekaran, 1999; Zhang & Sharifi, 2007). It can be argued that since SOA benefits (e.g. faster development and IT flexibility) are mainly IT-related, the assessment of IT agility rather than business agility in the study of SOA can reduce the effect of non-IT factors and provide better insight on how SOA can improve the responsiveness of IT.

Consideration of both sensing and responding capabilities. The agility literature has already shown the importance of sensing capability in achieving agility (Overby et al., 2006; Roberts & Grover, 2012). Despite such important role, the existing SOA studies are silent on this dimension of agility and have not reported how SOA affects the IT sensing capability. By focusing on both dimensions of agility, the current study provides further insight into the SOA interaction with the agility.

Gap in SOA theorisation. While SOA can have a considerable impact on the IT landscape and deliver many benefits, the lack of a theoretical perspective to conceptualise and explain how SOA achieves its benefits has constrained the SOA research. Identification and application of a suitable theory can provide explanation and possibly prediction on how certain outcomes are driven from the SOA characteristics. Such theoretical perspective can also unfold compound constructs such as complexity, which have been reported in the SOA adoption. The importance of the above issue has already been reflected in multiple calls for research thus encouraging further research on conceptualisation of SOA and identification of suitable theories to explain the SOA value creation (Bardhan et al., 2010; Joachim, 2011). By conceptualising SOA and theorising on how SOA enables IT agility, the current study is an initial attempt in addressing the lack of theorisation in the field of SOA. This is done by developing a theoretical foundation to conceptualise SOA and its effects on IT agility.

Focus on architecture in theory building. The last motivation for this study is related to the IT artefact and unpacking this "black box" as suggested by many IS scholars (Orlikowski & Iacono, 2001; Tiwana, Konsynski, & Bush, 2010). Historically, the IT artefact has tended to disappear from views, and has been treated either as a monolithic black box, or has become the "omitted" variable (Orlikowski & Iacono, 2001). The review of SOA as a style of architecture offers an opportunity to bring IT artefact into the core of the theory development and to contribute unique insights distinctive from strategy, economics, and software engineering. These issues are particularly germane to IS because understanding how systems and platforms evolve without considering their technical design attributes and relying solely on non-IS perspectives can mislead one into overlooking the important interactions of the IT artefact with its internal and external environment. Systems do not exist in a vacuum, and how well or how poorly they respond to the dynamics of their environment can be influenced by platform designers' technical choices (Tiwana et al., 2010).

Although the importance of architecture is acknowledged in practice, very little attention has been paid to incorporating it into the IS theory development. The use of architecture and its underlying characteristics in theory building provides the opportunity to conceptualise the IT artefact at a level lower than what is typically considered in IS, especially in studies that rely on the resource-based view (Barua, Konana, Whinston, & Yin, 2004; Feeny & Willcocks, 1998; Ross et al., 1996; Wade & Hulland, 2004). The use of architecture and its characteristics provide the ability to deconstruct the IT artefact and study how each factor interacts with its surrounding environment or contributes to the business value. Such deconstruction becomes even more important in the design of artefact such as design processes and methods as well as the adoption of SOA in the organisations to achieve certain objectives.

1.4 Contributions

This research contributes to the academia and practitioners in several ways. On the implications for academia, this study contributes to the service-oriented computing literature, IT agility literature and finally the real-options theory. Similarly, it provides insight to the practitioners on the SOA characteristics that are required to achieve IT agility.

By theorising how SOA affects the IT agility, the current study uncovers the complexity paradox of SOA. The insight gained from the current study is useful for design science researchers to develop and refine design methods for the SOA adoption. In addition, the approach taken to conceptualise and theorise the SOA value could be useful in similar studies of SOA values. Finally, the current research takes additional steps to address the gap between the two IT and SOA literature with respect to how they view systems in their conceptualisation.

The contribution of the current study to the IT agility lies in its inclusion of both sensing and responding capabilities, and how technology can impact these two capabilities. While many studies of the IT agility have focused on the IT responding capabilities, the current study shed light on the IT sensing capability and the impact of SOA enabled knowledge on the IT agility and IT sensing capability.

On the contribution to the academia, the current study takes the initial steps to study the real options theory in a context with multiple investments and options created from such investments. This study explores if and how investments and their options interact and impact one another.

On the practitioners' front, this study provides insight on the characteristics that are essential for the creation of IT agility. Review of SOA from both process and knowledge perspectives clarifies the significance of each SOA characteristic particularly in achieving sensing and responding capabilities.

1.5 Research Boundaries

To demonstrate the scope of the current study, this section introduces a few research boundaries. The first boundary is the consideration of SOA beyond a particular technology implementation and focus on the SOA infrastructure and SOA design characteristics. Additionally, the current study takes a tool view (Orlikowski & Iacono, 2006) of SOA as a technology. The tool view considers that technology as an independent variable can deliver organisational benefits. Finally, the current study assumes that the decision to adopt SOA and its characteristics in a system has already been made and the research focuses on the value (IT agility) that it can deliver.

The first boundary specified above relates to the SOA definition. The SOA literature has applied different terms to the SOA concept:

- Web Services (Currie & Parikh, 2006; Huang & Hu, 2004; Iyer, Freedman, Gaynor, & Wyner, 2003): mainly referring to the interfaces and technologies used in systems,
- Service-Oriented Architecture (SOA): the architectural design principles and guidelines, and
- Service-Oriented Computing (SOC); the software built on the basis of SOA.

Although the focus of each term is different, in some cases, they have been used interchangeably. The focus of this study is on the architectural characteristics of SOA and their impact on IT agility.

Regarding the technology implementations, while the current research considers SOA beyond specific technology implementations, the latter, however, must support the characteristics and criteria outlined by the Service-oriented Architecture and associated services (Legner & Heutschi, 2007). As such, implementations such as CORBA and DCOM are excluded from the scope of the current research due to them not supporting key SOA characteristics such as loose coupling and interoperability (Iyer et al., 2003).

The other boundary consideration relates to the SOA characteristics. This study focuses on the characteristics of SOA as a design approach applied to the information systems and the IT infrastructure. The above assumption excludes the characteristics of processes and governance structures associated with the SOA. Any reference to 'SOA characteristics' in this study conforms to the above assumption unless specified otherwise.

A similar limitation also applies to the study of IT agility. While there are non-technical factors such as those related to process, structure, or people, that independently impact on IT agility, this study focuses on the technical factors, particularly SOA characteristics and their impact on IT agility. This assumption is consistent with the "tool" view of the IT artefact when studying the relationship between SOA and IT agility. The tool view conceptualises SOA as an "engineered artefact, expected to do what its designers intend it to do" (Orlikowski & Iacono, 2006). The tool view considers SOA as a resource that is typically an independent variable in the studies adopting this view and reviews how they affect, alter or transform certain dependent variables. This view typically considers the (human or organisation) entity that is using the tool to benefit from labour substitution, increase performance, enhance information-processing capabilities, and shifting social relationships (Orlikowski & Iacono, 2006). SOA, in this case, works as a magnifier or driver which strongly determines the behaviour of individuals and organisations. As a magnifier, SOA can amplify the existing capabilities or behaviour, for example, as an information-processing tool, it can improve the information flow and improve the learning in the organisation (Markus & Robey, 1988).

Since this study conceptualises SOA as a technology and attempts to identify different characteristics that are involved in the creation of IT agility, the tool view is an appropriate view to unpack the black box of SOA as a technology. It also recognises that technology and its output is impacted by other non-technical factors such as socio-economic activities as flagged by Markus and Robey (1988) and other scholars.

The final boundary on the scope of the current research is related to the decision to embed or adopt SOA characteristics in a system. The real options theory relies on two processes of embedding and exercising options (Tiwana & Konsynski, 2010). The embedding process mainly focuses on the decisions to create an option through investments for future use, for example, decision to make the system modular. The study of decisions to embed options and the factors that impact on such decisions lends itself to the IT adoption research stream, similar to (Fichman, 2004). Considering that the current research's focuses on the value of SOA and how SOA contributes to IT agility, the current study assumes the decision to embed the options has already been taken and that the organisation utilises the embedded option, when applicable.

This assumption positions the current study within the process of exercising the options by assessing the value of these options when embedded and exercised. This is consistent with the stream of studies that review the business value of IT, such as the study of the firm-level benefits of IT-enabled resources (Nevo & Wade, 2011); the study of the business value of IT from the slack resource perspective (Rahrovani & Pinsonneault, 2012); and other similar studies assessing the business value of IT (A. Bharadwaj, Keil, & Mahring, 2009; Cao, 2010; Kohli & Grover, 2008; Ravichandran, Liu, Han, & Hasan, 2009). The study of SOA value through the identification of SOA characteristics that enhance IT agility provides input to the future decision on when and how the options must be embedded and exercised.

1.6 Definitions and Abbreviations

The below table defines some of the terms used in the current study, which require further clarification.

Table 1-1 - Definition of Terms

Term	Definition
IT (Information Techbnology)	The collection of IT resources and processes in an organisation. The IT consists of IT assets and capabilities such as IT infrastructure, human IT resources comprising IT technical and managerial skills, IT-enabled intangible resources such as knowledge assets (A. S. Bharadwaj, 2000) and, finally, IT processes such as planning, operations, support and delivery (Ross et al., 1996). The IT can be centralised in an IT organisational unit or decentralised in different business organisational units.
Dynamic capability	The capacity of an organisation to purposefully create, extend, or modify its resource base" (Helfat et al., 2007)
Architecture	Rules governing the overarching structure and properties of systems and their relationships (Tiwana & Konsynski, 2010).
Information system	Any organised combination of people, hardware, software, communications networks and data resources that collects, transforms and disseminates information in an organisation (J. A. O'Brien & Marakas, 2005).
Resources	Tangible and intangible assets and capabilities available to a firm that are useful in detecting and responding to market opportunities or threats (Sanchez, 1996). Tangible IT assets include physical IT infrastructure and human IT resources and Intangible assets include knowledge assets, customer orientation and synergy (A. S. Bharadwaj, 2000).

Capability	Firm's ability to deploy resources, usually in combination, using organisational processes, to effect a desired end (Amit & Schoemaker, 1993).

1.7 Proposal Outline

This thesis consists of six main chapters. Chapter 2 describes the IT agility and SOA literature with the intention to identify the research gaps and the underlying theories to address removal of the research gaps. This chapter concludes with a section summarises the findings and outlining the gap in the literature.

Using the findings from the literature, Chapter 3 describes the theoretical view that will guide the current research. This chapter proposes a high-level conceptual framework developed from the literature with relevant theoretical propositions. This framework will be used in the current research to create refined research models specific to SOA and IT agility across each of the sensing and responding capabilities.

Chapter 4 of the thesis focuses on the research design, and explains the planned empirical study. It explains the steps involved in building a new mid-range theory that can explain how SOA characteristics can influence IT agility.

Chapter 5 then analyses the data collected from the cases, which results in an integrated conceptual model. The last chapter, Chapter 6, then summarises the findings of the research and compares them to the extant literature. Chapter 6 also expands on the research contributions and future agenda for research.

There are a few appendices at the end of the thesis. A list of candidate measures sourced from the IT infrastructure and business value literature for the SOA characteristics is presented in Appendix A, B and C and the interview protocol developed for the current study is presented in Appendix D.

Chapter 2. Literature Review

2.1 Introduction

Chapter 2 provides a review of the literature concerning the impact of SOA on IT agility. The literature review has five objectives: (1) assessment of the existing knowledge and identification of a knowledge gap in respect to the SOA impact on IT agility; (2) further identification of boundaries for the current study; (3) adoption of appropriate definitions for IT agility and SOA; (4) conceptualisation of IT agility and SOA, when available in the literature; and (5) review and synthesis of the main theories that can explain the relationships between SOA and IT agility.

While the first and the second objectives set directions for this research, the other objectives support the theorisation and development of a conceptual framework in the next chapter.

This chapter has been structured to fulfil the above objectives through a review of the literature, particularly the agility and the SOA literature.

Section 2.2 reviews the IT agility concept. To better understand IT agility, section 2.2.1 first provides a background on agility by reviewing the business agility concept and the dynamic capabilities concept. This background on the agility concept provides a foundation for section 2.2.2 to review the existing IT agility definitions available in the IT literature and adopt a suitable definition for IT agility in the current study (Objective 3). Section 2.2.3 then reviews how different studies have conceptualised IT agility and how it must be conceptualised as per the adopted definition for IT agility in the current study (Objective 4). The last subsection (section 2.2.4) covers the antecedents of IT agility with the intention being to highlight factors that can impact on IT agility such as inflexibility of IT systems.

Section 2.3 reviews the SOA concept as an alternative design paradigm to overcome the agility inhibitors outlined in section 2.2.4. Section 2.3.1 and 2.3.2 first define the SOA concept and explain how it is different from other design paradigms (Objective 3). After gaining an understanding of SOA, section 2.3.3 discusses the benefits as well as challenges reported for SOA. This section highlights a conflicting report regarding the impact of SOA on system complexity. Complexity has been reported as an agility inhibitor that affects the sustainability of agility. To understand this issue, section 2.3.6 identifies the SOA characteristics from the SOA literature and use them to conceptualise the SOA concept. Conceptualisation of SOA using its characteristics provides a level of granularity required to uncover the complexity issue reported in the SOA literature and better understand the role of each characteristic in the SOA impact on IT agility (Objective 1).

After reviewing the IT agility and SOA concepts, the SOA literature is consulted in section 2.5 to provide a review of studies that have focused on the impact of SOA on IT agility. Since no study could be located that has explicitly reviewed the impact of SOA on IT agility, this section covers a broader scope and reviews studies that have focused on business agility. Using the knowledge created to date, this section highlights areas that require further research (Objective 1).

Section 2.4 focuses on the theories that explain the relationships between SOA characteristics and IT agility. In the next chapter (Chapter 3) the study uses two of the reviewed theories to build a conceptual framework for this study. The proposed framework provides a lens for further investigation of the impact of SOA on IT agility (Objective 5).

The last section, Section 2.5, summarises the literature review and reports the gaps that this research is going to address.

2.2 IT Agility

2.2.1 Background

Agility as a concept originated from the US car manufacturing industry in the early 1990s as it dealt with the lack of industry competitiveness. Agile manufacturing was introduced as a new manufacturing concept to deal with changes in customer requirements in a volatile market (Nagel & Dove, 1991). This concept has since extended to other disciplines such as strategic management (Dove, 2001), supply chain and business networks (Yang & Liu, 2012) and, more recently, IT (Overby et al., 2006; Roberts & Grover, 2012; Sambamurthy et al., 2003; Tseng & Lin, 2011) and appears in the literature with terms such as enterprise agility, business agility and organisational agility.

Due to uncertainty and unpredictability caused by factors such as globalization, technology innovations and outsourcing, the ability of an organisation to adapt to unexpected changes is critical so it can achieve and maintain its competitive advantage. This has positioned agility as a crucial factor for a firm to survive and thrive in an uncertain and turbulent market (Ganguly, Nilchiani, & Farr, 2009). In addition to survival in an uncertain environment, agility plays a crucial role for firms that choose to achieve superior performance compared to their competitors through relentless innovation and competitive actions (D'Aveni, 1996; Teece et al., 1997). In dynamic and fast-changing industries and markets, more competitive actions and complex action repertories ultimately translate into competitive advantage (Eisenhardt & Martin, 2000; Sambamurthy et al., 2003).

Reviewing the business agility definitions available in the current literature has highlighted two general themes (Yang & Liu, 2012). In both of these themes, agility is considered a capability, which is a 'firm's capacity to deploy resources, usually in combination, using organisational processes, to effect a desired end' (Amit & Schoemaker, 1993). Resources are tangible and intangible assets and the capability available to a firm that is useful in detecting and responding to market opportunities or threats (Sanchez, 1996).

The first theme, that has a closer tie to the manufacturing discipline, focuses on the internal capabilities of the firm, and tends to provide passive reaction to a change. These studies deal with business agility by improving internal capabilities that assist a firm to respond to unanticipated and sudden changes. This viewpoint considers agility as being proficient at change (Dove, 2001). These studies usually consider cost, time, quality and scope as dimensions of agility (Sherehiy, Karwowski, & Layer, 2007). Yusuf's definition (1999) is an example of this viewpoint. This viewpoint regards agility as an holistic strategy that is constructed on the extant capabilities of a lean or flexible strategy and then integrates parts of these capabilities into a new firm capability in order to adapt to unanticipated and sudden changes in the business environment (Yang & Liu, 2012).

The second theme considers business agility as a capability that proactively detects and then rapidly responds to the environmental changes. As such, the main dimensions of this approach for improving enterprise agility are sensing and responding (Sambamurthy et al., 2003). Sambamurthy (2003) defined business agility as:

> *"The ability to detect opportunities for innovation and seize those competitive*

> *market opportunities by assembling requisite assets, knowledge, and relationships with speed and surprise."*

This viewpoint, which is common in the organisational management discipline, considers business agility as an organisational capability which relies on assembly and exploitation of the firm's resources, processes, knowledge and relationships to survive a threat or strive for an opportunity (Oosterhout et al., 2006; Overby et al., 2006; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011). As the majority of the agility definitions have suggested, agility relies on the two capabilities of *sensing* and *responding* (Yang & Liu, 2012). These capabilities have their root in the dynamic capabilities concept (Dove, 2001). Dynamic capabilities, which are different from substantive (ordinary) capabilities, are a firm's ability to integrate, build and reconfigure internal and external competencies to address rapidly changing environments (Teece et al., 1997). These capabilities are in the form of organisational routines and processes that allow firms to achieve new resource configurations (Eisenhardt & Martin, 2000).

Unlike the dynamic capabilities that address all processes and capabilities required for organisations, business agility focuses only on two capabilities that are related to the processes needed to *sense* and *respond* to environmental changes with speed (Overby et al., 2006). Business agility is dynamic because organisations must continually build, adapt and reconfigure internal and external competencies to detect opportunities and threats, capture market opportunities and change and revise existing capabilities (both dynamic and substantive capabilities) (Teece et al., 1997). This indicates that such capabilities are more applicable in a *dynamic environment* in which change is unexpected or the approach for how to deal with change is unknown. In such a dynamic environment, the response has to be *timely* and *with speed* (D'Aveni, 1996; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011).

With the sensing and responding capabilities, better alignment between these two capabilities will improve performance and result in enhanced responses to the market threats and opportunities. Misalignment, on the other hand, could result in the loss of opportunities and waste of resources used to maintain each capability (Overby et al., 2006; Roberts & Grover, 2012).

In respect to the dynamic environment, Oosterhout et al. (2006) suggested unanticipated events or uncertainty can be related to three different situations: unpredictability if or when an event will happen, uncertainty about what the impact will be and/or uncertainty about the organisation's response to the event. In respect to the predictable changes (e.g. deregulation in the telecom and energy sector), whilst the events are predictable, often the speed, exact requirements of the organisation and the processes are quite unpredictable (Oosterhout et al., 2006). The consideration of system change as an unanticipated change relies on all three types of situations outlined. The development of a new system or requirements to change a system is not always planned. Similarly, the business requirements typically evolve during the project, while its impact on other downstream systems, people and processes are not fully known until the change is introduced and the change is in operation for a period.

Whilst agility is an organisational capability, research has found that a firm may be more agile in one domain (e.g. customer-based processes) than another domain (e.g. supply chain activities) (Oosterhout et al., 2006; Sambamurthy et al., 2003). This makes agility *domain-specific*. Organisations also have a different level of agility for different internal and external stimulators (change factors). This has also been reported by other scholars' (Broadbent & Weill, 1997) work in which a firm's ability to respond to one challenge promptly lacked detection and seize of a different opportunity.

Taking into account the different dimensions that have been considered for business agility, business agility has been recognised as a complex, multidimensional and context-specific concept (Sherehiy et al., 2007).

The business agility literature identifies many factors that impact business agility, for example, business network structure and governance (Roberts & Grover, 2012; Zaheer & Zaheer, 1997), entrepreneurial alertness (strategic foresight and systemic insight capabilities) (Sambamurthy et al., 2003), organisational culture (fostering knowledge sharing and learning) (Oosterhout et al., 2006), alignment between sensing and responding capabilities (Roberts & Grover, 2012) and information technology (IT).

The role of IT as a platform for business agility has been discussed and is reflected in the literature (Lu & Ramamurthy, 2011; Roberts & Grover, 2012; Sambamurthy et al., 2003). These studies have reviewed how IT infrastructure (Roberts & Grover, 2012), IT exploration and exploitation (O. K. D. Lee, 2012), IT alignment (Tallon & Pinsonneault, 2011) or IT capabilities (Lu & Ramamurthy, 2011) impact on business agility. Sambamurthy (2003) has suggested that IT's impact on business agility is indirect through the two mediators of business processes and knowledge.

In addition to the business agility concept, the IS literature refers to another type of agility which is applicable to IT. The next section provides current definitions of IT agility in the literature and adopts a definition for this study.

2.2.2 IT Agility Definition

The importance of IT agility and its impact on business agility has been discussed in different studies (Fink & Neumann, 2007; Lui & Piccoli, 2007). Unlike the business agility concept which is a well-studied subject (G. D. Bhatt, Emdad, Roberts, & Grover, 2010; Roberts & Grover, 2012; Tallon & Pinsonneault, 2011), the current study found that research into IT agility and IT's impact on this type of agility is scarce and limited. The existing literature has highlighted that the IT agility construct is not well-developed and that there is no consistent definition for it (Maurer, 2010; Sengupta & Masini, 2008). Table 2-1 provides various definitions of IT agility in the current literature:

Table 2-1 - IT agility definitions

Definition	Focus	Reference
The ability of IT infrastructure to build a system that can easily be reconfigured, scaled, deconstructed and reconstructed as needed to adapt to unanticipated changes.	IT infrastructure	(Ahsan & Ngo-Ye, 2005)
The ability of information systems development and deployment methods to swiftly adapt to changing business requirements. Context: Globally distributed software development	Development process	(Abrahamsson, Salo, Ronkainen, & Warsta, 2002; O. D. Lee et al., 2006)
Ability of the firm to identify needed changes in the information processing functionalities required to succeed in the new environment, which lends itself to the quick and efficient implementation of the needed changes.	All capabilities supporting IS changes	(Lui & Piccoli, 2007)
The ability of a firm to adapt its IT capabilities to market changes. It is about reconfiguring or replacing your information technology systems when new marketplace realities change the way you have to do business.	All capabilities supporting IS changes	(Sengupta & Masini, 2008)
The ability of IT artefacts, of information stored within those artefacts, and of the underlying processes that support and maintain the artefacts and information to quickly adapt to changing business needs.	All capabilities supporting IS changes	(Maurer, 2010)
The ability to accommodate change in information systems without incurring significant penalty in time or cost.	All capabilities supporting IS changes	(Fink & Neumann, 2007)
The capability of IT to rapidly adapt to changes in line function	All	(Tiwana &

Definition	Focus	Reference
demands.	capabilities	Konsynski, 2010)

In Table 2-1, there are different definitions available for IT agility in the literature, with some focusing on the agility of certain IT capabilities such as IT infrastructure (Ahsan & Ngo-Ye, 2005) and IS development and deployment methods (Abrahamsson et al., 2002; Conboy, 2009; O. D. Lee et al., 2006). The other group, which is the majority, do not limit the agility concept to any particular IT capability, but instead focus on the identification and implementation of needed changes in information systems (Fink & Neumann, 2007; Lui & Piccoli, 2007; Maurer, 2010; Sengupta & Masini, 2008; Tiwana & Konsynski, 2010).

The current study is consistent with the majority of the studies listed above (Fink & Neumann, 2007; Lui & Piccoli, 2007; Maurer, 2010; Sengupta & Masini, 2008; Tiwana & Konsynski, 2010) in that it does not limit the scope of IT agility to any particular IT capability and consider all capabilities required to quickly manage the change or adoption of an information system when a need rises.

Information system here refers to 'any organised combination of people, hardware, software, communications networks and data resources that collects, transforms and disseminates information in an organisation' (J. A. O'Brien & Marakas, 2005). In this definition, borrowing from the general systems concept (Von Bertalanffy, 1972), the role of people is to recognise the informational value of output. Information systems provide information to someone or something; they are not just self-operating control mechanisms and systems (Gregor & Iivari, 2007). From the IT agility perspective, however, IT agility targets the latter components of the system (hardware, software, communication networks, and data sources) to provide the output to the users (information consumers) faster.

To define the IT agility, the current study utilises the characteristics prescribed in the literature for the agility concept. The above approach reduces the risk of measuring a different concept or missing certain characteristics that are essential when dealing with IT agility.

Prior business agility research has suggested that agility has the following characteristics (D'Aveni, 1996; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011):

- It is an organisational *capability.*
- It relies on two capabilities of *sensing* and *responding*.
- Its response must be *timely* and *with speed*.

Agility is best viewed as an organisational *capability* which the capability is the firm's *ability* to deploy resources, usually in combination, using organisational processes, to effect a desired end (Amit & Schoemaker, 1993). Resources are tangible and intangible assets and capabilities available to a firm that are useful in detecting and responding to market opportunities or threats (Sanchez, 1996). Firms create competitive advantage by assembling resources that work together to create organisational capabilities. Capabilities, thus, refer to the firm's *ability* to deploy resources, usually in combination, using organisational processes, to effect a desired end (Amit & Schoemaker, 1993). Whilst the current definition of capability refers to *ability*, other definitions have considered capability as repeatable patterns of action (Sanchez, 1996; Wade & Hulland, 2004) and can include skills, such as technical or managerial ability, or processes, such as system development or integration (Wade & Hulland, 2004). The Merriam Webster dictionary defines capability as "the quality or state of being capable" or "the facility or potential for an indicated use or deployment". This indicates that a capability is the potential or ability to achieve a goal versus the process through which to achieve the goal. Having the capability, however, implies the existence of the required processes, business routines and assets to achieve the goal (A. S. Bharadwaj, 2000). On this basis, capability is here defined as firms' capacity to deploy resources (Amit & Schoemaker, 1993) rather than as repeatable patterns of actions (Sanchez, 1996; Wade & Hulland, 2004). Marketing, manufacturing and product development are examples of such capabilities (A. S. Bharadwaj, 2000).

As IT agility deals with IT's ability to sense and respond to change, it is a higher order capability. As previously suggested, IT in this context refers to the collection of IT resources and capabilities in an organisation such as IT infrastructure, human IT resources comprising technical and managerial skills, IT-enabled intangible resources such as knowledge assets (A. S. Bharadwaj, 2000) and, finally, IT processes such as planning, operations, support and delivery (Ross et al., 1996). On the *sensing* and *responding capabilities,* as indicated in the previous section, strong sensing capability and responding capability are critical to firm success in turbulent environments (Overby et al., 2006; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011). Finally, agility is especially important in dynamic, fast-paced environments. Hence, the ability to sense and respond *quickly* is also important.

Taking into consideration these characteristics, Sengupta et al.'s definition (2008) of agility as provided in Table 2-1 does not include the sensing capability: "Ability of a firm to adapt its IT capabilities to market changes. It is about reconfiguring or replacing your information technology systems when new marketplace realities change the way you have to do business" (Sengupta & Masini, 2008).

The same argument applies to Fink and Maurer's definitions (Fink & Neumann, 2007; Maurer, 2010) as listed in Table 2-1. The definition provided by Lui and Piccoli (2007) includes all three characteristics; however, it has also considered efficiency as a requirement to deliver a change: "a capability that enables the firm to identify needed changes in the information processing functionalities required to succeed in the new environment, which lends itself to the quick and efficient implementation of the needed changes". Efficiency focuses on all four measures of cost, productivity, quality and speed (Tallon, Kraemer, & Gurbaxani, 2000), whereas agility mainly focuses on firms providing a fast response to the market opportunity (D'Aveni, 1996; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011). This indicates that, while other measures such as quality, cost and productivity must not introduce a suppressing effect on the firm performance (Black & Boal, 1994) and hinder the organisation in seizing the opportunity, the speed of seizing the opportunity is critical for a firm's success in a turbulent environment (Overby et al., 2006; Sambamurthy et al., 2003; Tallon & Pinsonneault, 2011).

Due to the above limitations in utilising any of the existing definitions, the current study proposes a refined definition for IT agility, which inherits the three characteristics of agility as explained below:

> *"Ability of IT to identify needed changes in its information systems and swiftly implement required changes in an uncertain environment to support the organisation to survive and thrive."*

The proposed definition considers IT agility as a capability. It also includes the 'sensing' capability through identification and the 'responding' capability through implementation of changes and puts emphasis on the speed through swift implementation. In addition to the speed, the sense and the response need to satisfy the required level of quality and cost, otherwise they will not support the business to survive and thrive, that is, due to excessive cost or poor quality, which would further hinder the business.

The above definition also emphasises the purpose of IT agility as supporting the business to survive and thrive in a dynamic and uncertain environment. The uncertainty can apply to the business environment or the environment in which IT is operating. This could involve technology changes such as the introduction of emerging technologies and security attacks (Oosterhout et al., 2006) and the business introducing technology-enabled innovations and competitive actions in order to thrive (Sambamurthy et al., 2003). The other point implied in the definition is related to changes to the information systems which could involve reconfiguration, deconstruction, reconstruction or introduction of a new system. Such changes could originate from the organisation's external environment (as a result of social or legal pressures, mergers and consolidations, customer needs, or emerging technologies (Oosterhout et al., 2006; Overby et al., 2006)) or changes could originate internally (from new systems implementations, shifts in IT governance and changing IT strategies (Oosterhout et al., 2006)).

The definition adopted for IT agility has a few implications for this research that could be considered as the boundaries of this research. The definition as previously explained limits the scope of IT agility to changes in information systems, hence excludes any other potential services that the IT might be providing to the business. IT agility was considered as an IT capability rather than a capability that is organisation-wide and that must be developed by all business functions. This assumption becomes relevant due to IT agility being associated with changes to the information systems, which is the core function of IT.

2.2.3 IT Agility Conceptualisation

This section provides details of how IT agility should be conceptualised. While there are a few conceptualisations for the IT agility in the literature, this section justifies the conceptualisation of IT agility using two components of IT sensing and IT responding capabilities.

As presented in the previous section and in Table 2-1, the use of IT agility ranges from its application in IT infrastructure (Ahsan & Ngo-Ye, 2005) and IT system development and deployment capability (O. D. Lee et al., 2006) to agility of all IT capabilities to implement changes in information systems (Lui & Piccoli, 2007; Maurer, 2010; Sengupta & Masini, 2008; Tiwana & Konsynski, 2010). All the above studies mainly focused on one dimension of IT agility, which was the IT responsiveness. Among the above studies, two studies (Lui & Piccoli, 2007; Maurer, 2010) conceptualised the IT agility using a set of capabilities that supported IT agility. By conceptualising IT agility as a formative construct, they (Lui & Piccoli, 2007; Maurer, 2010) focused on dimensions such as technical infrastructure agility, process agility and people agility to represent the IT agility. The other studies (Fink & Neumann, 2007; Tiwana & Konsynski, 2010) conceptualised IT agility based on the responsiveness of IT and measured it using a set of reflective measures which represented how quickly a change could be delivered.

As shown above, the IT agility literature has been silent on the role of sensing capability in achieving IT agility, whereas the broader business agility recognises its role (Overby et al., 2006; Roberts & Grover, 2012; Teece, 2007). For an agile business, the sensing capability enabled by market intelligence, government relationship and research and development (R&D) allows organisations to anticipate market changes such as competitors' actions, regulatory changes or technological advancements. Oosterhout et al. (Oosterhout et al., 2006) recognised six groups of change factors that require sensing and responding capabilities. Among these change factor groups, two of the groups were directly related to IT and IT systems. Technology changes such as emerging technologies to connect to partners' information systems; and internal changes in the organisation such as restructure of IT unit and mergers and acquisitions (Oosterhout et al., 2006) present cases that the IT requires a proactive sense of change to provide the organisation with a better chance of providing a swift and fast response.

Furthermore, in a study assessing the impact of IT infrastructure on customer agility, Roberts and Grover (2012) showed that the sensing capability has a stronger effect than the responding capability on the ability of the company to produce the desired result. This could be due to the response being more aligned with the customer need and more targeted to achieve the optimum result.

Therefore, the current study considers that IT's ability to meet the business needs not only requires IT responding capability but also relies on the IT sensing capability. Consequently, the IT agility is conceptualised using its two dimensions of IT sensing and IT responding capabilities, as defined below.

IT sensing capability is:

> *"Ability of IT to swiftly sense needed changes in its information systems in order to support business to survive and thrive in an uncertain environment."*

As with the IT sensing capability, the IT responding capability is:

> *"Ability of IT to swiftly respond to needed changes in its information systems in order to support business to survive and thrive in an uncertain environment."*

2.2.4 IT Agility Antecedents

This section reviews the antecedents of IT agility from the IT literature and group them in four categories of 'people', 'structure', 'technology' and 'process'. Among the four categories, there are inconsistent reports regarding the impact of technology on agility. On one front, technology has been reported to have a positive impact on agility (Oosterhout et al., 2006; Park & El Sawy, 2012; Tiwana & Konsynski, 2010) but, on the other front, it has been considered to be an inhibitor of agility. The above inconsistency motivates the current study to review the impact of SOA, which is a new style of system design and architecture, on IT agility.

On the classification of the IT agility antecedents, the current study adopts the classification scheme suggested by the Lui and Piccoli (2007), who studied the information system agility using the socio-technical theory (Bostrom & Heinen, 1977; Mumford, 2003). Lui and Piccoli (2007) argued that information system agility is created from certain characteristics in the interacting social and technical systems, represented by: people, structure, technology and process. Each component needs to be agile which depends on its level of flexibility and rapid adjustment when required.

The current study adopts similar classification when it reports the antecedents found in the literature, as summarised in Table 2-2. In this classification, 'technology' refers to the IT and technical components of the information system, whereas 'process' represents a set of steps to perform an activity or set of activities that transforms inputs to outputs. The two remaining components are people and structure, of which the former represents the individuals associated with the information systems and their knowledge and skills and the latter is related to the flexibility and decision-making ability afforded to individual members of the information system (Lui & Piccoli, 2007).

Review of the literature identified a number of studies (Alliance, 2001; G. Lee & Xia, 2010; Mangalaraj, Mahapatra, & Nerur, 2009; McAvoy & Butler, 2009) that discuss the ‘process’ component, mainly by focusing on agile methods such as the Scrum, eXtreme Programming, dynamic systems development methods (DSDM) and feature-driven development (FDD) (Conboy, 2009). Studies adopting this perspective have uncovered the influence of an array of factors such as team autonomy, team diversity, individual knowledge, empowerment, project team management, team leadership, technological compatibility, nature and size of task, resource constraints, and method characteristics on the effective employment of agile development methods (Alliance, 2001; G. Lee & Xia, 2010; Mangalaraj et al., 2009; McAvoy & Butler, 2009).

Other studies have focused on the ‘people’ and reviewed the role of IT personnel capabilities in achieving IT agility (Clark, Cavanaugh, Brown, & Sambamurthy, 1997; Fink & Neumann, 2007): ‘IT business expertise and knowledge’ and ‘IT skills’ determine the firm’s ability to quickly develop and deploy information systems within short periods (Clark et al., 1997). The technical and behavioural capabilities of IT personnel through infrastructure capabilities showed a positive impact on the three dimensions of IT-dependent business agility: IT-dependent information agility, IT-dependent system agility and IT-dependent strategic agility (Fink & Neumann, 2007).

In respect to the ‘technology’ and ‘structure’ components, Tiwana and Konsynski (2010) reviewed their impact on IT agility. This study (Tiwana & Konsynski, 2010) reviewed how IT modularity influenced IT alignment through IT agility. Building on modular systems theory, Tiwana et al. (2010) showed that IT modularity improved IT agility which improves IT alignment. They also found that IT governance decentralisation had a negative impact on IT agility, whilst it moderated positively the impact of IT modularity on IT agility by giving line functions greater autonomy over IT decisions. They suggested the negative impact of IT governance decentralisation can increase interdepartmental coordination which could overwhelm the advantage of decentralised governance. Regarding the impact of modularity on agility, they suggested that higher modularity increases the rigidity in IT architectures (e.g. enforcing standardisation to increase modularity) which increases IT agility.

In addition to the above studies, there are also a set of exploratory or conceptual research studies which have reported a set of factors that can have either positive (enabler) or negative (inhibitor) impact on IT agility and business agility (Allen & Boynton, 1991;

Oosterhout et al., 2006; Tallon, 2008; Trinh-Phuong, Molla, & Peszynski, 2010).

Table 2-2 summarises the findings reported in the above studies and the antecedents outlined earlier for IT agility.

Table 2-2 - IT agility antecedents

Antecedents	Component	Findings	Source/ Method
IT personnel capabilities	People	Technical and behavioural capabilities of IT personnel through infrastructure capabilities positively affect system agility.	(Fink & Neumann, 2007)/Survey
IT business expertise, IT skills	People	IT business expertise and knowledge, and IT skills determine the firm's ability to quickly develop and deploy information systems within short period.	(Clark et al., 1997)/Case study
Managerial IT capabilities, technical IT capabilities, environmental dynamism	People & Process & Structure	Managerial capabilities (IT business partnership, strategic IT planning, post-implementation review) both directly and indirectly through technical capabilities have a positive impact on organisational process agility. IT governance and managerial IT capabilities are more effective for firms in volatile markets, while technical IT capabilities are more effective for firms in stable markets.	(Tallon, 2008)/ Survey
Training level, job rotation	People	Training provides employees with the ability to quickly redeploy when needed and perform alternative tasks. Higher frequency of employee job rotation in different positions allows the organisation to take rapid action when changes are needed.	(Lui & Piccoli, 2007; Tsourveloudis & Valavanis, 2002)/Case study
A range of factors such as individual knowledge, empowerment, project team management	People & Process	Reports a set of factors that influence the effective employment of agile development methods.	(Mangalaraj et al., 2009; McAvoy & Butler, 2009)/ Case studies
Team autonomy, team diversity,	Process & Structure	The team autonomy and team diversity positively impact on the software development agility (excluding the impact of team autonomy on team response extensiveness which has been reported as a negative impact)	(G. Lee & Xia, 2010)/Survey
Enterprise systems	Technology	Enterprise systems, for example, ERP, provides four options to address the systems agility requirements: use built-in capabilities, build on data and process integration, use add-on systems and install a vendor patch.	(Goodhue, Chen, Boudreau, Davis, & Cochran, 2009)/ Case

Antecedents	Component	Findings	Source/ Method
			study
Information system architecture	Technology	Information systems are inflexible and disablers of flexibility. Two solutions are proposed to address the “speed and flexibility” and “low cost and efficiency” challenges: the low-road option (full extent decentralisation of IS) and high-road (centralisation of IS)	(Allen & Boynton, 1991)/ Conceptual
Enterprise software	Technology	Complexity, high risk, inflexibility (rigidity) and uncertainty of enterprise systems (e.g. ERP) have created barriers to change rather than agility.	(Rettig, 2007)/ Conceptual
IT	Technology	IT can be both enabler and inhibitor for business agility. Complexity of information systems (e.g. hardcoded embedded business processes, complex nests of links between applications), slow time to market (long time to implement requirement changes), inflexibility of systems, rigid IT architecture and inflexibility to support external integration are some of the factors that inhibit business agility.	(Oosterhout et al., 2006)/Case study
IT capability	Technology	While in turbulent environments IT plays a core role in achieving high performance, in stable environments a high level of IT capability should be absent for a configuration to result in a high performance. This finding also implies a possible contingency effect of environmental turbulence.	(Park & El Sawy, 2012)/ Case-Configuration theory
IT architecture modularity	Technology	IT architecture modularity moderated by IT governance decentralisation positively impacts on IT agility which, in turn, improves IT alignment.	(Tiwana & Konsynski, 2010)/Survey
IT governance decentralised	Structure	IT governance decentralisation (IT specification decentralisation and IT implementation decentralisation) has a negative impact on IT agility.	(Tiwana & Konsynski, 2010)/Survey
Workforce empowerment, distributed decision-making authority, and flatter managerial hierarchies	Structure	An empowered workforce and distributed decision-making authority allow employees to take leadership in decision-making and to implement the decisions quickly. Flatter managerial hierarchies enhance communication within the organisation and speed up the decision-making process in the face of more general and strategic level changes.	(Tsourveloudis & Valavanis, 2002)/ Conceptual

As shown in Table 2-2, the literature has suggested that there are different factors across the people, process, structure and technology components that can impact on IT agility.

Regarding the ‘people’ and ‘process’ components, the literature has provided a consistent view in respect to the impact of these components on agility. The literature however has suggested a partial view for the ‘structure’ component, especially concerning the IT governance and its impact on IT agility (Tallon, 2008; Tiwana & Konsynski, 2010; Tsourveloudis & Valavanis, 2002). While there are suggestions that distributed decision making can improve agility (Tsourveloudis & Valavanis, 2002), Tiwana et al. (2010)’s study partially supported this proposition.

On the impact of the ‘technology’ on ‘agility’, there are inconsistent reports regarding such impact. On one front, technology has been reported to have a positive impact on agility (Oosterhout et al., 2006; Park & El Sawy, 2012; Tiwana & Konsynski, 2010) but, on the other front, it has been considered to be an inhibitor of agility. Factors such as complexity of information systems, IT slow time to market (longer to implement requirement changes), inflexibility of systems, rigid IT architecture and inflexibility to support external integration have been identified as some of the factors impacting on agility (Allen & Boynton, 1991; Oosterhout et al., 2006; Tallon, 2008).

The inconsistency reported above motivates the current study to review the impact of SOA, which is a new style of system design and architecture, on IT agility and assesses if SOA can address issues such as inflexibility of systems, rigid IT architecture and inflexibility to support external integration.

While this research sets its boundary and focus on the impact of technology on IT agility, it recognises the independent impact of the other components, namely, people, structure and process on IT agility.

2.3 Service-Oriented Architecture (SOA) and its impact on IT Agility

Following the discussions in the previous section, this section introduces the SOA concept, its impact on IT agility and the knowledge gap that exists in how SOA affect the IT agility. To achieve this goal, the next subsections first define what SOA is and how SOA is different from other design paradigms. It then reports the benefits and challenges reported in the SOA literature, particularly to achieve agility. Using the literature, it then identifies the current knowledge gap on the impact of SOA on the IT agility.

Finally, to take the initial step in investigating the knowledge gap, the SOA literature is reviewed to identify the extant SOA conceptualisation and SOA characteristics.

2.3.1 Defining SOA

Considering that SOA is a style of IT architecture, the first step is to define architecture. IT architecture is (Tiwana & Konsynski, 2010):

> *"The overarching structure and properties of the relationships among the systems and applications in an organization's IT portfolio."*

IT architecture includes the organising logic for applications, data, and infrastructure technologies, as captured in assets of policies, technical choices and design rules, and is intended to enable the organisation's business strategy (Ross, 2003).

In relation to the definition of term 'architecture', three definitions emerge from the literature for SOA. Papazoglou (2003) defined Service-Oriented Computing (SOC) and SOA as:

> *"Service-Oriented Computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications/solutions. Services are self-describing, platform-agnostic computational elements that support rapid, low-cost composition of distributed applications. Services perform functions, which can be anything from simple requests to complicated business processes. To build the service model, SOC relies on the Service-oriented Architecture (SOA), which is a way of reorganizing software applications and infrastructure into a set of interacting services."* (Papazoglou, 2003)

A second definition (Sprott, 2004) considers SOA as:

> *"A style of architecture that promotes the reorganisation of enterprise information resources as independent and reusable services"*

And finally, Arsanjani (2004) defined SOA as:

> *SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business processes realization using interface-based service descriptions.*

The first definition (Papazoglou, 2003) distinguishes the SOC concept from that of SOA and provides some of the characteristics of services such as self-describing and platform-agnostic, which are the underlying elements of SOC and SOA. The second definition (Sprott, 2004) only focuses on SOA and suggests the reorganisation of enterprise information resources as independent and reusable services. The last definition (Arsanjani et al., 2004) suggests an objective for employing SOA which is to enhance business flexibility by enhancing IT capability to dynamically reconfigure systems supporting business processes.

The current study adopts the Papazoglou's definition (2003) to describe SOA. The selected definition provides a more precise definition for SOA by first defining SOC and the services concept. Papazoglou's definition (2003) also focuses on the reorganisation of systems and interaction between services which is more consistent with the definition of architecture (Tiwana & Konsynski, 2010) and the relationship among applications and systems. This selected definition, by focusing on "*support rapid, low-cost composition of distributed applications",* encapsulates characteristics such as reusability.

SOA as a new concept has introduced two changes to the IT landscape: changes to the way systems are designed and integrated, and addition of new infrastructure components to the IT infrastructure.

From the perspective of the design of information systems, SOA promotes the grouping functionality required in an end-to-end business process and packages it as interoperable services (Newcomer & Lomow, 2004). These services are typically made available for reuse within an organisation or might be exposed externally as a commercial offering (Bell, 2008). The exposed services can be combined to deliver a new business service which creates services with different granularities. The exposed services communicate with each other and can coordinate an activity between two or more underlying services. In an SOA environment, services can be accessed without knowledge of their underlying platform implementation (Bell, 2008).

In addition to how applications are designed and implemented, SOA also prescribes additional IT infrastructure. IT infrastructure is considered as a set of shared, tangible IT resources that form a foundation to enable present and future business applications (Broadbent & Weill, 1997; Duncan, 1995). In the IT literature, IT infrastructure consists of *IT components* (e.g. computer and telecommunication technologies), *shared services* (e.g. EDI capabilities, firm-wide database management) and *human IT infrastructure,* people with knowledge, skill and experience, binding the commodity components to the shared infrastructure services (Broadbent & Weill, 1997). In this study, the IT infrastructure only refers to the IT components and shared services. The human IT infrastructure will explicitly be mentioned, when required. The SOA infrastructure facilitates the integration and exchange of data between multiple systems when they participate in a business process. The typical infrastructure components considered for SOA are (Papazoglou, 2008):

- Middleware facilitating connectivity among services (Enterprise Service Bus), which provides:
    - Connectivity and Message Exchange between service providers and consumers,
    - Support and provision of facilities such as transactions, security, performance metrics, etc. in a declarative and composable manner,
    - Support for dynamic configuration
    - Monitoring of internal behaviour and state to management systems (services)
    - Performing data and protocol adaptation,
    - Support for services discovery.
- Service composition and choreography platform which allows independent services with well-defined interfaces to be called in a defined sequence as part of a business process automation using industry standards such as business process execution language (BPEL). This layer also includes the modelling of the business processes and monitoring of the service level agreements (SLAs) for a business process.
- Service management and monitoring to manage exposed services and to monitor them as they interact. Service management spans a range of activities from installation and configuration to collecting metrics and tuning to ensure responsive service execution. It includes many interrelated functions such as service level agreement (SLA) negotiation, management, auditing, monitoring, and troubleshooting, service lifecycle/state management, performance management and services and resources provisioning.

It is important to note that SOA as a style of architecture does not mandate these infrastructure components. They have, however, been positioned to manage and facilitate the SOA characteristics such as service independence.

While this section has explained the SOA concept and its fundamental design components, the question remains of how SOA is different from other styles of architecture. The next section addresses this question and provides more details on how SOA has changed the way that information systems are designed.

2.3.2 SOA – A New Architectural Paradigm

This section describes how SOA is different from the existing styles of architecture or enterprise systems such as enterprise resource planning (ERP). Understanding these differences could suggest that SOA can introduce a different impact on IT agility than existing styles of architecture or enterprise systems.

SOA covers and utilises the existing concepts of object-oriented and component-oriented paradigms; however, SOA, unlike other styles delivers the expected functionalities as self-describing, platform-agnostic computational elements which are exposed in a distributed environment (Zimmermann et al., 2004). SOA is a multiple-layer distributed information system (IS) architecture which covers concepts relevant to the *IT infrastructure* (e.g. through the introduction of new middleware ad technologies – Enterprise Service Bus, service registry and repository) (Papazoglou, 2008), *application design and architecture* (e.g. distributed and loosely coupled architecture) and, finally, *business process modelling and automation* (Zimmermann et al., 2004). The process modelling and automation allow SOA to contribute in business process improvement (Luthria & Rabhi, 2009c). Process automation could include internal and external processes through the orchestration of services and data exchange with partners and external organisations (Moitra & Ganesh, 2005).

SOA also changes how information systems are designed. Traditionally, the architectural paradigms focused on the design of vertical systems, each with certain functionalities, and a business process tended to be across multiples of these systems. In SOA, however, the design has shifted to a vertical design which focuses on design and implementation of a business process end to end. This makes IS process-centric rather than application-centric (Demirkan et al., 2009; Vidgen & Wang, 2006). While information systems such as ERP are also process-centric, they are predesigned fixed systems with limited extension capabilities.

In the new SOA paradigm, information systems usually have multiple hierarchical layers that can be shared with other systems. The application service layer as the lowest layer, encompasses the data access functionality. The service layer exposes a service in conjunction with its underlying layer. The orchestration layer tends to be cross-functional and cross-application and typically implements a business process. And finally, the desktop integration layer providing viewpoints for users (Erl, 2004; Legner & Heutschi, 2007).

This shift in design has raised needs for new system analysis and design processes which has resulted in a new framework and methodologies being suggested in the SOA literature covering identification, analysis and management of services (Arsanjani et al., 2008; Erradi, Anand, & Kulkarni, 2006; Papazoglou & Van Den Heuvel, 2006; Ramollari, Dranidis, & Simons, 2007; Terlouw & Dietz, 2008). These processes and framework are outside the scope of this study.

The changes that SOA has introduced to the IT landscape indicate that it can have a different impact on IT performance and deserves special attention to analyse and understand this impact. The next section reviews some of these impacts as outlined in the SOA literature.

2.3.3 SOA Benefits and Challenges

This section reviews and reports the SOA benefits and challenges. Reviewing the conceptual and empirical benefits of SOA indicates the SOA realised business values. Review of SOA benefits reveals several benefits including faster software development, IT flexibility and improved business agility. While certain elements of IT responding capability can be found in the reported benefits, the extant literature is however silent on the impact of SOA on the IT sensing capability. Additionally, the literature presents a paradox concerning the impact of SOA on the IT complexity as covered below.

SOA Benefits

The SOA conceptual studies have promised to overcome many existing limitations of information systems and IT such as inflexibility and complexity by offering a higher degree of standardisation, uncomplicated interoperability, reusability and high flexibility

(Beimborn & Joachim, 2009; Hagel & Brown, 2001; Mueller, Viering, Ahlemann, & Riempp, 2007).

On the empirical front as Table 2-3 highlights, SOA empirical studies have already reported a variety of benefits for SOA. Most of the reported benefits are IT-specific. The benefits include faster development due to modularity and reuse (Abelein et al., 2009; Baskerville et al., 2005; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c); complexity hiding and reduction due to abstraction (Baskerville et al., 2005; Luthria & Rabhi, 2009c); faster system integration due to standards, interfaces and loose coupling (Baskerville et al., 2005; Henningsson, Svensson, & Vallen, 2007; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c; Yoon & Carter, 2007); and improved business and IT communication (Legner & Heutschi, 2007; Yoon & Carter, 2007).

Table 2-3 – SOA benefits

Category	Benefit	References
Software and integration development	Faster development due to reuse (associated with modularity) and toolsets	(Abelein et al., 2009; Baskerville et al., 2005; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c; J. Schelp & Aier, 2009)
	Hiding complexity due to abstraction	(Baskerville et al., 2005; Luthria & Rabhi, 2009c)
	Composition of services to build new functionality	(Baskerville et al., 2005)
	Easier to integrate systems due to standards, interfaces and loose coupling	(Baskerville et al., 2005; Henningsson et al., 2007; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c; Yoon & Carter, 2007)
	A degree of adaptability and flexibility in development, allowing change or unclear requirements	(Baskerville et al., 2005)
System maintenance and management	Reduce complexity by reducing the number of interfaces, redundancy, possibility of preventative maintenance and easier combination of different technologies	(Abelein et al., 2009)
	Improve data quality due to interface specifications	(Abelein et al., 2009; Haines & Haseman, 2009; J. Schelp & Aier, 2009)
	Improve process monitoring	(Abelein et al., 2009; Haines &

Category	Benefit	References
		Haseman, 2009)
Business and IT Alignment	Better business and IT alignment due to process visibility, improved communication	(Legner & Heutschi, 2007; Yoon & Carter, 2007)
	Better process visibility due to process modelling	(Luthria & Rabhi, 2009c)
IT flexibility	Extendibility of existing services with limited impact	(Abelein et al., 2009; Baskerville et al., 2005; J. Schelp & Aier, 2009)
IT knowledge improvement	Improved IT knowledge due to interfaces and their documentation	(Legner & Heutschi, 2007)
	Improved IT knowledge about business processes and their documentation/models	(Legner & Heutschi, 2007)
Improved business agility	Quick IT response to market change or customer demand	(J. Schelp & Aier, 2009; Yoon & Carter, 2007)

On the non-IT benefits, the above table reports the 'improved business agility' as the main business benefit of the SOA. The next section expands on the studies that focused on the SOA agility value.

SOA Delivering Agility

As shown in Table 2-3, the review of SOA literature revealed two empirical studies (J. Schelp & Aier, 2009; Yoon & Carter, 2007) that focused on business agility. The SOA conceptual studies (Hagel & Brown, 2001; Huang & Hu, 2004; Lim, Ishikawa, Platon, & Cox, 2008) followed similar trail and explored the impact of SOA on the business agility.

While the researcher found no study that focused on IT agility and its two dimensions of IT sensing and responding capabilities, the measurements or descriptions of agility in the above studies somehow lend themselves to IT's responsiveness rather than business responsiveness. Therefore, the above studies are reviewed here to assess if they can shed light on how SOA impacts the IT agility.

Starting with the conceptual studies, Huang and Hu (2004) have argued how web services and SOA can improve business agility across the three dimensions of operational, partnership and customer agility. On the operational front, web services can enable the creation of a flexible service-oriented enterprise IT infrastructure and a faster software development capability, which can improve operational agility. Web services through their industry-accepted standards and protocols can be the basis for facilitating inter-firm communication and collaboration which can improve partnering and customer agility (Huang & Hu, 2004).

SOA provides organisations with a foundation on which the organisation becomes more agile (Hagel & Brown, 2005; Tiwana & Konsynski, 2010). In particular, IT architecture modularity (the degree to which an organisation's IT portfolio is decomposed into relatively autonomous subsystems) can plausibly foster agility by decreasing the need for overt coordination among organisational subunits (Hagel & Brown, 2001).

On the empirical research front, however, there are few studies that have reviewed the impact of SOA on agility (J. Schelp & Aier, 2009; Yoon & Carter, 2007).

Yoon and Carter (2007) through analysing publicly available secondary data (existing SOA studies) concluded that SOA can improve business agility through 'easier system integration', 'better alignment of IT with business', 'quick IT response to market change or customer demand', 'better data flow' and 'better customer service'. The above authors considered improved business agility as quick IT responses to a range of events including business environment, market changes and customer demands.

Whilst this study (Yoon & Carter, 2007) has provided some insight on SOA's potential to improve agility, the authors (Yoon & Carter, 2007) have suggested further studies on this topic to investigate the impact of SOA on agility using in-depth investigation to assess which SOA characteristics contribute to agility.

Schelp and Aier (2009) took one extra step and, through five case studies, compared and reported if SOA contributed to business agility. The authors used Yusuf's definition of agility (1999), which has its root in the manufacturing discipline and considered agility to mean being proficient at change (Dove, 2001; Sherehiy et al., 2007). Agility in this discipline is usually considered as a passive response to unanticipated and sudden changes through improvement to speed and a better time to market, flexibility, quality and profitability. Schelp and Aier (2009) assessed SOA's contribution to agility by assessing if SOA contributed to one of the competitive bases suggested above (speed and a better time to market respectively, flexibility, quality and profitability) (Sharifi & Zhang, 1999; Yusuf et al., 1999). All the studied cases indicated that SOA has improved 'speed/time to market' and 'quality' consistently and 'reuse', 'flexibility' and 'profitability' in some of the cases. They have also observed that complexity of the overall system has grown due to additional decoupling and modularity (J. Schelp & Aier, 2009). They argued that sustainability of agility through SOA is dependent on certain factors such as management and governance of architecture in the organisation and maintaining the level of system complexity (J. Schelp & Aier, 2009).

SOA Challenges

There are also challenges that have been reported for SOA. As listed in Table 2-4, these challenges include additional complexity due to further decoupling and modularity (Baskerville et al., 2005; Kokko et al., 2009; J. Schelp & Aier, 2009) and trade-offs between performance and modularity due to more abstraction (Eckert, Bachhuber, Miede, Papageorgiou, & Steinmetz, 2010; Luthria & Rabhi, 2009c).

Table 2-4 - SOA challenges

Category	Challenge	References
Software and integration development	Additional complexity due to further decoupling and modularity	(Baskerville et al., 2005) (Kokko et al., 2009; J. Schelp & Aier, 2009)
	Trade-off between performance and modularity due to more abstraction.	(Eckert et al., 2010; Luthria & Rabhi, 2009c)
	No solid evidence of reuse or increased flexibility, when used in a bottom-up approach.	(Kokko et al., 2009)
System maintenance and management	Complication with change management due to reusability and dependency.	(Kokko et al., 2009; Luthria & Rabhi, 2009c)

The complexity and complications is the consistent theme observed in the above table. As highlighted in the agility literature, complexity is an agility inhibitor. To better understand the issue of complexity and the factors that contribute to its creation, this study consulted the literature concerning the complexity concept. Complexity is generally viewed as an intrinsic or structural property of the system which could encapsulate the other issues. Buzacott (1999) defined structure as how individual system components relate to each other and how the relationship determines overall system behaviour. The more significant relationships in an enterprise are the material and information flows between the system components, the organisational relationships and the communication network connecting people with other people or machines. Complexity hinders a company's ability to react to change and reconfigure its products, processes or organisational structure (Arteta & Giachetti, 2004).

By relying on the extant knowledge, Xia and Lee (2003) proposed a multidimensional framework for understanding and measuring the complexity of an information system development project. Their framework consists of two dimensions: (1) technological versus organisational aspects and (2) structural versus dynamic factors.

The *technology complexity* is considered to be a composite measure of diversity of technologies, database intensity and system integration effort (Meyer & Curley, 1991). McKeen (1994) extended this measurement (Meyer & Curley, 1991) and proposed that *system complexity* includes the complexity of technology platform, design techniques and computing languages; development methodologies; and system integration.

Organisational complexity, however, is concerned with the complexity of the organisation environment. In an information system development project, the organisation elements include the user groups, top management, project team, external contractors and vendors, organisational structure and business processes (Xia & Lee, 2003).

Xia and Lee (2003) defined structural complexity as: (1) variety, multiplicity and differentiation of project elements; and (2) interdependency, interaction, coordination and integration of project elements. Dynamic complexity, however, is defined as uncertainty, ambiguity, variability and dynamism which are caused by the environment change.

Applying this framework (Xia & Lee, 2003) to SOA suggests that SOA can increase the structural complexity of the system. This can be associated to the SOA decomposition leading to higher multiplicity of elements and higher integration between modules (Baskerville et al., 2005; Luthria & Rabhi, 2009c).

2.3.4 SOA Governance

One of the contributing factors that the SOA literature (Joachim, 2011; Joachim, Beimborn, & Weitzel, 2013; Luthria & Rabhi, 2009c; Yoon & Carter, 2007) reports in the study of SOA business value is SOA governance.

Brauer et al. define SOA governance as (Brauer & Kline, 2005):

> *"SOA Governance is a set of solutions, policies and practices which enable companies to implement and manage an enterprise SOA."*

The main task of SOA governance is to define and introduce company-wide policies for the adoption and operation of an SOA, as well as to introduce mechanisms, which control their enforcement (Manes, 2005; J Schelp & Stutz, 2007; Windley, 2006). SOA governance targets issues such as enterprise organisation and structure in order to support and manage ownership of the services, service life cycles, standards, financial reports and accounting, etc.

There is a number of SOA governance models (Afshar, Cincinatus, Hynes, Clugage, & Patwardhan, 2007; Marks & Bell, 2006; Niemann, Eckert, Repp, & Steinmetz, 2008) suggested in the literature. Majority of the models and SOA studies (Bieberstein, Bose, Walker, & Lynch, 2005; J. H. Lee, Shim, & Kim, 2010; Schepers, Iacob, & Van Eck, 2008; Walker, 2007) propose a central governance structure for the SOA to manage and promote SOA and its adoption in the organisation. Some of the above studies take an extra step and suggest a decision-making body for the SOA separate from the existing IT decision-making body.

On the separation of decision-making body, however, a recent study (Joachim et al., 2013) showed that a newly created decision-making body for SOA can actually hinder the reuse of the services in the organisation. The authors of the above study (Joachim et al., 2013) suggested that creation of a new governing body, when IT already has an existing governing body, can create confusion and reduce the reuse of the services.

The IT governance literature offers a mixed view on centralisation of IT governance (Magnusson, 2013; Tiwana, Konsynski, & Venkatraman, 2013), which encompass centralisation of IT specification and IT implementation decisions (Fama & Jensen, 1983). The IT specification included decisions about what business processes in the line function IT must support, the service ownership, the associated constraints (schedule, budget, quality), objectives, priorities, and performance expectations (e.g., service levels). However, the IT implementation encompasses decisions about the methods, platforms, IT standards and policies, and IT sourcing (e.g., outsourcing, purchase, or internal development) (Tiwana & Konsynski, 2010; Walker, 2007).

2.3.5 Knowledge Gap on the Impact of SOA on IT Agility

As highlighted in the above, SOA can deliver many benefits. The SOA identified benefits have been associated with the SOA due to its specific characteristics such as modularity and standardisation. These characteristics have delivered benefits such as reusability and complexity hiding and reduction (Baskerville et al., 2005; Luthria & Rabhi, 2009c; Xia & Lee, 2003).

Conversely, there are also reports of SOA increasing structural complexity of the system (Kokko et al., 2009; J. Schelp & Aier, 2009), which hinders the agility (Allen & Boynton, 1991; Oosterhout et al., 2006; J. Schelp & Aier, 2009; Tallon, 2008).

Despite the above paradox and agility being the main SOA benefit, the existing SOA studies do not shed light on the mechanisms involved in the interaction between SOA and agility.

Additionally, none of the above studies explicitly reviewed the IT agility and the IT sensing capability which plays a crucial role in achieving IT agility. Further study on the topic is consistent with recommendation from both studies (J. Schelp & Aier, 2009; Yoon & Carter, 2007), which suggested further study of SOA's impact on agility.

A study which provides a theoretical explanation on the impact of SOA characteristics on the IT sensing and responding capabilities can shed light on the SOA complexity paradox and how it can be managed through the SOA characteristics. Additionally, such a study can provide insight on the role of each SOA characteristic in realising IT sensing and IT responding capabilities.

2.3.6 SOA Conceptualisation

This section reviews the process of conceptualisation of SOA in the current study. It first reviews the existing SOA literature to assess how other studies have conceptualised SOA. Through this process, it also identifies the characteristics that the SOA literature has reported for the SOA.

Since SOA is still in the realm of IT systems, the IT literature is also consulted and compared to what the SOA literature reports. The comparison between the two literatures can assist the current study to adopt a refined set of concepts as its a priori constructs.

2.3.6.1 Conceptualisation of SOA as an IT Asset

To identify how SOA should be conceptualised in the current study, the SOA literature is reviewed first. The review of literature presents an opportunity to conceptualise SOA as an IT asset, which its behaviour can be captured using its characteristics.

The SOA literature, while offering multiple studies reporting the benefits of SOA through descriptive case studies from several researchers (Baskerville et al., 2005, 2010; Eckert et al., 2010; Henningsson et al., 2007; Legner & Heutschi, 2007; Luthria & Rabhi, 2008, 2009a, 2009c; Moitra & Ganesh, 2005; Yoon & Carter, 2007), only has limited studies which have conceptualised SOA in the study of SOA business value (Beimborn & Joachim, 2010; Daskalakis & Mantas, 2008; Joachim, Beimborn, Schlosser, & Weitzel, 2011; Oh, Leong, Teo, & Ravichandran, 2007).

Among the above studies, two studies, that is, (Beimborn & Joachim, 2010; Joachim, Beimborn, Schlosser, et al., 2011)) have conceptualised SOA based on its level of adoption in the organisation. The first study (Beimborn & Joachim, 2010) reviewed the impact of SOA adoption and business process management on business process quality. The second study (Joachim, Beimborn, Schlosser, et al., 2011) assessed the impact of SOA adoption on technical IT flexibility through close IT/business collaboration. This study (Joachim, Beimborn, Schlosser, et al., 2011) considered a broader scope for the service orientation and includes its adoption in the modelling of business activities and non-technical activities. Since both studies (Beimborn & Joachim, 2010; Joachim, Beimborn, Schlosser, et al., 2011) conceptualised the SOA adoption, their selected instruments did not measure the SOA characteristics, hence, this was not suitable for the current study which requires conceptualisation of SOA using its characteristics rather than its level of adoption in an organisation.

Daskalakis and Mantas (2008) in another study used the system quality and information quality to measure the effectiveness of SOA in achieving health care interoperability. This study (Daskalakis & Mantas, 2008) did not provide the instrument for the measurement of factors involved in the system and information quality. Also, the factors considered in this study were not specific to the SOA characteristics.

The last study (Oh et al., 2007) conceptualised SOA as an IT asset. SOA was conceptualised as IT technical standards (indicators: extensible mark-up language [XML], web services description language [WSDL], SOAP, UDDI) and IT architectural design (indicators: reusable, modular, integrated, interoperable, configurable). While the current study focuses on the characteristics of SOA instead of particular technical standards such as WSDL and SOAP, their conceptualisation approach (Oh et al., 2007) can be adopted and extended here to represent both SOA and SOA platform characteristics.

The conceptualisation of SOA as an IT asset, particularly an intangible asset, lies on SOA being a style of architecture and being concerned with rules governing the overarching structure and properties of systems and their relationships (Tiwana & Konsynski, 2010). The rules are knowledge assets that can be captured in information repositories (A. S. Bharadwaj, 2000). Such rules are presented through characteristics such as system modularity and connectivity, which are *embedded* in a system upfront during the system design (Tiwana & Konsynski, 2010). Conceptualising SOA as an IT asset is also consistent with Ross (1996), who considered architecture and standards as technology assets.

Conversely, there are other studies (Feeny & Willcocks, 1998; Peppard & Ward, 2004) in the IT literature that have considered architecture an IT capability. Such studies however have mainly focused on the architecture planning rather than the architecture itself. For instance, Feeny et al. (1998) recognised nine core IS capabilities with architecture planning as one of these identified capabilities. Architecture planning consists of creating a coherent blueprint for a technical platform that responds to current and future business needs. The outcome of such a blueprint is to create capabilities such as flexibility and integration in IS services (Feeny & Willcocks, 1998). Similarly, Peppard (2004) also considered architecture planning to be in one of their six domains of IS capabilities. Finally, Wade and Hulland (2004) considered architecture as a capability in IS planning and change management.

To clarify the use of characteristics vs. capability, the SOA characteristics reflect the SOA ‘functionings’, which is what SOA is and it does (Sen, 2008). The capability, however, represents what SOA can achieve in combination with its surroundings to establish a new functioning for itself or its surrounding (Sen, 2008). For an instance, a person is thin (*being*) and runs fast (*doing*). Combination of such characteristics enables the person to be agile (*if he decides to - capability*).

Since the focus of the study is on SOA and its functions, the current study conceptualises the SOA through its characteristics as an intangible IT asset to assess its impact on IT agility, which is a dynamic capability.

2.3.6.2 Review of SOA Characteristics to develop a set of a priori constructs

This section provides a literature review on the characteristics of SOA with an intention to develop a set of a priori constructs for the SOA conceptualisation. To achieve this goal, the SOA literature is first reviewed and a set of characteristics are reported, as presented in Table 2-5. To refine these concepts, the IT literature is then consulted and several constructs that are relevant to the current study is captured in Appendix A. The concepts captured from the SOA literature, Table 2-5, is then compared with similar concepts reported from the IT literature, Appendix A, to extract a set of a priori constructs for the SOA conceptualisation. The result of this contrast has been captured in Appendix B covering a proposed definition for each concept, its proposed indicators, traceability to original SOA characteristic and any potential gap that need to be addressed in the current research.

On the SOA characteristics, Table 2-5 summarises the characteristics found in the SOA literature, which are applicable to the SOA based systems and the SOA platforms. To explain each characteristic, Table 2-5 provides a description or sample definition of each concept with references to its sources.

Table 2-5 - SOA characteristics

Characteristics	Sample definition	SOA References
Loosely coupled, modular and decomposable	Loosely couple: hide the implementation from the service user, encapsulating the implementation. Modular: a world of services being loosely coupled which can be flexibly combined to create dynamic business processes, new applications. Decomposable: breakable to finer grain functions.	(Erl, 2004; Joachim, 2011; Legner & Heutschi, 2007; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003; Schulte et al., 2008)
Technical standardisation	In order to guarantee seamless integration of applications in a heterogeneous environment, an SOA relies on interoperable, standard-based interfaces.	(Baskerville et al., 2005; Kontogiannis et al., 2007; Legner & Heutschi, 2007; Newcomer & Lomow, 2004; Papazoglou, 2003)
Abstraction from service implementation/ implementation independence	[…] concept of services, functional entities whose location and implementation are abstracted from the client or user, to allow the integration and communication of diverse and distributed technology domains (Luthria & Rabhi, 2009b). This will allow the service providers or	(Abelein et al., 2009; Baskerville et al., 2005; Erl, 2004, 2005; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003, 2008)

Characteristics	Sample definition	SOA References
	consumers to change with minimal effort.	
Comprehensive, uniform service specification	Services possess uniform interface descriptions and communicate by means of uniform protocols and data formats (Papazoglou, 2003). Otherwise, the capability to adapt the services (definitions or protocols) is required.	(Erl, 2004; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003, 2008)
Stable, managed service contracts	An SOA is a component model that interrelates the different functional units of an application, called 'services' through well-defined interfaces and contracts between these services (Walker, 2007). These services are discoverable and available for use.	(Baskerville et al., 2005; Erl, 2004, 2005; Mueller et al., 2007; Newcomer & Lomow, 2004; Papazoglou, 2008; Walker, 2007)
Interoperability using open, widely applied industry standards	Interoperability refers to the ability of a collection of communicating entities to share specific information and operate on it according to agreed-upon operational semantics (L. O'Brien, Merson, & Bass, 2007). SOA should, if possible, use open and widely applied industry standards.	((Baskerville et al., 2005; S. Kumar, 2007; Legner & Heutschi, 2007; Luthria & Rabhi, 2009a; Newcomer & Lomow, 2004; L. O'Brien et al., 2007; Papazoglou, 2003)
Autonomous	The logic governed by a service resides within an explicit boundary. The service has complete autonomy within this boundary and is not dependent on other services for the execution of this governance.	(Erl, 2005; L. O'Brien et al., 2007)
High service cohesion and weak logical coupling	Cohesion is the degree of the strength of functional relatedness of operations within a service.	(Baskerville et al., 2005; Erl, 2004; Newcomer & Lomow, 2004; Papazoglou & Yang, 2002)
Business standardisation	Technical standardisation has to be complemented by common semantics for business tasks and data.	(Baskerville et al., 2005; Kontogiannis et al., 2007; Legner & Heutschi, 2007; Newcomer & Lomow, 2004; Papazoglou, 2003)
Service granularity oriented towards business concepts	Services […] represent complete business functions, they are intended to be reused and engaged in new transactions not at the level of an individual program or even application but at the level of the enterprise or even across enterprises.	(Henningsson et al., 2007; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003)
Services composability/ choreography	Independent services with well-defined interfaces, which can be called in defined sequences to form business processes.	(Erl, 2004; Henningsson et al., 2007; Newcomer & Lomow, 2004; Offermann, Hoffmann, & Bub, 2009; Vitharana, Bhaskaran, Jain, Wang, & Zhao, 2007)

As presented in Table 2-5, many of the reported characteristics in the SOA literature are related and have overlaps, possibly due to the SOA studies being mainly exploratory studies. For example, ‘loosely coupling’ involves hiding the implementation from the service user by encapsulating the implementation and similarly ‘abstraction from service implementation’ involves hiding the location and implementation from the client to allow a diverse and distributed technology domain.

Considering the maturity of the IT literature in construct development, this study reviewed the IT literature, particularly the IT infrastructure and flexibility literature, and summarises relevant concepts applicable to this research in Appendix A.

One of the characteristics noted in the comparison between the two literatures is modularity. The SOA literature considers modularity as a world of services being loosely coupled which can be flexibly combined to create dynamic business processes and new applications.

As listed in Appendix A, the IT literature provides different definitions for modularity such as ‘integration of disparate and geographically distributed systems’ (G. D. Bhatt et al., 2010), ‘ability to add, modify, and remove any software, hardware, or data components of the infrastructure with ease and with no major overall effect’ (Byrd & Turner, 2000; Duncan, 1995) and ‘the degree of decomposition of an organization’s IT portfolio into loosely coupled *functionality discrete* subsystems that communicate through standardized interfaces’ (Nambisan, 2002).

Of these definitions, the last one (Nambisan, 2002) is closest to the definition sighted from the SOA literate by focusing on the loosely coupling of subsystems which are services in the SOA. Regarding the flexibility in the definition sourced from the SOA literature, flexibility is considered a higher order capability influenced by multiple factors (Duncan, 1995). Therefore, it will be addressed collectively by a number of SOA characteristics.

The IT literature (Fink & Neumann, 2009; Tiwana et al., 2010) already recognised that a modular system relies on *decoupling* (loosely coupling) of its modules and *standardisation* of its interfaces. This definition and conceptualisation could address the ‘technical standardisation’ in SOA literature, the ‘loosely coupled’ through the decoupling/loosely coupling dimension and the ‘stable managed service contracts’ partly through the standardisation of the interfaces. As per the description of ‘stable managed service contract’ in the SOA literature, the services are discoverable and available for use.

This aspect has not been addressed by modularity and requires further development during the data collection and analysis of the current research.

For the SOA platform characteristics, IT compatibility and IT connectivity are two key constructs reported in the IT infrastructure literature (G. D. Bhatt et al., 2010; Byrd & Turner, 2000; Duncan, 1995; Fink & Neumann, 2009). These two constructs have been reported as first-order constructs for integration (Byrd & Turner, 2000). Byrd and Turner (2000) considered integration as transparent access into all organisational platforms. This construct can assess the SOA platform capability in improving connectivity and interoperability between different service consumers and providers (Papazoglou, 2008). The data collection will identify any other characteristics that should be included for the SOA platform.

Appendix B summarises the comparisons between the two literatures and presents eleven SOA characteristics which is used as a priori constructs in this study.

2.4 Relevant Theories

This section reviews the theories that are adopted in the current research to provide insight on the relationships between the SOA characteristics and IT agility. The theory selected for the current research needs to provide a theoretical lens guiding the research. The selected theory enables the researcher to develop a theoretical framework which will be used as a "sensitizing device" (Gregor, 2006) at a high level to view the phenomena in a certain way. This approach is consistent with what Gregor (2006) considered as a subtype of the 'theory for explaining' which serves as a device for enlightenment (Gregor, 2006).

The next sections review two key theories: dynamic capabilities and the real options theory that are used in the current study. While these sections review the dynamic capability and real options, the justification of their adoption in the current research is addressed in the next chapter.

In addition to dynamic capabilities and the real options theory, a few other rival theories, which could partly explain the impact of SOA on IT agility, have also been reviewed and discussed in the next sections.

2.4.1 Dynamic Capabilities

2.4.1.1 Overview

The strategic management literature indicates a major stream of research on how an organisation resource and capabilities could contribute to organisation performance and sustained competitive advantage (Barney, 1991; Eisenhardt & Martin, 2000; Teece et al., 1997). This stream of research conceptualises business enterprises as portfolios of idiosyncratic resources (Wernerfelt, 1984). Competitive advantage can flow at a point in time from the deployment and use of valuable, rare, non-substitutable and inimitable resources and capabilities that might be heterogeneously distributed across firms (Barney, 1991; Conner, 1991). More specifically, resources that are valuable and rare can lead to the creation of competitive advantage. This advantage can be sustained over longer time periods so long as the firm is able to protect against resource imitation, transfer or substitution (Barney & Arikan, 2001).

Firms leverage two distinct strategic mechanisms to create idiosyncratic resources: resource picking and capability building (Makadok, 2001). Resource-picking mechanisms, which are codified into a 'resource-based view' (Barney, 1986; Makadok, 2001; Wernerfelt, 1984), are the main mechanism for the creation of economic rent through firms applying superior information and knowledge to gain advantage in selecting resources in the marketplace (Barney, 1986).

The capability-building mechanism, which has been codified into a 'dynamic-capability view' (Amit & Schoemaker, 1993; Makadok, 2001; Teece et al., 1997), however, is the firm's ability to build unique resources by leveraging their existing resources and capabilities (Teece et al., 1997). This ability involves integration, the building and reconfiguration of internal and external resources in creating higher-order capabilities to address rapidly changing environments (Teece et al., 1997). These capabilities are embedded in the organisation's social, structural, and cultural contexts, that make them comparatively more valuable and inimitable, thereby making them superior to resource-picking and holding as determinants of long-term performance (Eisenhardt & Martin, 2000).

The next section reviews different definitions available for the dynamic capabilities through which to gain further insight on this concept.

2.4.1.2 Dynamic Capability Definition

Despite the theoretical and practical importance of dynamic capabilities to a firm's competitive advantage (Eisenhardt & Martin, 2000; Teece et al., 1997), the dynamic capabilities literature contains contradictions and inconsistencies (Zahra, Sapienza, & Davidsson, 2006). Table 2-6 lists the key definitions of dynamic capabilities in the literature.

Table 2-6 - Key definitions of dynamic capabilities

Source	Definition
(Helfat et al., 2007)	The capacity of an organisation to purposefully create, extend and modify its resource base.
(Teece et al., 1997)	The firm's ability to integrate, build, and reconfigure internal and external resources to address rapidly changing environments.
(Eisenhardt & Martin, 2000)	The firm's processes that use resources – specifically the processes to integrate, reconfigure, gain and release resources – to match or even create market change. Dynamic capabilities thus are the organisational and strategic routines by which firms achieve new resources configurations as markets emerge, collide, split, evolve and die.
(Shaker A Zahra & G George, 2002)	Essentially change-oriented capabilities that help firms redeploy and reconfigure their resource base to meet evolving customer demands and competitor strategies.
(Winter, 2003)	Those that operate to extend, modify or create ordinary (substantive) capabilities.
(Zollo & Winter, 2002)	A dynamic capability is a learned and stable pattern of collective activity through which the organisation systematically generates and modifies its operating routines in pursuit of improved effectiveness.

Taking into account these various definitions of dynamic capability, we define dynamic capability as "the capacity of an organisation to purposefully create, extend, or modify its resource base" (Helfat et al., 2007). The "resource base" of an organisation includes tangible, intangible and human assets as well as organisational capabilities. The term "capacity" refers to the ability to perform a task or set of tasks in at least a minimally acceptable manner. This also implies that the function that a dynamic capability performs is a repeatable and relatively stable activity. Finally, the word "purposefully" indicates that dynamic capabilities reflect some degree of intent, even if not fully explicit (Helfat & Peteraf, 2009).

The current definition is precise enough to be meaningful, yet broad enough to allow researchers to examine more about the nature and origins of dynamic capabilities. This application of dynamic capabilities to a firm's resource base is consistent with prior definitions: 'organizational skills, resources, and functional competences' as per Teece (1997) and 'physical, human and organizational assets' as Eisenhardt and Martin (2000) have suggested. For Zollo and Winter (2002), dynamic capabilities act on ordinary capabilities which is already covered in the selected definition.

Regarding the purpose of dynamic capabilities, there are different goals considered in the definitions listed in Table 2-6. Zollo and Winter (2002) have considered the purpose of the dynamic capabilities as improvement of organisational effectiveness, whereas the majority have focused on the environment and market changes (Eisenhardt & Martin, 2000; Teece et al., 1997; Shaker A Zahra & G George, 2002). The definition selected for this study does not limit the dynamic capabilities to a specific purpose.

Dynamic capabilities are distinguished from substantive ('ordinary') organisational capabilities in that dynamic capabilities refer to the ability to change or reconfigure existing substantive capabilities or resources (Eisenhardt & Martin, 2000). For instance, new product development, knowledge transfer and strategic decision making are examples of dynamic capabilities which lead to new resources or capabilities (Eisenhardt & Martin, 2000).

The effect of dynamic capabilities on the creation, extension and modification of a firm's resource base varies with environmental dynamism. Eisenhardt and Martin (2000) argued that in moderately dynamic markets, routines in the form of dynamic capabilities are embedded in cumulative, existing knowledge. These routines involve analysis and implementation using the existing knowledge; hence, dynamic capabilities exhibit the properties suggested in the traditional research where effective routines are efficient and robust processes. In contrast, in high-velocity markets, dynamic capabilities rely extensively on new knowledge created for specific situations. Routines are purposefully simple although not completely unstructured to allow for emergent adaptation. Since new knowledge must be rapidly gained in each new situation, experiential activities such as prototyping, real-time information, multiple options and experimenting that quickly generate immediate knowledge replace analysis. In order to adapt to changing information, routines are iterative and cognitively mindful, not linear and mindless. This implies that effective routines are adaptive to changing circumstances in high-velocity markets. The price of that adaptability is unstable processes with unpredictable outcomes.

These routines go beyond the usual view of efficient and robust processes and include more fragile, semi-structured routines that are effective in high-velocity markets. The varying effect of dynamic capability in different environmental dynamism has been observed in empirical studies such as new product development, in which the impact of dynamic capabilities on functional competencies is positively moderated by environmental turbulence (Pavlou & El Sawy, 2006).

We note that response to a change does not require possession of a dynamic capability. For instance, ad hoc problem solving is one mechanism by which managers react to change (Winter, 2003). While a dynamic capability is a relatively stable activity, ad hoc problem solving is not routine, highly patterned or repetitious.

The dynamic capabilities concept can be applied to this research on multiple fronts: the conceptualisation of IT agility as a dynamic capability and as a theory to partially explain the impact of SOA on IT agility. The next chapter expands on how dynamic capability is applied to this research.

2.4.2 Real Options Theory

2.4.2.1 Overview

Real options, as a theoretical lens, views investments in organisational resources in terms of the investment's ability to generate future strategic choices with the assumption that there is an underlying source of uncertainty (Bowman & Hurry, 1993). The real options concept is based on the financial call option, which provides an investor with rights to future investment choices without a current obligation for full investment. Holding such rights typically involves small initial investment which enables the option holder to exercise the option when an opportunity arrives or to withdraw from further investment under unfavourable conditions. The value of holding an option increases due to the options holder's ability to exercise his/her option in exploiting an opportunity as opposed to those who do not hold those options (Amram & Kulatilaka, 1999). The more environmental change that is envisaged during the life of the option, the higher the additional value, because the option holder has an opportunity to exercise the option sometimes at an extraordinary profit. Therefore, in an options perspective, the expected variability and uncertainty represent potential future profit rather than a threat to current profitability (Anderson, 2000). This exercise becomes more economically advantageous especially when it is guided by the path dependencies in the form of prior learning, investment or experience (Amram & Kulatilaka, 1999).

In strategic management literature, the importance of options in creating competitive advantage has been reported (Bowman & Hurry, 1993; Copeland et al., 1994; Trigeorgis, 1996). Bowman and Hurry (1993) argued that options create inimitable resources that can provide organisations with sustained performance and competitive advantage. Copeland et al. (1994) suggested that real options provide flexibility to managerial decision making by allowing managers to use tangible and intangible assets in completely new or alternative ways in the future without having the obligation to do so. When making resource-committing decisions, assets can be arranged to enhance alternative uses. Trigeorgis (1996) drew a similar conclusion especially in uncertain investment environments, and developed real options perspective for analysing corporate budgeting and resource allocation decisions.

The real options theory has also been applied to the research field of IT. The next section provides a review of the IT literature concerning the application of real options theory.

2.4.2.2 Application of Real Options Theory to IT

In the area of IT, the real options concept helps managers conceptualise and assess the value of IT investments under uncertainty in two ways: as a decision tool and, qualitatively, as a way of thinking.

For the former, the real options concept provides a decision tool for IT-related services investments due to the high uncertainty character inherent in IT investment and their embedded real options (Fichman, 2004). For example, investment on infrastructure such as Intranet may be unattractive in terms of net present value. However, it provides a good foundation for possible e-commerce activities in future (R. L. Kumar, 2004). IS research on real options is therefore mainly concerned with the identification of various options in IT investments, and then in their framing as pricing problems, their valuation and the interpretation of the results. For example, Dos Santos (1991) applied real options theory to a two-stage IT investment, treating the first stage as an option to speedy implementation of the second. Benaroch and Kauffman (1999) presented the use of real options techniques in the context of a decision to delay the application of a banking ATM network. Chen et al. (2009) used the real options approach to propose an evaluation method for IT investments subject to public and private risks. Dai et al. (2007) used an analytical model based on real options to value IT infrastructure investment. Kim and Sanders (2002) provided a qualitative decision-support tool of strategic actions based on real options analysis of IT investment.

In addition to the real options providing a mechanism to determine the value of IT investments, they also provide a new way of thinking (i.e. options thinking) about how investments are evaluated and structured in respect to the managerial flexibility they provide under the conditions of high uncertainty and expenditure irreversibility (Fichman, 2004; Fichman et al., 2005). The essence of options thinking is more a philosophy than a science of precise quantification (Fichman et al., 2005). As Fichman (2005) stated “real option theory does not dictate use of any particular pricing models. It is instead an approach that recognises the value of management flexibility in investment evaluation”. “The bigger win comes from using real options concepts to actively create and extract the value of embedded options that can otherwise be difficult to see” (Fichman et al., 2005).

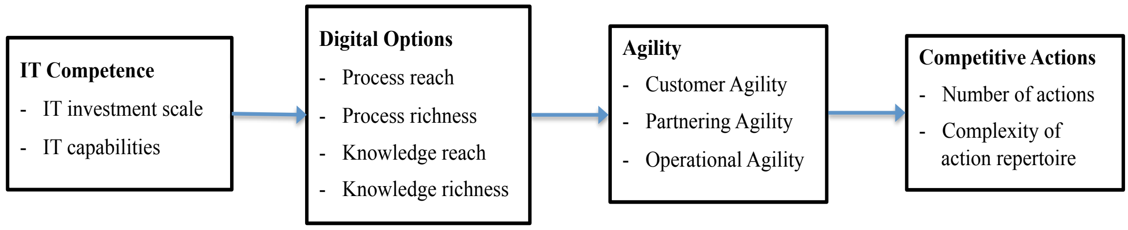


Figure 2-1 - Capability Building and Entrepreneurial [adapted from (Sambamurthy et al., 2003)]

Following this train of thought, Kogut and Zander (1992) expanded the notion of options to organisational capabilities and described a firm's knowledge and combinative capabilities as its strategic options. Sambamurthy and Bharadwaj (2003) drew on real options theory to conceptualise IT-enabled capabilities as options. Sambamurthy and Bharadwaj (2003), as shown in Figure 2-1, postulated a redefined role for which enables business agility. In this sense, they considered digital options as a set of IT-enabled capabilities in the form of digitised enterprise work processes and knowledge systems (e.g. customer capture, order fulfilment, supply chain, product innovation, manufacturing) for automating, informing and integrating the activities of an extended enterprise (Sambamurthy et al., 2003).

The concept of option has also been extended to the multi-phase adoption of information technology. Accordingly, an initial investment in IT creates “growth” options (Fichman et al., 2005; Tiwana, Keil, & Fichman, 2006), especially in the first phase of a multiphase implementation (Alfred Taudes, 1998). Growth options refer to the opportunity to create one or more additional related assets through follow-on investments after the initial investment (Tiwana et al., 2006). Growth options are most likely to be present on more innovative projects and on projects that implement a platform for future applications (Fichman et al., 2005). For example, in the ERP case examined by Taudes et al. (2000), the positioning investment was the baseline implementation of R/3 from SAP, which opened the door to follow-on projects related to the EDI, workflows and e-commerce. Based on the Black-Scholes Option Pricing Model, the estimated option value of follow-on projects exceeded the conventional net present value (NPV) estimates by a factor of four (A. Taudes et al., 2000). Those firms that defer investment in IT may not have quite the same claim to future benefits because of the time-compression diseconomies (Fichman, 2004).

2.4.2.3 Types of real options

Prior research (Benaroch, 2002; Copeland et al., 1994; Sambamurthy et al., 2003; Trigeorgis, 1996) suggested alterative taxonomies for real options.

As shown in Figure 2-1, Sambamurthy and Bharadwaj (2003) categorised the types of options that IT provides to a firm based on their outcome (Evans & Wurster, 2000) to four groups: digitised process reach, digitised process richness, digitised knowledge reach and digitised knowledge richness. The process reach dimension is concerned with options that provide firms with better integration internally and externally with customers, suppliers and partners to facilitate greater process participation among relevant stakeholders. The process richness dimension, however, improves the quality of information available to process participants by making it more timely, accurate, relevant and customised (Sambamurthy et al., 2003). Finally, the knowledge reach and richness improves the comprehensiveness, accessibility and quality of codified knowledge that is available to a firm (Sambamurthy et al., 2003).

Trigeorgis (1996) extended the options available for an investment and developed a taxonomy of real options based on the type of managerial flexibility that each option provides in the context of an investment. The proposed taxonomy includes six types of real options in respect to an investment: *stage*, *defer*, *alter scale*, *abandon*, *switch* and *grow* (Fichman et al., 2005; Trigeorgis, 1996). The stage, defer and abandon options provide flexibility to divide, postpone or terminate an investment based on the market conditions. In the stage option, the management can divide an investment into several distinct stages and make decisions in a stepwise fashion, whereas the defer and abandon options involve making the decision to postpone or terminate the investment (Trigeorgis, 1996).

The other three options: alter scale, switch, and grow deal with maintaining and growing an existing investment. The alter scale option provides management with the ability to expand investment if the market conditions are favourable, and to contract investment if the market conditions turn out to be unfavourable. The switch option provides the flexibility to redeploy an asset to provide a different purpose (switch use) or to replace and swap one technology for another (switch inputs) (Fichman et al., 2005). In the last option, the growth option, as already discussed, management's initial investment opens up follow-up growth opportunities (e.g. innovative products, new market access and strengthened core competencies).

Benaroch (2002) drew an important distinction between the first five option types which he called *operational options* and the sixth option type, *growth options*. *Operational options* pertain to discretionary actions that managers can make to reduce the potential for losses (usually) or increase the potential for gains (occasionally) on an existing investment and asset (Tiwana et al., 2006), for example, scaling an existing system. *Growth options*, in contrast, capture the possibility of building additional assets on top of the existing asset (Tiwana et al., 2006), for example, building new systems by reusing existing SOA services. In the case of operational options, the focus is on the potential for modification of the nature of the asset created by the base investment while, for strategic growth options, the focus is on assets enabled by the base investment (Tiwana et al., 2006). It can be argued that growth options are more valuable than operational options due to their potential to create new assets and new capabilities (Tiwana et al., 2006).

There are a few variations of the real options taxonomy for investments. For example, Copeland and Keenan (1994) identified grow, defer and quit as the major types of real options and Benaroch (2002) added *"explore"*, *"outsource"* and *"lease"* in the options for IT investments. The explore option provides an opportunity to pilot and prototype before the full investment (Benaroch, 2002). This option can be considered as a special type of 'stage' option in which the first stage of the investment is to prototype. The lease and the outsource options refer to sourcing of the assets, the former provides an option to hire the resource rather than purchase and the latter proposes a third party to provide the service (Benaroch, 2002).

The next chapter will use the reviewed real options to propose a set of options that SOA can provide to improve the IT sensing and responding capabilities.

2.4.3 Rival Theories

This section provides some of the rival theories applicable to the current study. The theories provided in this section are not used in the theorisation and development of the conceptual framework presented in the next chapter. They only present alternative viewpoints which might be used during the synthesis of a new model and theory developed at the end of the current research.

The list is only indicative, as the same phenomena can be observed through different theories and lenses with a focus on different aspects beyond the scope of the current research.

2.4.3.1 Complexity Theory

The complexity theory focuses on explaining the behaviour of complex systems and how the nature of a system may be characterised with reference to its constituent parts in a non-reductionist manner, for example. multinational corporations, or mass extinctions, or ecosystems such as rainforests, or human consciousness (Manson, 2001). Complexity theory, rather than being a single theory, consists of a number of theories concerned with complex systems gathered under the general banner of complexity research (Manson, 2001). This stream of research can be considered as three major divisions. “Algorithmic complexity", dealing with mathematical complexity theory and information theory, contends that the complexity of a system lies in the difficulty faced in describing system characteristics. “Deterministic complexity” deals with chaos theory and catastrophe theory, which posit that the interaction of two or three key variables can create largely stable systems prone to sudden discontinuities. “Aggregate complexity” concerns how individual elements work in concert to create systems with complex behaviour (Manson, 2001). The complexity theory suggests that systems with complex behaviour have emergent or synergistic characteristics that cannot be understood without reference to sub-component relationships (Manson, 2001).

Within the aggregate complexity concept, two different theories can be adopted to theorise the behaviour of SOA: evolutionary selection theory (Simon, 1962, 2002) and complex adaptive theory (Holland, 1992).

2.4.3.1.1 Evolutionary Selection Theory

The 'evolutionary selection' theory (Simon, 1962, 2002) and its derived 'modular systems theory' (Sanchez & Mahoney, 1996; Schilling, 2000) are concerned with the degrees of coupling between product and organisational architectures (Cabigiosu & Camuffo, 2012; Hoetker, 2006; Sanchez & Mahoney, 1996) and IS development (Benbya & McKelvey, 2006).

The theory of evolutionary selection has its root in the *nearly decomposable complex system* concept (Simon, 1962, 2002). Simon (1962), in his classic essay on the "architecture of complexity", argued that hierarchy is an organising principle of many complex systems which are essentially composed of interrelated subsystems that, in turn, have their own subsystems. Being a complex system, the system consists of a large number of components or modules with emerging characteristics that are dependent on its subcomponents relationships and cannot be predicted by considering the system components (Manson, 2001). Simon further defined a *nearly decomposable system* as one in which interactions among subsystems are weak (but not necessarily negligible). Simon (1962) provided a few examples of these systems in different disciplines, that is, the biological organism, "which is composed of organs, which are composed of cells, which contain organelles, which are composed of molecules, and so on" (Simon, 1962). System hierarchies can overlap enabling components to serve multiple systems, for instance, an individual may simultaneously be a component of a family system, a business corporation and several other community systems (e.g. the individual's church).

Based on this structure, in his *theory of evolutionary selection* Simon (1962) argues that decomposable complex systems evolve faster because they require less time to evolve by recombination and will undergo more diverse evolutionary experiments. He proposed that complex systems that evolve at a faster rate and with greater diversity are more likely to evolve to achieve better fit with their environment than those that do not possess these traits (Simon, 2002).

Simon's idea re-emerged again as the "loose coupling" concept (Weick, 1976) and, more recently, as modular product design (Sanchez & Mahoney, 1996; Schilling, 2000). In

'Modular Systems Theory' (Schilling, 2000), a system with no (strong) interdependencies between subsystems represents a perfectly modular (integral) system. The modular systems theory argues that greater modularity facilitates rapid changes in individual subsystems by lowering the need for coordinated changes in others (Schilling, 2000). Four system characteristics defined by the modular systems theory are: (a) can decrease overt coordination costs involved in the development and extension of a system between developers and management by providing an embedded coordination mechanism (Sanchez & Mahoney, 1996); (b) can decrease the effort required by a module developer to manage dependencies with the rest of the system, decreasing cross-module and module-to-platform systems integration costs; (c) can substitute for formal process control, thereby increasing module developers' autonomy; and (d) can decrease the need for knowledge outside module developers' task boundaries, engendering deeper specialisation (Tiwana et al., 2010).

The Evolutionary Selection theory can be applied to SOA, by considering SOA systems as decomposable complex systems. This provides a framework from which to conceptualise SOA characteristics and theorise its behaviour. The evolutionary selection and the modular systems theory explain the influence of decomposition and modularity, which are SOA core characteristics, on speed of adopting a change and the evolution rate of a system, main indicators of IT agility. The structural hierarchal decomposition and modularity could improve the survivability and adaptability of the overall system in a turbulent environment by limiting the impact of environmental disturbance to specific subcomponents (Orton & Weick, 1990).

While application of this theory can explain the behaviour of certain SOA characteristics (modularity and decomposability), it does not explain other characteristics such as scalability and connectivity. It also does not explain the impact of SOA on the sensing capability. On this basis, this theory is not considered a suitable theoretical lens for the current study.

2.4.3.1.2 Complex Adaptive Systems (CAS)

A complex adaptive system (CAS) consists of a complex system that not only self-organises, but also can direct its activity towards its own optimisation (Holland, 1992). Typical examples of CASs include ant colonies, immune systems, brains, markets and companies. The commonality across the cited examples is that they are composed of a large number of components (agents) that interact (Benbya & McKelvey, 2006). Holland (1992) defined CASs as exhibiting order creation generated from simple specifications. He further defined CASs as systems composed of interacting agents that respond to stimuli, and stimulus-response behaviour that can be defined in terms of "simple rules". Agents adapt by changing their rules as experience accumulates.

There is no definitive account of CAS theory but Volberda and Lewin (2003) summarised the academic and practitioner writing on complexity studies to propose three principles of co-evolving, self-renewing organisations: match co-evolutionary change rate, optimise self-organisation and synchronise exploitation and exploration. These principles provide the theoretical structure for the selection and encapsulation of key CAS concepts.

While CAS and its characteristics can be used to explain the evolution of SOA, CAS adoption, however, requires the inclusion of technical and organisational resources together as an agent. This conceptualisation is required as the SOA technical characteristics have no capacity to be self-organising.

Considering the scope of the current research, this theory cannot explain the behaviour of SOA and how it contributes to IT agility.

2.4.3.1.3 Other Theories

One theory that could be considered for the current research is the absorptive capacity theory (Cohen & Levinthal, 1990), which refers to the ability to identify, assimilate, transform and apply external knowledge. Through certain activities such as research and development (R&D), a firm develops collective knowledge about certain areas of markets, science and technology and how those areas relate to the firm's products and services (Cohen & Levinthal, 1990). This knowledge base facilitates the firm's ability to identify and value external knowledge. Over time, firm develops processes, policies and systems that facilitate sharing and transferring knowledge internally which provide the ability to assimilate and transform external knowledge (Cohen & Levinthal, 1990). This process creates commercial and knowledge outputs, thereby increasing firm performance (Shaker A Zahra & Gerard George, 2002). While this theory has many similarities with dynamic capabilities, it only focuses on managing knowledge. This limits its focus and does not explain how resources and capabilities can be mobilised to respond to information system changes, as highlighted in the IT agility definition.

2.5 Conclusion and Knowledge Gap on the impact of SOA on IT Agility

This section concludes the literature review by first justifying the significance of IT agility as the subject of the current study. It then motivates the study of SOA as a technology that can overcome many of the limitations of information systems, which inhibits agility. While the review of SOA literature presents promising results in improving the IT responding capability, there are also knowledge gaps, particularly concerning the effect of SOA on the IT sensing capability. Additionally, conflicting reports of complexity generated by SOA raise questions on how SOA affects IT agility.

Starting with the significance of IT agility, uncertainty and unpredictability caused by factors such as globalization, technology innovations and outsourcing have forced organisations to improve their ability to adapt to unexpected changes in an effort to achieve and maintain their competitive advantage. This has positioned agility as a crucial factor for a firm to survive and thrive in an uncertain and turbulent market (Ganguly et al., 2009). Considering the heavy reliance of organisations on information systems, IT agility is becoming a critical capability of firms and an enabler of business agility (Overby et al., 2006; Sambamurthy et al., 2003). IT agility provides an organisation with the ability to identify needed changes in its information systems and to swiftly implement the required changes in order to support the business to survive and thrive in an uncertain environment.

Enhancing IT agility requires a focus on both sensing and responding capabilities. These two capabilities together enable an organisation to proactively seize opportunities or to survive threats (Sambamurthy et al., 2003; Yang & Liu, 2012).

The review of the IT agility antecedents has indicated that IT agility can be affected by four groups of resources and capabilities: people-related factors such as skills, process-related factors such as agility of processes, structure-related factors such as governance structure (centralised/decentralised decision making) and, finally, technology-related factors such as the flexibility of information systems. The IT agility antecedents' literature has highlighted that there are conflicting reports regarding the impact of technology on agility. On one front, technology has been reported to have a positive impact on agility (Oosterhout et al., 2006; Park & El Sawy, 2012; Tiwana & Konsynski, 2010), on the other front, it has been considered an inhibitor of agility. Factors such as complexity of information systems, IT slow time to market (longer to implement requirement changes), inflexibility of systems, rigid IT architecture and inflexibility to support external integration have been identified as some of the factors impacting on agility (Allen & Boynton, 1991; Oosterhout et al., 2006; Tallon, 2008). This conflict sets the course for the current research to study the impact of technology (the fourth antecedent) on IT agility. To achieve this goal, the current study focuses on SOA and the information systems implemented using this style of architecture. SOA, as a new computing paradigm, has promised to overcome many of the above limitations such as inflexibility and complexity of information systems by offering a higher degree of standardisation, uncomplicated interoperability, reusability and high flexibility (Mueller et al., 2007).

In this new style of architecture, systems are broken down to individual services: self-describing and platform-agnostic modules that support rapid composition of new distributed applications (Papazoglou, 2003). Services and SOA through their focus on characteristics such as business-driven, standard-based, loosely coupled and highly modulated characteristics have been able to improve software development and integration capabilities (Abelein et al., 2009; Baskerville et al., 2005; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c), IT operational maintenance (Abelein et al., 2009; Haines & Haseman, 2009) and complexity reduction (Legner & Heutschi, 2007; Yoon & Carter, 2007).

Conversely, there are also reports of SOA increasing IT complexity due to additional modularity and dependency (Baskerville et al., 2005; Kokko et al., 2009; Luthria & Rabhi, 2009c). This contradictory effect of SOA on system complexity is unexplored in the existing studies of SOA and agility. Considering that complexity is a structural property of a system, conceptualisation of SOA through its characteristics and the review of how different SOA characteristics affect and contribute to IT agility can provide insight on the issue of complexity.

Additionally, while the IT is the first beneficiary of the SOA in improving its processes, the extant SOA studies (J. Schelp & Aier, 2009; Yoon & Carter, 2007) have mainly focused on business agility and have kept the IT agility unattended. As recognised in the agility literature, business agility can be impacted by many non-IT factors such as organisation structure, teamwork, decision-making, governance structure and business network structure (Yang & Liu, 2012; Yusuf et al., 1999; Zhang & Sharifi, 2007). Since SOA benefits are mainly IT-related and are tuned to improve IT responsiveness, the study of IT agility rather than business agility can provide a better measurement for SOA's impact on agility.

In also considering the exploratory nature of the previous studies (J. Schelp & Aier, 2009; Yoon & Carter, 2007), they *do not explain* which SOA characteristics are more significant in achieving IT agility.

The other unexplored area is the impact of SOA on individual IT sensing and responding capabilities. As already observed in the business agility context (Overby et al., 2006; Roberts & Grover, 2012), the ability to sense opportunities has a stronger impact on the outcome than the responding capability. Understanding how SOA can improve the IT sensing capability can provide interesting insights.

To summarise, this research is focusing on the following gaps:

- Lack of clear understanding on the SOA characteristics that have a significant impact on IT agility, particularly the IT sensing capability.
- Lack of clear understanding on the relationships between the SOA characteristics and IT agility through its two dimensions: the sensing and responding capabilities. This is specially magnified considering the conflicting reports concerning the impact of SOA on system complexity which is an agility inhibitor.

To address these gaps, this research conceptualises SOA using its characteristics. Despite the fact that there are many studies in the SOA literature reporting SOA characteristics and benefits, the question of how SOA should be conceptualised needs to be answered first before assessing the SOA value. This is due to the SOA concept being relatively new and there being a need to conceptualise it (Joachim, 2011; J. Schelp & Aier, 2009).

To assess and explain the relationship between SOA and IT agility, the two theories of dynamic capabilities and real options theory are adopted. The adoption of these two theories creates a conceptual framework which allows further exploration of SOA impact on IT agility in the course of the current research.

Chapter 3. Conceptual Framework

This chapter constructs a conceptual framework based on the literature review presented in the previous chapter and the options theory and the dynamic capability theories.

As highlighted in the literature review, the SOA concept is not a well-conceptualised and well-theorised topic. This limits the ability of this research to construct a theoretical model and to propose hypotheses on the effect of SOA on IT agility. Therefore, the current research follows Teece's approach (2007) for the creation of the conceptual framework, and to avoid the above limitation. According to this approach, a conceptual framework endeavours to identify classes of relevant variables and their interrelationships (Teece, 2007) and is used as a "sensitizing device" at a high level to view the phenomena in a certain way (Gregor, 2006). The framework is then used to develop relevant theoretical propositions in respect to the effect of SOA on IT agility.

This chapter has the following structure. Section 3.1 positions the use of real options theory in the current research, followed by section 3.2 which theorises IT agility as a dynamic capability with two sensing and responding capabilities. Section 3.3 suggests explains the use of knowledge-based options as capabilities that improve the IT sensing and responding capabilities.

Drawing on the propositions made in section 3.3, section 3.4 identifies some of the knowledge-based and process-based options that SOA can generate and suggests a link between the SOA capabilities and the process-based and knowledge-based options.

Finally, section 3.5 provides a summary of the chapter along with the constructed conceptual framework and the propositions of the study. It also discusses how the propositions and the conceptual framework together direct the current research in answering the research questions outlined in section 1.2.

3.1 Real Options Theory

As highlighted in section 2.4.2, the real options theory provides an opportunity to identify the value of IT in respect to the future choices and options that it can provide to organisations in sensing and responding to business opportunities (Sambamurthy et al., 2003).

As per the options theory, the value of options increases as the environment variability and uncertainty increase (Anderson, 2000). Such value creation is due to the further flexibility in the managerial decision making, which allows managers to use tangible and intangible assets in completely new or alternative ways in the future without having the obligation to do so. There are also suggestions that options create inimitable resources that can provide organisations with sustained performance and competitive advantage (Copeland et al., 1994).

In the current research, SOA is a style of architecture (Sprott, 2004) and architecture is concerned with rules governing the overarching structure and properties of the relationships among the systems and application (Tiwana & Konsynski, 2010). Such rules, when embedded in a system at the design time, deliver capabilities in the system that provides future options for use or extension. Architectural attributes such as system connectivity and system modularity are examples of such rules that can be *embedded* in a system upfront to provide future system extension or reuse options (Abelein et al., 2009; Baskerville et al., 2005; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c). These options can be *exercised* when suitable in future to deliver additional values. The embedded options have higher value in an environment with high variability and uncertainty (Anderson, 2000).

The application of real options theory in the current study is consistent with the previous studies (Fichman, 2004; Fichman et al., 2005; Alfred Taudes, 1998; A. Taudes et al., 2000; Tiwana et al., 2006) which considered a multiple-phase implementation of IT systems, such as ERP, as an option generator and other studies (Overby et al., 2006; Sambamurthy et al., 2003) that considered IT resources a source of generating digital options. Use of the real options theory in this fashion is consistent with what Fichman (2005) considered to be an ‘essence of options thinking’ (Fichman et al., 2005).

On the suitability of real options theory to explain ‘how SOA creates IT agility’, the current study follows Sambamurthy and Bharadwaj’s work (2003) who drew on real options theory to conceptualise IT-enabled capabilities as options. They proposed how IT can create Business agility through a set of Digital options such as knowledge and process options.

The real options theory assists in identifying the value of IT in respect to the future choices that IT can provide to organisations to sense and respond to business opportunities (Sambamurthy et al., 2003). Similarly, the link between SOA and RO follows the theoretical link proposed by Sambamurthy and Bharadwaj (2003) and Overby et al. (2006) in that IT resources are generators of knowledge and process options. Since SOA is a type of IT resource (Tiwana & Konsynski, 2010), the linkage between SOA and real options follow the same doctrine.

To further assess if real options theory is a suitable viewpoint for this study, three conditions to use real options concepts (Dixit, 1994) is assessed here. The three conditions are prerequisite to using real options concepts to structure the management of technology investments: uncertainty regarding net payoffs, irreversibility in project cost and managerial flexibility regarding how projects are structured (Dixit, 1994). All three conditions hold strongly for the current study. Net payoffs for SOA are typically quite uncertain because of multiple feasible implementation configurations and reuse in future implementations. The value of reuse is dependent on the future applications and its business value (Beimborn & Joachim, 2009; Hagel & Brown, 2001; Mueller et al., 2007). Regarding the irreversibility in project costs, the implementation costs involve developing software systems and implementing them within an organisation. Such software development and implementations are typically specific to an organisation, tightly coupled with organisational capabilities (Fichman, 2004). Finally, managers have considerable flexibility in how they approach IT investments. This flexibility can take two basic forms: flexibility in the process of delivering the new system, and flexibility in the result, that is, what the system offers for future uses and enhancements (Fichman, 2004). Flexibility in the former is promoted by managerial discretion in how projects are decomposed and staged, while flexibility in the latter is promoted by SOA flexibility to allow the organisations to use it as appropriate in multiple ways as well as by proactive steps to make systems more generic, multi-purpose, interoperable and scalable. Such flexibility, referenced as interpretive flexibility, allows organisations greater discretion in how they choose to appropriate a technology and adapt it over time which promotes managerial flexibility in the structuring and execution of options (Fichman, 2004).

The real options theory relies on two processes of embedding and exercising options (Tiwana & Konsynski, 2010). Considering that the current research focuses on the value of SOA and how SOA contributes to IT agility, the current study assumes the decision to embed an option has already been taken and that the organisation utilises the embedded option, when applicable.

On this basis, the researcher argues that the SOA characteristics generate real options that enhances the broader IT capabilities such as software development. Such capabilities, when used and executed, will facilitate the sensing and responding components of IT agility.

3.2 IT Agility as Dynamic Capability

The dynamic capabilities concept can be applied to this research on multiple fronts: the conceptualisation of IT agility as a dynamic capability and as a theory to partially explain the impact of SOA on IT agility.

Teece (2007) proposed a dynamic capabilities framework for firms exposed to rapid technological change and highlighted organisational and (strategic) managerial competences that can enable an enterprise to achieve competitive advantage and then semi-continuously morph so as to maintain it. In the dynamic capability framework, Teece (2007) disaggregated dynamic capabilities into the capacity: (1) to sense and shape opportunities and threats; (2) to seize opportunities; and (3) to maintain competitiveness through enhancing, combining and reconfiguring the firm’s intangible and tangible assets.

Based on the IT agility definition used in the current study, agility captures the sensing and seizing components of dynamic capabilities in Teece’s (2007) framework. More specifically, IT agility refers to the degree to which IT is able to sense and respond quickly to IT system change opportunities to support the business to survive and thrive. Hence, the current study conceptualises IT agility as a dynamic capability enabled by key organisational capabilities.

While the sensing and seizing capabilities in Teece’s (2007) dynamic capability model are applicable to IT agility, Teece’s (2007) third component – reconfiguration to sustain competitiveness – addresses the sustainability of IT agility over time within an organisation, which is outside the scope of the current study. The reconfiguration of capabilities and resources is required to effectively adapt to and evolve with environmental changes, whether they are threats or opportunities (Teece, 2007). Moreover, reconfiguration requires the continuous alignment and realignment of specific tangible and intangible assets. There are numerous mechanisms by which firms reconfigure their capabilities, including capability substitution, capability evolution and capability transformation (Lavie, 2006). While an agile firm may excel at capability reconfiguration, agility characterises only deliver the firm's ability to sense and respond to environmental change (Sambamurthy et al., 2003).

The next section reviews the antecedents of IT agility, by reviewing the factors that affect IT agility sensing and responding capabilities. It then uses the options theory and the digital options concept (Sambamurthy et al., 2003) to position the options as an enabler of IT agility.

3.3 Options Enhancing IT Agility

This section applies the two theories discussed above and proposes how knowledge-based options and process-based options can improve the IT sensing and responding capability.

The next subsection first defines the knowledge-based options as a set of SOA-enabled knowledge-based capabilities which improve the firm's ability to leverage and create knowledge required in detecting changes necessary in its information systems. It then uses the literature to propose that *the IT sensing capability is positively affected by the knowledge-based options that SOA offers*.

Similarly, the latter subsection defines the process-based options as a set of SOA-enabled process-based capabilities which improve the IT's flexibility and coordination required in responding to changes necessary in its information systems. Relying on the above theories and the literature, it then proposes that *the IT responding capability is positively affected by the process-based options that SOA offers.*

3.3.1 Sensing Capability affected by knowledge-based options

Sensing new opportunities is scanning, creation, learning and interpretive activity (Teece, 2007). To identify and shape opportunities, firms must constantly search and explore technologies and markets, both local and distant (Benner & Tushman, 2003; March, 1991). Sensing activities involve investing in research activities, probing customer needs, understanding latent demand and assessing likely supplier and competitor responses (Teece, 2007). Such activities in addition to individual's cognitive and creative capabilities require extant organisational knowledge and organisational processes such as R&D (Teece, 2007) and market intelligence (Overby et al., 2006). Four groups of processes have been recognised as participating in the sensing of market and technological opportunities. These include: (i) ‘processes to direct internal R&D and select new technologies’, (ii) ‘processes to tap supplier and complement innovation’, (iii) ‘processes to tap developments in exogenous science and technology’ and (iv) ‘processes to identify target market segments, changing customer needs and customer innovation’ (Teece, 2007). Sense of market and technology changes through these processes would also indicate if their supporting information systems need to change. For instance, processes to identify and select new technologies directly cause changes in the ‘information systems’.

The above four process groups not only leverage the existing knowledge within the organisation, they also create new knowledge by tapping into the external sources such as suppliers, customers and exogenous science and technology. This suggests a firm’s ability to sense market opportunities depends on its ability to create and leverage knowledge (Haeckel, 1999; Overby et al., 2006; Sambamurthy et al., 2003). Access to high-quality codified knowledge enhances firms’ sensing capability by providing managers with high-quality information about the state of their business which helps them to identify emerging opportunities and threats (Overby et al., 2006). Therefore, from the options perspective, creation of such knowledge is an option for IT that can be appropriately utilised to sense the opportunities in improving its information systems.

The consideration of knowledge as an option is based on the choice that the knowledge provides to the organisation to utilise knowledge or to leave it unused (Overby et al., 2006; Sambamurthy et al., 2003). Such conceptualisation of knowledge as an option is consistent with existing research. For example, Kogut and Zander (1992) consider firm’s knowledge and combinative capabilities either as its strategic options, or the concept of digital options as a set of IT-enabled capabilities in the form of digitised enterprise work processes and knowledge systems (Sambamurthy et al., 2003).

The current study, therefore, defines knowledge-based options as *a set of SOA-enabled knowledge-based capabilities which improve the firm’s ability to leverage and create knowledge required in detecting changes necessary in its information systems.*

As suggested above, the knowledge-based options can be considered capabilities due to their enabling power to deploy relevant resources using organisational processes in order to effect a desired end (Amit & Schoemaker, 1993) which is to improve the IT-sensing capability.

Holding knowledge-based options will assist IT to detect the opportunities with its information systems. For example, rich knowledge related to internal processes can help managers to identify operational deficiencies such as fulfilment problems (Overby et al., 2006) and to propose changes to the process and systems to address the issues in a timely manner. ‘Sense-making’ and ‘perspective sharing’ are other examples of knowledge-based options. ‘Sense-making’ and ‘perspective sharing’ are enabled by advanced knowledge technologies and collaborative tools for knowledge sharing (Sambamurthy et al., 2003) or service management and monitoring provided by the SOA platform, which provides insight on the performance of business processes (Papazoglou, 2008). Such options provide insight into the changes required in the processes and the supporting information systems.

On this basis and considering that the current study is concerned with the SOA impact on IT agility, the following proposition is put forward:

> **Proposition (P1):** *The IT sensing capability is positively affected by the knowledge-based options that SOA offers.*

While the above proposition provides a baseline to study the relationship between SOA generated knowledge options and the IT sensing capability, the type of knowledge options which influence the IT sensing capability is not clear. For instance, although majority of firms that adopt SOA use resources with specialised SOA knowledge, the existing SOA case studies are silent on achievement of a high sensing capability in these firms.

3.3.2 Responding Capability affected by process-based options

Once an opportunity for innovation or competitive action is sensed, it must be addressed by mobilising the firm’s processes or services. In essence, a firm’s responding capability is basically its physical ability to act (Dove, 2001). Responding to opportunities involves maintaining and developing technological resources and complementary assets and then, when the time is right, investing heavily in the particular technologies and designs most likely to achieve marketplace acceptance (Teece, 2007).

Scholars propose that the firm’s ability to respond to market opportunities depends on the *coordination* and *flexibility* of its products and processes (Dove, 2001). For instance, by speeding the flow of information and reducing potential bottlenecks, well-coordinated organisational processes and routines will enable the firm to quickly respond to opportunities (Malone & Crowston, 1994). A similar effect has been observed in software development which has improved the collaboration and coordination of development teams internally and, with clients, has improved the agility of software development (Sarker & Sarker, 2009). The firm's response capability may also be enhanced by effective *coordination* with its channel partners (Mohr & Nevin, 1990). Therefore, the above argument suggests that the flexibility and coordination of processes (internally and with external partners) improve the IT responding capability.

Relying on the options theory, process-based options provide choices to the IT in the form of flexibility and coordination improvement in the IT processes. The process-based options can be exercised in IT processes such as information systems development and maintenance processes to improve the IT responding capability.

The current study, therefore, defines the process-based options as *a set of SOA-enabled process-based capabilities which improve the IT's flexibility and coordination required in responding to changes necessary in its information systems.*

The process-based options are still considered capabilities due to their enabling power to deploy relevant resources using organisational processes to effect a desired end (Amit & Schoemaker, 1993), which is to improve the IT responding capability.

Being a real option, the process-based options provide the IT with flexibility to use tangible and intangible assets and capabilities in the IT processes in completely new or alternative ways in the future without having the obligation to do so. Holding process-based options assists the IT to seize and respond to opportunities with its information systems.

Process-based options support firms' responding capability by improving the coordination both internal and external to the firm, which in turn will enhance responding capabilities such as product development, systems development and supply chain (Overby et al., 2006). Furthermore, IT enables the creation and sharing of boundary objects (Karsten, Lyytinen, Hurskainen, & Koskelainen, 2001) such as technical specifications and a technical grammar (Argyres, 1999). This in turn, will facilitate collaboration among individuals and firms (Overby et al., 2006). Process-based options, based on their impact, can facilitate greater process participation among relevant stakeholders and improve the quality of information available to process participants by making it more timely, accurate, relevant and customised (Overby et al., 2006; Sambamurthy et al., 2003).

On this basis and considering that the current study is concerned with the SOA impact on IT agility, the following proposition is put forward:

> **Proposition (P2):** *The IT responding capability is positively affected by the process-based options that SOA offers.*

On the assessment of the above proposition if it can be further developed and be falsifiable, there are cases reported in the SOA literature (Kokko et al., 2009) that process-based options, e.g. reuse, although available they do not contribute to the IT responding capability. The further development of the proposition can better explain such cases.

To summarise the above discussions, Figure 3-1 presents interactions between the knowledge-based and process-based options, in the form of option-like capabilities, and the IT agility components.

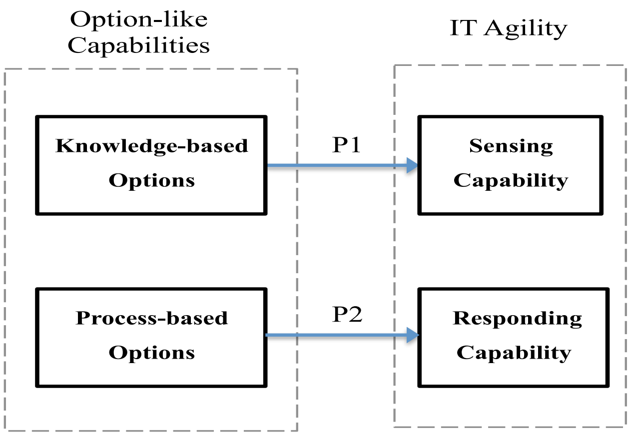


Figure 3-1 - Effects of option-like capabilities on IT agility

3.4 SOA Generating Options

This section analyses the existing real options and the SOA literature to achieve two objectives: (1) to develop a convincing argument for proposing SOA as an option generator, and (2) to propose a framework which can be used and refined during the data collection and analysis of the current research to identify the process and knowledge options embedded by the SOA.

Relying on the literature, the first subsection focuses on the knowledge-based options and proposes that *the knowledge-based options are positively affected by the SOA* *characteristics already embedded in the information systems.*

Similarly, the latter subsection makes two propositions based on the types of process options, which are operational and growth options.

3.4.1 SOA Creating Knowledge-based Options

As highlighted before, access to high quality codified knowledge enhances firms' sensing capability by providing managers with high-quality information about the state of the business which in turn will help them to identify emerging opportunities and threats (Overby et al., 2006). Hence, from the options perspective, creation of such knowledge is an option for IT which can be utilised when appropriate to sense the opportunities in improving its information systems. 'Sense-making' and 'perspective sharing' are examples of knowledge-based options that are enabled by advanced knowledge technologies and collaborative tools for knowledge sharing (Sambamurthy et al., 2003).

The current study acknowledges the fact that there are different types of knowledge: "migratory knowledge" which can be transferred via books, formulas and machines and "embedded knowledge" such as individual craftsmanship, know-how, and team-based knowledge (Badaracco, 1991). The current study mainly focuses on the migratory knowledge as primarily, the effect of technology relates to the migratory knowledge.

Sambamurthy et al. (2003) categorised the types of digitised knowledge options that IT provides to a firm based on their outcome, along the two dimensions of reach and richness (Evans & Wurster, 2000). Knowledge reach refers to the comprehensiveness and accessibility of codified knowledge that is available to a firm (Sambamurthy et al., 2003). Well-architected IT systems can assist firms in accessing, synthesizing and exploiting knowledge from a wide range of sources (Overby et al., 2006). IT also enhances knowledge richness by providing firms with high-quality information that is timely, accurate, descriptive and customized to the recipient (Sambamurthy et al., 2003). Information technologies such as decision support systems can help firms develop rich knowledge through real-time data monitoring, pattern recognition and strategic scenario modelling (Overby et al., 2006).

By relying on business process modelling, the role of SOA in creating and sharing knowledge lies in the multiple capabilities that it offers. SOA improves the process visibility to IT and the business which improves the knowledge sharing and knowledge creation between the teams (Legner & Heutschi, 2007; Yoon & Carter, 2007). Such knowledge, and especially the business knowledge absorbed by the IT staff and technical knowledge offered to the business forms peripheral knowledge which is the knowledge outside a department's specialised domain (Tiwana & Keil, 2007).

Additionally, through its service management and monitoring, SOA infrastructure provides insight on the performance of business processes (Papazoglou, 2008) which provides IT with insight on the changes required in the processes and in the supporting information systems.

On this basis, the researcher puts forward the following proposition:

> **Proposition (R1):** *The knowledge-based options are positively affected by the SOA characteristics already embedded in the information systems.*

3.4.2 SOA Creating Process-based Options

As explained in previous sections, process-based options are a set of SOA-enabled process-based capabilities which improve the IT responding capability through flexibility and coordination of processes (Dove, 2001; Malone & Crowston, 1994; Mohr & Nevin, 1990; Sarker & Sarker, 2009). To identify the linkage between the SOA characteristics and the process-based options, this section uses the real options already reported for an IT investment to identify a suitable taxonomy for options that SOA can provide. Relying on these options and their contribution in creating flexibility and coordination in IT processes, this section proposes the linkage between SOA and process-based options.

Prior research (Baldwin & Clark, 2000; Benaroch, 2002; de Neufville, Hodota, Sussman, & Scholtes, 2008; Trigeorgis, 1996) suggest several taxonomies for process-based options. Trigeorgis (1996) proposed six types of real options in respect to an investment: *stage*, *defer*, *alter scale*, *abandon*, *switch* and *grow.* Benaroch (2002) drew an important distinction between the first five option types which he called *operational options* and the sixth option type, *growth options*. The *operational options* pertain to discretionary actions that managers can make to reduce the potential for losses (usually) or increase the potential for gains (occasionally) on an existing investment and asset (Tiwana et al., 2006); for example, scaling an existing system. *Growth options*, in contrast, capture the possibility of building additional assets on top of the existing asset (Tiwana et al., 2006), for example, building new systems by reusing existing SOA services. It has been argued that growth options are more valuable than operational options due to their potential to create new assets and new capabilities (Tiwana et al., 2006). Benaroch (2002) also added *"explore"*, *"outsource"* and *"lease"* in the options for IT investments. De Neufville et al. (2008) suggested that options such as abandon and stage are 'real options on projects' and distinguished them from 'real options in engineering systems', which are embedded in the systems at the design time. Options such as *splitting* into modules, *substitution* of one with another, *augmenting* by adding a new module (Baldwin & Clark, 2000) are examples of 'real options in engineering systems', which become available when system follows a modular design.

The above taxonomies have been applied to many studies in different contexts such as studies of IT infrastructure value (Dai et al., 2007); a service organisation and its conceptualisation as a set of strategic options (Su, Akkiraju, Nayak, & Goodwin, 2009); decisions to continue projects in escalation situations (Tiwana et al., 2006); IT project evaluation and management (Benaroch, 2002; Fichman et al., 2005); and human resource management (Sanyal & Sett, 2011).

The operational and growth options are process-based options due to providing the IT processes with options, for example, the growth option reduces the software development effort by providing reuse, hence improving the 'development and maintenance' processes as per Boynton et al.'s classification of IT processes (Boynton, Zmud, & Jacobs, 1994). Another explanation is the alternative routine that an option, for example 'growth through reuse', offers to IT. Such alternative routines in delivering and maintaining information systems position the above options as process-based options.

Applying the real options to the SOA concept suggests that SOA can improve the *operational options*, including 'switch', 'scale' and 'outsource' options, due to its modularity, service abstraction and composition (Erl, 2005; L. O'Brien et al., 2007; Papazoglou, 2008). The more modular, abstract and standard with uniform service specification that the services are, the more options will be available to replace them with alternative services (switch option). The distributed architecture and modularity of the SOA systems (Erl, 2005; L. O'Brien et al., 2007; Papazoglou, 2008) enable them to scale easier or to outsource services to a third party provided that they comply with the standard service definitions (Legner & Heutschi, 2007). On this basis, the researcher puts forward:

> **Proposition (R2):** *Operational options are positively affected by the SOA characteristics already embedded in the information systems.*

Boynton et al. (1994) identified IT management processes as consisting of eight groups of processes: project management, strategic management, services control, services planning, resource planning, IS services, IS function management, and development and maintenance. Benaroch's notion of operational options (2002) has overlapped with multiple processes, such as processes concerning the service planning, resource planning, IS function management and system maintenance.

Operational options are important in reducing the potential losses or increase potential gains on an existing investment and asset (Tiwana et al., 2006). Growth option provides higher value by creating new assets and capabilities. Die to its characteristics such as modularity, decomposability and standardisation, SOA provides options to reuse and compose new assets. On this basis, the following proposition can be put forward:

> **Proposition (R3):** *IT growth option is positively affected by the SOA characteristics already embedded in the information systems.*

The growth option is mainly concerned with the processes involved in software development such as project management and development (Boynton et al., 1994).

To summarise, Figure 3-2 presents the interaction between the SOA characteristics and the knowledge- and process-based options in the form of option-like capabilities.

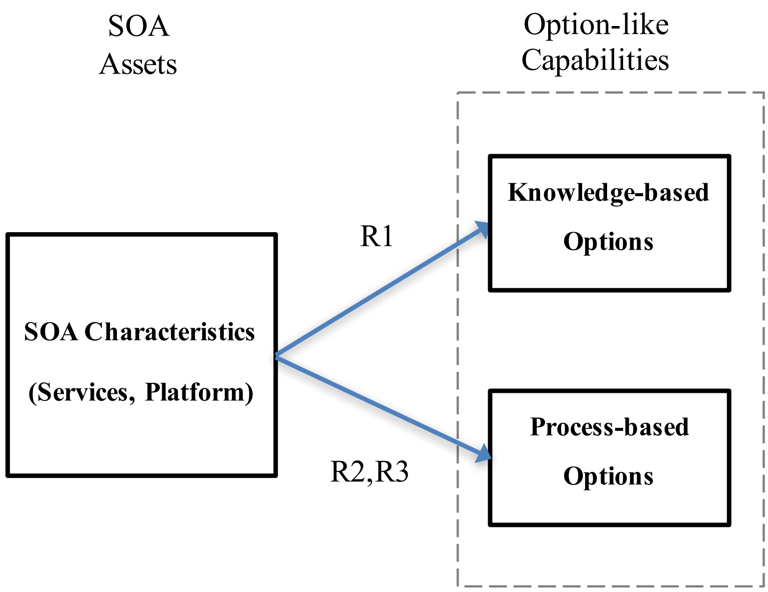


Figure 3-2 - Effects of SOA capabilities on option-like capabilities

3.5 Summary

By relying on two theories of real options theory and dynamic capabilities, Chapter 3 proposed SOA as an IT option generator with a positive effect on IT agility through its two sensing and responding capabilities.

SOA, as a style of architecture (Sprott, 2004), and therefore an IT asset, is concerned with rules governing the overarching structure and the properties of the relationships among the systems and application (Tiwana & Konsynski, 2010). These rules, when embedded in a system at the time of design, deliver capabilities in the system that provide future options for use or extension. These options can be exercised whenever suitable in future to deliver additional values. The embedded options have higher value in an environment with high variability and uncertainty (Anderson, 2000).

Figure 3-3 presents the conceptual framework constructed in the current chapter with three sets of concepts: SOA assets in the form of SOA services and platform characteristics, option-like capabilities created by the SOA characteristics and, finally, the IT agility components that consist of the sensing and responding capabilities.

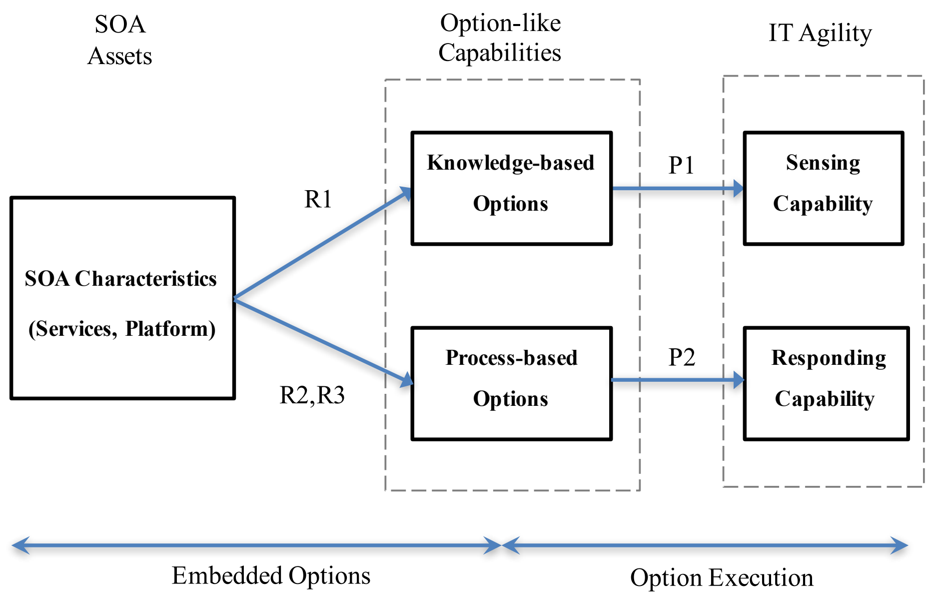


Figure 3-3 - Research framework of the study

The relationship between SOA characteristics and IT agility can be explained by both real options theory as well as dynamic capabilities. As per the real options theory, holding options provide flexibility for managerial decision making by allowing managers to use tangible and intangible resources in completely new or alternative ways in the future without having the obligation to do so (Copeland et al., 1994). Due to its design rules, SOA embeds options in the system, hence providing flexibility to the IT to use the assets and capabilities in completely new or alternative ways in future. Meanwhile, IT agility, as a dynamic capability, requires the integrating, building, and reconfiguring of internal and external resources to address rapidly changing environments (Teece et al., 1997); therefore, it can benefit from the use of resources that hold flexibility-like options, such as SOA.

Building on the two above theories, this chapter proposed that the relationship between SOA and IT agility is mediated by a set of knowledge-based and process-based options that SOA creates.

SOA characteristics such as service management, service monitoring and modelling improve the visibility and knowledge concerning the information systems (Legner & Heutschi, 2007; Yoon & Carter, 2007) and therefore create sensing-making options for the IT. Such options provide the IT with insight on the changes required in its supporting information systems.

Similarly, SOA characteristics such as decomposition and abstraction provide growth options to the IT processes to build new systems or maintain the existing systems through SOA created operational options.

On this basis, this chapter put forward:

> **Proposition (R1):** *The knowledge-based options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (R2):** *Operational options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (R3):** *IT growth option is positively affected by the SOA characteristics already embedded in the information systems.*

On the relationships between the SOA-enabled option-like capabilities and the IT agility, the extant literature showed that the knowledge-based options when used improve the firm's ability to leverage and create knowledge required in detecting changes in its information systems. (Haeckel, 1999; Overby et al., 2006; Sambamurthy et al., 2003). Similarly, the process-based options when used improve the IT flexibility and coordination required in responding to changes in information systems swiftly (Dove, 2001).

On this basis, this chapter proposed:

> **Proposition (P1):** *The IT-sensing capability is positively affected by the knowledge-based options that SOA offers.*
>
> **Proposition (P2):** *The IT-responding capability is positively affected by the process-based options that SOA offers.*

The current study through the process of assessing the validity of the proposed propositions extends the constructed conceptual framework. Such an extension involves the operationalisation of SOA characteristics and identification of knowledge-based and process-based options to which SOA contributes. The other activity involves verifying and extending the relationship between the proposed components in the framework. This will lead to a refined and concrete conceptual model.

Regarding the conceptual framework in Figure 3-3, it is important to acknowledge its limitations concerning recognition of all relationships that might exist between its components. Consideration of the main relationships at this point maintains the simplicity of the framework and avoid introducing antecedents for the options. Any additional relationships if exist will be identified and captured in the proposed conceptual models at the later stage of the research. The adopted approach is consistent with what Teece (2007) suggests for a conceptual framework: "A framework, like a model, abstracts from reality. It endeavours to identify classes of relevant variables and their interrelationships. A framework is less rigorous than a model as it is sometimes agnostic about the particular form of the theoretical relationships that may exist."

Assessment of the suggested propositions through the proposed theory development process will identify additional relationships and answer the research questions that the current study is striving to answer. This includes identification of SOA characteristics that contribute to the sensing and responding capabilities (RQ1 & RQ2).

> ***RQ1****: What are the SOA characteristics that affect the sensing component of IT agility?*
>
> ***RQ2****: What are the SOA characteristics that affect the responding component of IT agility?*

Answers to these questions will highlight the service and platform characteristics that contribute to IT agility. These questions are answered by operationalising SOA through its characteristics and then assessing their impact on IT sensing and responding capabilities through the option-like capabilities. Operationalisation of SOA will capture concepts such as the composition of services, service autonomy and decomposition of services as identified in the SOA literature.

The other two questions of the current study (RQ3 and RQ4) strive to explore the underlying mechanism that SOA facilitates the IT sensing and responding capabilities.

> ***RQ3**: How do the SOA* embedded characteristics *facilitate the sensing component of IT agility?*
>
> ***RQ4**: How do the SOA* embedded characteristics *facilitate the responding component of IT agility?*

These questions will be answered as the propositions are assessed and the options and their relationships with SOA and IT sensing and responding capabilities are uncovered.

Answers to the research questions through assessment of the proposed propositions and further development of the conceptual framework accomplish the objective of the current research which is to study how SOA as a style of architecture affects IT agility.

Regarding the reported issue concerning the additional complexity of SOA, the current study, by identifying the characteristics that contribute to IT agility and the mechanisms involved in their interactions, uncovers the complexity paradox of SOA. Additionally, such finding contributes to the fine-tuning of SOA implementation and consequently the possible reduction of system complexity through the reduction of parts (other characteristics) and their relationships.

Chapter 4. Research Methodology

This chapter focuses on the methodology to execute the research and identify answers to the research questions. It first covers the research method of the research and justifies the use of case study and qualitative research. It then discusses the data collection method and explains how interviews have been conducted. Next, it discusses the data analysis strategy. The last section reviews the validity and reliability of the measurement as well as the biases involved in the process. This is followed by a discussion on approaches to minimise the bias as well as improving the validity and reliability.

4.1 Research Method

This section describes a suitable research method for this study in detail. Research methods are at the basis of the production of knowledge in any given field (Pinsonneault & Kraemer, 1993). All methods are based on underlying philosophical assumptions of what constitutes 'valid' research (Orlikowski & Baroudi, 1991).

This section first identifies the epistemological and theoretical perspective that best match the research, followed by the research method adopted for the study.

4.1.1 Philosophical Paradigms

Philosophical assumptions relate to underlying epistemological assumptions about knowledge and how it can be acquired and essentially guide selection of an appropriate methodology (Hirschheim & Klein, 1992). The scholars have classified the philosophical paradigms into four groups according to their underlying assumptions: positivistic, interpretive, critical and post-positivist (Guba & Lincoln, 1994; Orlikowski & Baroudi, 1991).

The epistemological and theoretical perspective taken in this research is post-positivism. To better position the post-positivism, it is helpful to review the positivist and interpretivist paradigms first.

The *positivist studies*, epistemologically, are premised on the existence of a priori fixed relationships within phenomena that are capable of being identified, apprehended and tested via hypothetic-deductive logic and analysis. Causal relationships, which are the basis for generalised knowledge, can predict patterns of behaviour across situations. Ontologically, positivist research assumes an objective physical and social world exists independently of humans. The researcher is seen to play a passive, neutral role, and does not intervene in the phenomenon of interest (Guba, 1990; Guba & Lincoln, 1994; Orlikowski & Baroudi, 1991).

Conversely, the *interpretivist studies*, epistemologically, consider people creating and associating their own subjective and intersubjective meanings as they interact with the world around them. Interpretive researchers thus attempt to understand the phenomena through accessing the meanings that the participants assign to the phenomena. Generalisation from the setting to a population is not sought, rather, the intent is to understand the deeper structure of a phenomenon which, it is believed, can then be used to inform other settings. Such an assumption, ontologically, rejects the possibility of an objective or factual account of events, instead seeking a subjective, relativistic and shared understanding of the phenomenon. Finally, the researcher can never assume a value-neutral stance and is always implicated in the phenomena being studied, because researchers' prior assumptions, beliefs, values and interests will always intervene to shape and interpret their investigations (Guba, 1990; Guba & Lincoln, 1994; Orlikowski & Baroudi, 1991).

Post-positivism shares many assumptions with positivism; however, it tries to overcome criticisms of the latter. Post-positivism, unlike positivism, believes that human knowledge is not based on unchallengeable, rock-solid foundations; rather, it is conjectural (Philips & Burbules, 2000). This gives this approach a subjective flavour by making knowledge abstract and not fixed. Unlike interpretivists however, post-positivists believe that social phenomena exist not only in the mind but also in the objective world. These phenomena are still lawful with mostly stable relationships among them (Miles & Huberman, 1994). Therefore, epistemologically, post-positivists believe in modified subjectivism. They say that 'objectivity' is a regulatory ideal, but can only be approximated by a human being. Ontologically, post-positivism is considered as *critical realism* which assumes that reality exists but can never be fully apprehended by researchers because of the latter's imperfect sensory and flawed intellectual mechanisms (Guba, 1990).

By post-positivism taking a middle position between positivists and interpretivists, it overcomes the criticisms that positivism faces, such as relevance, richness and theory-building ability (Guba, 1990; Guba & Lincoln, 1994). Guba (1990) identified four areas in which post-positivist research overcomes the criticisms of positivist research:

a) *The imbalance between rigour and relevance*: Trade-offs exist between internal and external validity ensuring rigorous internal validity hampers the generalisability of the findings. Post-positivist research, as it is carried out in a more 'natural setting', tries to achieve a balance between rigour and relevance.
b) *The imbalance between precision and richness*: While precision is very important, to achieve it, positivists rely heavily on statistical and mathematical methodology and ignore the richness of the data. Post-positivist research tries to tackle both precision and richness by including more qualitative methods (e.g. case study).
c) *The imbalance between elegance and applicability*: While achieving generalisability, positivist researchers ignore the locality and specificity and thus lose the scope of theory building. Theory building fits with post-positivism approaches where research is carried out in such a way that theory is the product rather than the precursor of the research.
d) *The imbalance between discovery and verification*: The positivist approach focuses on the verification (falsification) of the hypothesis rather than the discovery of theories and developing hypothesis. Post-positivism takes the middle position of a continuum where 'pure' discovery lies at one end and 'pure' verification lies at the other end.

Since this research studies the impact of SOA as a technology on the sensing and responding capabilities of IT, it requires a viewpoint that considers this phenomenon objectively independent of perceptions; however, the researcher still needs to be aware of the context in which the studied phenomenon resides in. Also, as already presented in the literature review and the conceptual framework, the current study requires the development of concepts and constructs in order to successfully present the SOA characteristics and the options that such characteristics provide IT in terms of sensing and responding to information system changes.

Use of post-positivism in the current study provides the opportunity to reveal pre-existing relationships as well as being open to new data emerging from the field. This assumes that the phenomena under investigation are relatively stable and exist objectively, which is consistent with a positivist view. However, unlike the positivist approach, post-positivism is not limited to examining pre-identified constructs and is designed to bring other constructs to the surface as well.

By applying the post-positivist approach, the current study identifies relevant constructs and propositions, which theorise the relationship between such constructs and the IT agility concept. The adoption of post-positivism to develop a theory is consistent with existing studies such as (Feller, Finnegan, Fitzgerald, & Hayes, 2008; Leidner, Pan, & Pan, 2009; Sarker & Lee, 2003; Webster, 1998), which explore and identify theoretical propositions in their studied context.

4.1.2 Case Study as the Research Method

Considering the selected epistemology (post-positivism) and that SOA is a contemporary phenomenon with its boundaries and effects not well understood, a positivist case study is a suitable research strategy that can be utilised in the current study (Benbasat, Goldstein, & Mead, 1987; Eisenhardt, 1989; Yin, 1994). Table 4-1 lists the requirements relevant to this study and how the case study as a research strategy addresses these requirements.

Table 4-1 - Study requirements

Study Requirements	Case Study Research	Reference
SOA is a new style of architecture and implementation of SOA might not be consistent in the industry. This was observed in contradictory reports provided in the literature.	A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context when the boundaries between the phenomenon and the context are not clearly defined and where multiple sources of evidence are used.	(Benbasat et al., 1987; Yin, 1994)
Not all organisations have implemented SOA to the same extent.	The case study emphasises the benefit of contextual understanding in its real-life context.	(Yin, 1994)
SOA characteristics and their influence on IT agility could involve a large number of variables.	The strength of the case study is its ability to capture reality in greater detail, and the analysis of a considerably greater number of variables than is possible with any other research method.	(Yin, 2009)

As Cavaye (1996) described, case study research can be used for various research aims:

> *Case study research can be used to achieve various research aims: to provide descriptions of phenomena, develop theory, and test theory. Case study research has often been associated with description and with theory development, where it is used to provide evidence for hypothesis generation and for exploration of areas where existing knowledge is limited* (Cavaye, 1996).

Case study research relies on multiple sources of evidence with data needing to converge in a triangulating fashion and achieving one result. The positivist case study also benefits from the prior development of a theoretical framework such as a priori specification of constructs to guide data collection and analysis (Yin, 2009).

Alternative research methods such as experimental research due to the lack of real-life context, or survey-based research due to limited sample size and the exploratory nature of the research, do not address these requirements. The selection of case study as the research strategy/approach for the current study is consistent with the three conditions that Yin (1994) suggested for the selection of case study in a research. Case study is a suitable research strategy when the researcher does not require control over behavioural events and needs to focus on contemporary events. In such studies, research questions usually start with 'how' or 'why' due to their explanatory nature or 'what' questions, when asked as part of an exploratory study.

The first two research questions of the current study are 'what' questions to explore and identify the SOA characteristics that affect the two IT agility components:

> RQ1: What are the SOA characteristics that affect the sensing component of IT agility?
> RQ2: What are the SOA characteristics that affect the responding component of IT agility?

The other two research questions are 'how' questions due to their exploration of explanations on the effect of SOA on IT agility. The use of options theory and dynamic capabilities as a priori theories provide the framework to view this phenomenon:

> RQ3: How do the SOA embedded characteristics facilitate the sensing component of IT agility?
>
> RQ4: How do the SOA embedded characteristics facilitate the responding

component of IT agility?

4.1.3 Research Type

The literature (Benbasat et al., 1987; Yin, 1994) has reported various classifications of research based on the outcome sought by the research. This section presents that the current study is a hypothesis-generating (Benbasat et al., 1987), exploratory (Yin, 1994) study. Following Yin's (1994) and Miles and Huberman's (1994) suggestion, the current study adopts an initial theoretical statement/proposition as the basis for hypothesis-generation.

Table 4-2 lists different research types and highlights their requirements and the potential outcome that they are trying to achieve at the end of the case studies.

Table 4-2 - Research types (Benbasat et al., 1987; McCutcheon & Meredith, 1993)

Research Type (Benbasat et al., 1987)	**Yin's Terminology** (Yin, 1994)	**Requirements**	**Potential Outputs**
Exploration	Description	May be no ***a priori*** theory when events are examined; important constructs are not likely to be defined.	Description of events and outcomes to allow other researchers to understand the processes and environment. May indicate the relative importance of some factors.
Hypothesis generation	Exploration	May have some ***a priori*** theory that is used to select case sites and the constructs to be examined.	Propositions developed, based on the observations at one or more sites. Operational constructs may be refined or developed; however, some of the measured constructs may not prove useful in the evolving theories (Eisenhardt, 1989).
Hypothesis testing (confirmation or disconfirmation)	Explanation	Theory and perhaps operational measures of constructs are defined well enough to allow hypotheses to be proposed prior to conducting site visits.	Indication of theory's validity; may involve assessment of reliability and validity of measures. Evidence that disconfirms one or more theories designed to explain events and outcomes in the case situations.

Comparing the research types listed in Table 4-2 with the status of the SOA literature, the current study is a hypothesis-generating (Benbasat et al., 1987), exploratory (Yin, 1994) study. This is due to the existence of multiple studies in the SOA literature which have provided empirical observation on the value of SOA. As highlighted before, the extant studies have not, however, presented a theoretical stand to explain the effect of SOA on IT agility. Lack of construct development for SOA and SOA-generated options limits the ability of the current study to suggest hypotheses and test such hypotheses. Hence, this study develops a middle-range theory (Eisenhardt, 1989) by generating propositions which are based on constructs developed in the current study. Such a theory-building (Eisenhardt, 1989), hypothesis-generating (Benbasat et al., 1987) approach provides insight on the SOA factors that are involved in the creation of IT agility through the options that they provide.

On this basis and relying on the approaches suggested for theory building (Dubé & Paré, 2003; Eisenhardt, 1989; Miles & Huberman, 1994; Yin, 1994) and conditions for a good theory (Eisenhardt, 1989; A. S. Lee, 1991), the current study extracted "existing theories" (dynamic capability and options theory) and "existing theoretical constructs" from the IT literature to conceptualise SOA. These constructs (a-priori specification of constructs) and their relationships are refined, dropped or extended during the data collection and analysis (Dubé & Paré, 2003; Eisenhardt, 1989).

Regarding the use of theory and establishing a theoretical framework, there are different viewpoints. Eisenhardt (1989) suggested not to have a theory initially in order to avoid bias. Yin (1994) suggested an initial theoretical statement/proposition which, as the cases are analysed and reviewed, are refined and updated. Miles and Huberman (1994) also suggested having a theoretical framework to be used only as a guide initially. As the data are analysed, the researcher constructs and refines the concepts and their relationships as presented in the data.

The current study followed Yin's (1994) and Miles and Huberman's (1994) tradition by applying the options and dynamic capabilities theories as a priori theories and creating a high-level conceptual framework and propositions theorising the relationship between different concepts. The current study operationalises the conceptual framework suggested in chapter Chapter 3 and investigates potential relationships between the SOA characteristics and the option-like capabilities to improve the sensing and responding capabilities of the IT.

In addition to the process and knowledge options, the researcher remains open to other factors that might surface during the data collection. Any potential factor beyond knowledge and process options are grouped as per the Lui and Piccoli's (2007) classification described in section 2.2.4 ("process", "people", "structure" or "technology").

Finally, it is important to acknowledge that while theory building from case offers advantages such as possibility of generating novel theories, testable and empirically valid theories, it also has weakness. The theories developed using cases can become overly complex due to the large volume of data cases deal with. The other weakness is that building theory from cases may result in narrow idiosyncratic theory (Eisenhardt, 1989). The current study by developing a theoretical framework and using that as a guide for the research reduced the risk of overly complex or narrow idiosyncratic theory (Miles & Huberman, 1994; Yin, 1994).

4.2 Research Design

A research design is a logical sequence that connects empirical data to research questions and the conclusion (Yin, 1994). A research design helps to operationalise the research in order to collect data and to analyse and answer the research questions (Yin, 1994). The current study adopts Eisenhardt's (1989) process of building a theory with a variation. The variation is the selection of a priori grand theories (options and dynamic capabilities) to guide the theory building process as recommended by Miles and Huberman (1994) and Yin (1994) to prevent the researcher from becoming overloaded with data and guides the data collection and analysis. The following table lists the steps involved in such study.

Table 4-3 - Process of building theory from case study research (Dubé & Paré, 2003; Eisenhardt, 1989; Miles & Huberman, 1994; Yin, 1994)

Step	Activity
Initial step	- Definition of research question - Definition of a priori constructs - A priori theory
Selecting Cases	- Specified population - Theoretical, not random, sampling
Crafting Instruments and Protocols	- Multiple data collection methods
Entering the Field	- Overlap data collection and analysis, including field notes

Analysing Data	- Within-case analysis - Cross-case pattern analysis
Shaping Propositions	- Iterative tabulation of evidence for each construct - Replication, not sampling, logic across cases - Search evidence for "why" behind relationships
Enfolding Literature	- Comparison with conflicting literature - Comparison with similar literature

The previous chapters satisfied the first three activities: research questions, a priori constructs and theories. The rest of this section and next chapters satisfy the other activities.

4.2.1 Unit of Analysis

In a case study research, the unit of analysis identifies what a case is (Yin, 1994). Generally speaking, it refers to the person, collective or object, that is the target of the investigation. Typical unit of analysis includes individuals, groups, organisations, countries, technologies, objects, and such (Bhattacherjee, 2012). As presented below, the current study has a main unit of analysis which is at the IT level and an embedded unit of analysis which is at the system change level.

To determine the unit of analysis, Benbasat et al.(1987) suggested visiting the research questions and the generalisation that the researcher hopes to achieve. The research questions in the current study are concerned with identification of SOA characteristics that affect IT agility and explanation of such impacts:

RQ1: What are the SOA characteristics that affect the sensing component of IT agility?

RQ2: What are the SOA characteristics that affect the responding component of IT agility?

RQ3: How do the SOA embedded characteristics facilitate the sensing component of IT agility?

RQ4: How do the SOA embedded characteristics facilitate the responding component of IT agility?

As highlighted in the review of the IT agility literature, IT agility is a capability which is applicable to the whole IT in an organisation. Therefore, the unit of analysis is set at the IT level to provide an opportunity to assess the agility of the IT in dealing with events (change factor) that trigger a change in the information systems landscape (Oosterhout et al., 2006; Yusuf et al., 1999). Such factors could be due to businesses requesting a change to the information systems landscape, technological opportunities such as emerging technologies or technological threats such as viruses and security hacks (Oosterhout et al., 2006).

While the unit of analysis is at the IT level, a change occurs at a lower level such as 'software development using SOA capabilities' or a 'SOA infrastructure change'. This level, called 'system change level', captures the context and the nature of a change, which needs to be studied and analysed in the review of a case.

The multiple levels for the unit of analysis is consistent with the Yin's (1994) embedded case study design which includes multiple units of analysis. In embedded designs, a study may include main and smaller units on different levels, which allows the search for consistent patterns of evidence across units, but within a case.

By applying the embedded case study design to the current study, system change level is the embedded unit of analysis, whereas IT is the main unit of analysis.

4.2.2 Case Selection

The next items that need to be addressed are the selection of the cases suitable for the current study. This section provides a justification for the use of multiple cases for the current study as well as the number of cases required for answering the research questions. It then proposes a set of criteria for the selection of cases. This section finally lists the selected organisations and their corresponding projects reviewed in this study.

An exploratory study such as the current research can benefit from multiple cases by comparing and contrasting different cases to refine the emerging concepts and relationships (Benbasat et al., 1987). Multiple-case design allowed cross-case analysis and investigations of SOA in its diverse settings to predict similar results (literal replication) and to produce contrasting results to extend the emergent theory (theoretical replication) (Yin, 1994).

Regarding the number of cases, there are no hard rules to select the number of cases for a study. While the number of cases needs to satisfy the saturation condition (marginal improvement to the theory when additional cases are analysed), a number between four and ten cases usually works well, unless there are mini-cases within each case (Eisenhardt, 1989; Yin, 1994).

Considering the above guidelines, the current study selected three organisations as three cases, each with four system change implementations as embedded cases (mix of projects and infrastructure changes).

This study also adopted a pilot study to improve the rigour of the current research (Dubé & Paré, 2003). The pilot provided the opportunity to refine the research design, the data collection plan, and research instrument (questionnaire) (Dubé & Paré, 2003).

The three cases and their subsequent embedded cases provided both literal replication (homogeneous case selected to predict similar results) as well as theoretical replication logic (heterogeneous sample selected to predict contrasting results) (Yin, 1994). The contrasting cases were at the level of SOA infrastructure and SOA services that the organisation has already deployed and which were available for use. Having contrasting scales of service orientation allowed the researcher to identify different SOA characteristics, their generated options and to finally extend the merging theory (Eisenhardt, 1989; McCutcheon & Meredith, 1993). The use of replication logic in the current study improved the generalisability of the emerging theory.

Table 4-4 - Case selection criteria

Case/Embedded Case	Criteria	Justification
Cases	Two cases from similar industry and one from a different industry. Medium to large size organisations (based on the size of their IT operation).	Selecting two cases the same or from similar industries will reduce the variability hence improves the literal replication to predict similar results. The third case from a different industry provides theoretical replication to bring variability. Size ensures the cases are comparable.
Case	Organisations in dynamic environment, with heavy reliance on IT to provide their services.	Agility concept is more applicable in dynamic environments. IT division needs to deal with uncertain events rather than planned events.
Embedded Cases	Embedded cases with different level of SOA use in the projects/ system change	Use of embedded cases within the same organisation can provide natural controls (A. S. Lee, 1989) for factors that are not the focus of the current study and could impact on IT agility, such as IT structure, software

		development and change management processes.
Cases	Contrasting level of SOA use in projects/system change	Generalisation of the findings by including the organisational context in the analysis, as per Yin's recommendation (Yin, 1994).
Cases and Embedded Cases	Selection of a mix of software development and system changes	To assess if the emerging theory is generalisable for both 'sense' and 'respond' to software development and system changes.
Case	SOA adoption is organisation-wide and has been implemented within the organisation for at least two years with systems already using SOA.	Ensure the services are available and a high level of organisational maturity in SOA. An organisation-wide adoption of SOA typically indicate that SOA benefits have already been observed and there are options already embedded for future use, e.g. number of available reusable services. Use of SOA in different activities such as system integration, business process automation, data exchange with external organisations

The sampling for the selection of the organisations were based on the criteria listed in Table 4-4, making the sampling method, non-probability and purposive sampling.

To select suitable cases, the researcher used a spreadsheet to identify, shortlist and track the organisations suitable for the current study. This process resulted in twenty-eight candidate organisations captured in the spreadsheet, grouped based on their industry.

By using the above criteria, the researcher shortlisted nine organisations from three industries. These are:

- Five Australian and International Banks
- Two Airline and Airport
- Two Insurance company

An invitation letter prepared and sent to each of the organisations to its CIO or head of Architecture, followed by a meeting with some of the organisations to discuss the research. From the organisations that the researcher approached, the Airline, the Airport and one International Bank agreed to participate in the research. As a condition to participate, the organisations' identity remains anonymous in this study.

Upon their agreement and receiving the ethical approval, the researcher coordinated a planning meeting with the organisation representative. In the meetings with their representative, the researcher and the organisation representative discussed the process, the candidate projects, access to relevant documentation and potential interviewees.

Table 4-5 lists the selected cases and how they satisfied the case selection criteria outlined above.

Table 4-5 - Selected cases

Organisations (Cases)	Sector	Embedded case	How met case selection Criteria
Australian **Airline** (AL)	Aviation	A380	Airport and Airline share the same industry. SOA use involved both system integration and service reuse (120 services) particularly for their core operational systems (flight planning and operational management) High number of IT projects and capital reflected a high number of change in the IT landscape. Loyalty team had further autonomy compare to the rest of organisation. Similarly, they required a shorter time to market.
		Blackberry	
		Emirates Integration	
		Loyalty team – Online channel	
Australian **Airport** (AP)	Aviation	AOS	Same industry as the airline SOA use involved both system integration and service reuse (30 services) High number of IT projects and capital reflected a high number of change in the IT landscape.
		PA	
		Internet	
		CUTE	
International **Bank** – Australian Entity (BK)	Banking	Digital Channel	Different industry to provide contrasting results. SOA use involved service reuse (250 services) and to lesser degree system integration High number of IT projects and capital reflected a high number of change in the IT landscape. Digital team worked very closely with the IT team and used agile methodology in their project. They required a shorter time to market.
		DGW	
		BSL	
		Living Super	
		Saving Maximiser	
		Bank in the Box	

As shown in the above table, the cases and their embedded cases have similarities and differences which provided literal and theoretical replications within each case and between the cases. The two teams of Airline loyalty and the Bank Digital teams both had shorter time to market compared to the rest of the organisations. They also had additional autonomy in their decision making concerning the system development due to them managing their own IT teams working for them.

The use of SOA in the organisations varied between service reuse and system integration. The Bank had high number of services, however limited system integration to facilitate exchange of data between its systems. Conversely, the airline and airport used SOA to expose the services available in their systems for service reuse as well as data integration between their systems.

4.3 Data Collection

This section reviews the data collection techniques suitable for the current study. Yin (1994) identified several sources of qualitative evidence in case study research including interviews, documentation, direct observation and physical artefacts.

Considering the exploratory nature of the current study, this research used interviews and documentation as its two main methods of data collection. These two methods are the most common data collection methods in the IS positivist case research (Dubé & Paré, 2003) and provide access to rich data embedded in the case. Use of multiple sources of evidence (triangulation) such as documentation and interview and multiple interviews in each organisation also improved the construct validity of the current research (Yin, 1994).

Following Miles and Huberman (1994) and Yin (1994), below provides how each technique applied in the current study:

- Interviews as the primary source
    - Semi-structured open-ended interviews with key informants
    - Field notes prepared for each interview
    - Each interview recorded, after the interviewee's agreement, and transcribed to improve the reliability.
- Documentation as the secondary source including:
    - Public documents to provide the background information concerning the organisation.
    - Internal documents to provide information concerning the IT, SOA adoption and projects.

The project documentation provided the background knowledge to prepare the researcher for the interviews. It also provided information concerning SOA adoption and how SOA affected the delivery of projects.

Among different styles of interview the semi-structured interview was deemed suitable for the exploratory nature of the current study (Yin, 1994) as it provides the interviewer with the ability to probe beyond the questions previously prepared.

Appendix D – Interview Protocol, the interviews followed an interview protocol, which included a set of questions derived from the conceptual framework developed in the previous chapter. The open-ended questions provided the researcher with further flexibility during the interview process to probe further or to identify related themes and concepts, which might play a role in the emerging theory. Identification of 'Continuous business and IT engagement' was an example of such theme that surfaced during several interviews.

To assess the validity of the interview instrument, the researcher conducted a pilot study to improve the questionnaire instrument. The next subsection explains the outcome of the pilot study. It then reports the conducted interviews, followed by the reviewed documents.

4.3.1 Pilot case description

The objective of the pilot study was to assess the suitability of the interview instrument in answering the research questions of the current study.

The researcher conducted the pilot case in Dec 2013 at the Airport. The Airport adopted SOA as a design paradigm in 2008. Several projects used SOA to facilitate integration between different systems and expose core flight related services to internal and external consumers. The key SOA technologies adopted at the Airport was IBM Enterprise Service Bus, which is the company's SOA platform.

To verify the interview instrument, the researcher conducted two interviews involving:

- The Airport SOA Support Engineer
    - The focus of the interview was on the characteristics of SOA and how these characteristics have improved IT sensing and responding capabilities.
- SOA Developer

- The focus of the interview was on the SOA characteristics and how they supported the projects to be more agile.

Based on the findings from the first interview, the instrument and the style of questioning was refined and improved in the next interview.

Regarding the first two research questions, the instrument elicited the SOA characteristics that had impact on the IT agility in the projects. The line of questioning was maintained, the wordings of the questions were refined to simplify the questions.

Regarding the last two research questions, the existing instrument was trying to explicitly ask about each characteristic, their corresponding options and their relationships with the IT agility. Use of this approach was not however successful and restricted the probing and relevance for the participant. To address this issue, the research applied a more holistic approach in questioning by asking the participant to describe the SOA characteristics that affected the IT agility and the consequence of the identified characteristics on the IT processes, knowledge within the organisation or project structure. The inclusion of the process, knowledge and structure dimensions allowed the researcher to further probe the options and consequences of the SOA on the organisation. Regarding the processes that SOA might have impacted, the researcher used the existing taxonomies of IT processes (Boynton et al., 1994) to assess the SOA impact.

Finally, the interview protocol has been designed to allow probing at both levels of embedded cases and the case level. Such embedded design, that is, multiple levels of analysis within a single study, will ensure that the research creates operational data, while it maintains its holistic view (Yin, 1994).

'Appendix D – Interview Protocol' includes the revised instrument and the original instrument included at the end of the section.

4.3.2 Conducted interviews

Twenty-nine interviews were conducted with the IT executives, business managers, architects and project delivery teams in the three selected organisations. Excluding the pilot, the researcher conducted all the interviews in 2014. All interviews were face to face in the organisations' offices.

The first studied case was the Airport and was conducted in the first quarter of 2014. The case study included seven formal interviews with the Head of IT, two IT managers, a project manager, SOA architect, SOA developer and SOA support engineer.

The second case study were conducted in the Airline Head Quarter during the months of April till July 2014. In addition to the face to face meetings, there were correspondence via email to clarify some of the points discussed during the interviews.

Finally, in the third round, nine interviews were conducted in the main Sydney office of the Bank between September 2014 and December 2014. Similarly, the sessions covered the main interviews and the follow up sessions with the interviewees.

As previously mentioned, the researcher's experience with the banking and aviation industries greatly helped him to understand the organisational context and processes. This also provided him access to the information faster and more effectively.

Table 4-6 - List of interviewees

Organisation	Department	Interviewee's position	Role in SOA projects	Duration of interviews
The Airport	IT & T	Head of IT	SOA Sponsor	0:40
	IT & T	IT Manager	Managing SOA platform	0:41
	IT & T	IT Manager	Consuming SOA services	0:44
	IT & T	Project Manager	Managing a few projects, which involved SOA	1:05
	IT & T	SOA Architect	Solution design and architecture for SOA projects	1:01
	IT & T	Engineer	SOA Support & project team member	1:16
	IT & T	SOA Developer	Developer	1:35
The Airline	IT & T	CTO	SOA Sponsor	0:21 0:18
	IT & T	IT Manager	Solution Architect and Enterprise Architect in three projects	1:27 0:18
	IT & T	SOA Governance (CoE) Manager	Governance and management of SOA	0:38 0:26
	IT & T	SOA Architect (J)	SOA Architect	0:53
	IT & T	SOA Delivery & Support Manager	Delivery & Support of SOA solutions	1:18

	Loyalty	Project Manager (B)	Managing SOA projects for loyalty team. Also manage Loyalty service layer.	0:58
	Business	Business user / Project manager	Business project manager	0:45
The Bank	IT & T	Head of IT	Responsible for all IT services	0:20 0:22
	IT & T	Architecture and Strategy Manager (Ben)	Planning and governance of SOA	0:32
	Business - products	Product manager	Project sponsor and service consumer	0:53
	Business – call centre	Business manager	Project sponsor and service consumer	0:35
	Digital	Project Manager	SOA project manager – scrum master	1:08
	IT & T	Development & Support Manager	Managing system development & support	1:03
	IT & T	PMO Manager	Managing project portfolio	1:07
	IT & T	Enterprise Architect (Wes)	Solution and enterprise architecture	2:10 0:19
	IT & T	Architect (Jon)	Solution and enterprise architecture	1:39 0:24

4.3.3 Reviewed documents

The current research gathered published and unpublished organisational documents, and project documents with the aim of obtaining a rich quality of data (Benbasat & Weber, 1996). Publicly available documents were collected and reviewed prior to conducting interviews, and interview questions reflected the outcomes of document analysis.

The document analysis focused on capturing information concerning the organisation and projects such as the organisation size and structure, project team size, project methodology and the way SOA has been applied in the projects. It also identified the following items concerning the SOA adoption in the organisation:

- How long has SOA been practised in the organisation?
- What was the major driver(s) and/or objective(s) for the SOA in your organisation (e.g. agility)? Was that driver/objective part of the company's business or IT strategy?

- In what areas the SOA has been used in the organisation (e.g. integration, composite application development, process automation, etc.)?
- What technologies are used in the SOA implementation for service development and deployment and SOA platform (integration middleware, ESB or tool used for SOA)?
- Use of services in the projects
- Use of SOA platform in the projects

Table 4-7 shows the list of documents accessed and used in the current research from each organisation.

Table 4-7 - List of documents

Organisation	Document	Content
The Airport	Company web site	General information about the background of the organisation, its products and structure.
	Organisation chart	Organisation structure
	ESB Solution Architecture	The setup and configuration of the SOA platform
	Projects Plans and status reports	Projects information about the project plan and execution
	Project change logs	List of changes occurred in the projects
	Projects solution architecture and design	Solution design for the projects
	Systems interface specifications	Interface specifications for each of the systems
The Airline	Company web site	General information about the background of the organisation, its products and structure.
	Organisation chart	Organisation structure
	ESB Solution Architecture	The setup and configuration of the SOA platform
	Service catalogue	List of services
The Bank	Company web site	General information about the background of the organisation, its products and structure.
	Organisation chart	Organisation structure
	DGW Solution Architecture	The setup and configuration of the SOA platform
	Projects Plans and status reports	Projects information about the project plan and execution (DGW, Bank in the box)
	Projects solution architecture and design	Solution design for some of the projects (DGW, Bank in the box)
	Service catalogue	List of services

	Benchmark TCO Applications 2014	Benchmarking of IT within Australian division of Bank compare to other branches.

4.4 Data Analysis

This section describes the data analysis utilised in this research. The data analysis approach followed Yin's suggested *explanation building* technique (2009), and the coding methods described by Miles and Huberman (1994). According to the Yin's (2009) explanation building technique, the initial theoretical propositions made in the research were reviewed and refined continuously as each case was analysed. Findings were then compared against the theoretical proposition. Yin (2009) noted that at the end of this process, the final explanation and propositions could be different from those that were initially outlined. The above replication strategy will allow similar and conflicting patterns in the cases to be identified which in turn will lead to the refinement of the drafted conceptual model or confirmation of the first model (Yin, 1994).

To analyse a case, the data collected from the case had to be reduced by the coding process (Dubé & Paré, 2003) which can be created in deductive and inductive ways (Miles & Huberman, 1994). In the *deductive way* of creating codes, a 'provisional list of codes' prior to fieldwork is prepared (a priori codes). These codes are based on prior models, research questions and research areas. However, as the research unfolds, the provisional list of codes may be amended, as some do not work and some decay (Eisenhardt, 1989; Miles & Huberman, 1994). Subsequently, pattern coding and memoing (i.e. making notes) will be used to identify patterns or repeatable regularities in the data set to build a conceptually coherent explanation of the research phenomenon (Miles & Huberman, 1994). The use of a priori definitions in the deductive approach prevents the researcher from becoming overloaded with data and guides the data collection and analysis (Miles & Huberman, 1994). This method was selected for the current study.

Since many of the SOA concepts applicable in the current study have already been recognised in the SOA literature, the current study takes a deductive approach by reusing the existing codes for the SOA (as presented in Appendix B – A Priori Constructs and Mapping to SOA Characteristics) to refine and provide a solid definition and conceptualisation for the SOA. On that basis, it uses the deductive coding methods suggested by Miles and Huberman (1994) which consists of descriptive coding and pattern coding.

To simplify the coding and to create a case database, the researcher used NVivo version 10 and 11 for data analysis. NVivo is a qualitative data analysis tool provided by QSR International. It supports text, images and multimedia information. The tool allows the users to tag, classify, sort, and arrange information and examine relationships in data represented in a variety of ways.

As the first step, the researcher set up NVivo with the identified a priori codes from the literature. As the case interviews and the document reviews completed, the researcher added the audios, their transcriptions and the relevant documents to the Nvivo and started the descriptive coding process. In descriptive coding, the focus is mainly on naming and classifying the concepts in the data, and involves summarising segments of data by assigning a tag or code to the text segments. Descriptive coding will became the basis for higher-order coding later (Miles & Huberman, 1994). As new codes were identified, the researcher associated them to the existing a priori nodes, or if they did not match the definition of the existing nodes, new nodes were created. To ensure the codes and concepts were consistent within the research, the definition of concepts and codes were maintained in the course of the data analysis. These definitions and codes were discussed and reviewed with the researcher's main supervisor and other scholars for reliability.

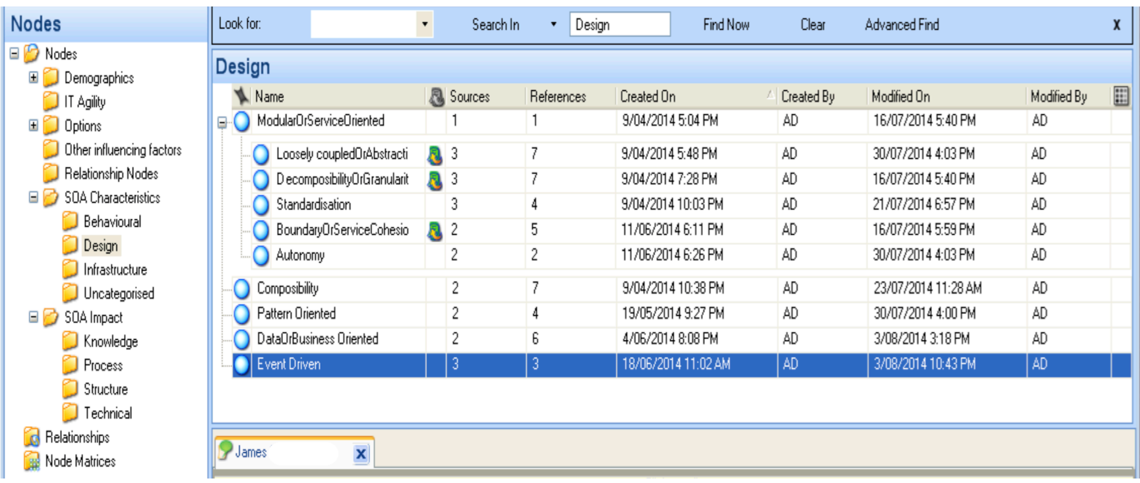


Figure 4-1 - Coding in NVivo 10

Figure 4-1 shows an early version of the NVivo project and the project structure when the a priori codes were added to the NVivo. As the data analysis progressed, memos (the green icons in the above figure) were added to capture the researcher's thoughts on how the codes must be refined and their relationships. The node names were not final and in some cases, alternative options were added to the node names, e.g. BoundaryOrServiceCohesion. Going through an iterative process, the codes merged, split or removed and their definitions refined.

Codes related to the case demographics, SOA characteristics, SOA-enabled options and IT agility were maintained in their own folders. While the interviews reflected the options that SOA created, the data analysis also revealed a set of impacts on the organisation across the four dimensions of knowledge, process, structure and technology. Example of SOA impact on the IT processes was the additional coordination required in the change management process and reduction of development cycles due to reuse. Additionally, there were other influencing factors identified in the data analysis that could somehow be important for the study. The identified factors were coded and maintained in the 'Other influencing factors'. Example of such factors were the continuous business and IT engagement observed in several embedded cases and the IT governance structure applied in certain cases.

Finally, the strength of each code in the studied cases was recorded using one of the following values: 'Absent', 'Weak', 'Average', 'Sufficient' and 'Strong'. The strength shows the overall adequacy judgement of the observed factor by the analyst in the studied case (Miles & Huberman, 1994), with 'Absent' reflecting non-presence of the factor and 'Strong' full availability of the factor. 'Sufficient' shows that the identified code was present in certain cases or groups to have sufficient impact, however not consistently in all projects or groups to provide its full impact.

The descriptive codes and their assigned strength facilitated the next stage of coding, which is pattern coding. Pattern coding brings the descriptive codes together (Miles & Huberman, 1994). The created patterns usually represent themes, causes, explanations and more theoretical constructs. This provides the opportunity to develop the constructs and examine relationships between the concepts (Miles & Huberman, 1994). Table 4-8 provides an example of how pattern coding applied to identify common themes. As the researcher identified and refined the SOA characteristics and summarised the data in matrix displays (Miles & Huberman, 1994), they revealed themes such as 'structure-centric'. As shown below all the descriptive codes in the example below are associated with the structure of a service or structure of services together in a system, which present themselves as being structure-centric.

Table 4-8 – Development of Structure-centric construct

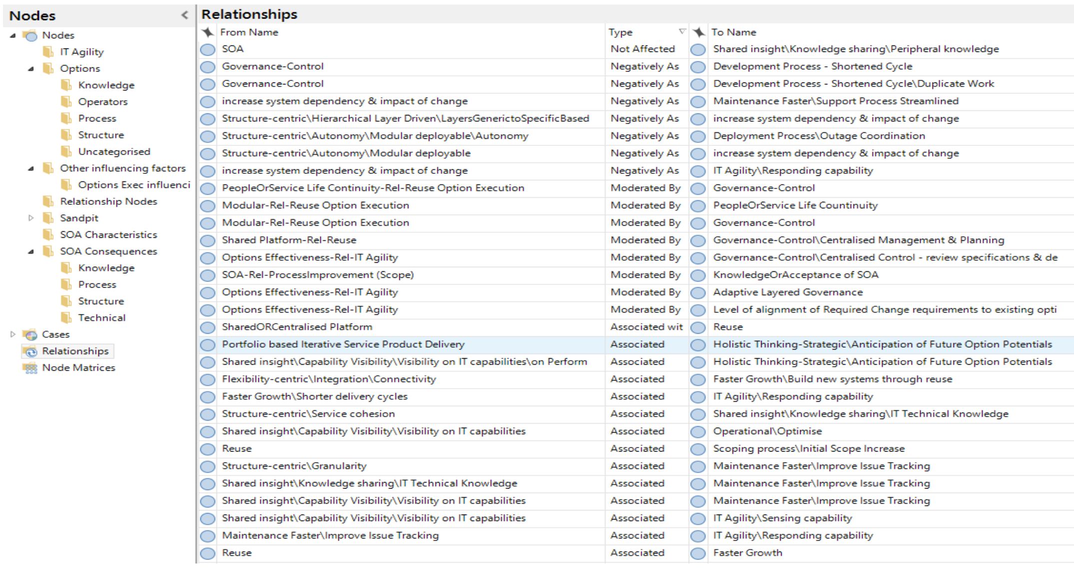
Text	Descriptive Code	Theme

"…each service doesn't do too much and doesn't do just a little bit … basically each service had to do something specific but isolate it to other services"	Granularity	Structure-centric
"…So some of the backend systems could change without impacting the frontend systems.. . "	Loosely coupled	
"by building that we actually catered for any new applications which comes later. we changed the data services to make that generic "	Generalisable	
"Sensible boundary around them refer to … It is the discrete functionality is what I am thinking about than granularity "	Service cohesion	
"A very interesting finding … have a separate layer up on top and then the core service layer… to forces developers, to reuse those core services…"	Hierarchical Layering	

Similar method applied to the options and SOA impacts, in which the pattern coding of certain impacts revealed a new construct (options depreciators) and other impacts merged with the option effectiveness constructs.

In addition to the construct development, e.g. the SOA constructs and SOA-enabled options, the researcher identified a set of relationships between the developed concepts. Using the 'type of relationship' in NVivo, the type of effect and interaction between different factors were captured and stored in the system. Example of types are 'Not Affected', 'Negatively Associated', 'Moderated by' and 'Associated'. Each relationship was supported by text from the interviews or documents explaining the proposed relationships. Figure 4-2 shows the relationships and their type captured in NVivo.

The researcher's memos developed during data analysis also provided invaluable insight and help to uncover relationships or develop ideas. The memo involves the theorizing write-up of ideas concerning codes and their relationships as the researcher came across them in the coding process. This allows easier movement between empirical data and conceptual levels (Miles & Huberman, 1994).



Nodes
- Nodes
  - IT Agility
  - Options
    - Knowledge
    - Operators
    - Process
    - Structure
    - Uncategorised
  - Other influencing factors
    - Options Exec influenci
  - Relationship Nodes
  - Sandpit
  - SOA Characteristics
  - SOA Consequences
    - Knowledge
    - Process
    - Structure
    - Technical
- Cases
- Relationships
- Node Matrices

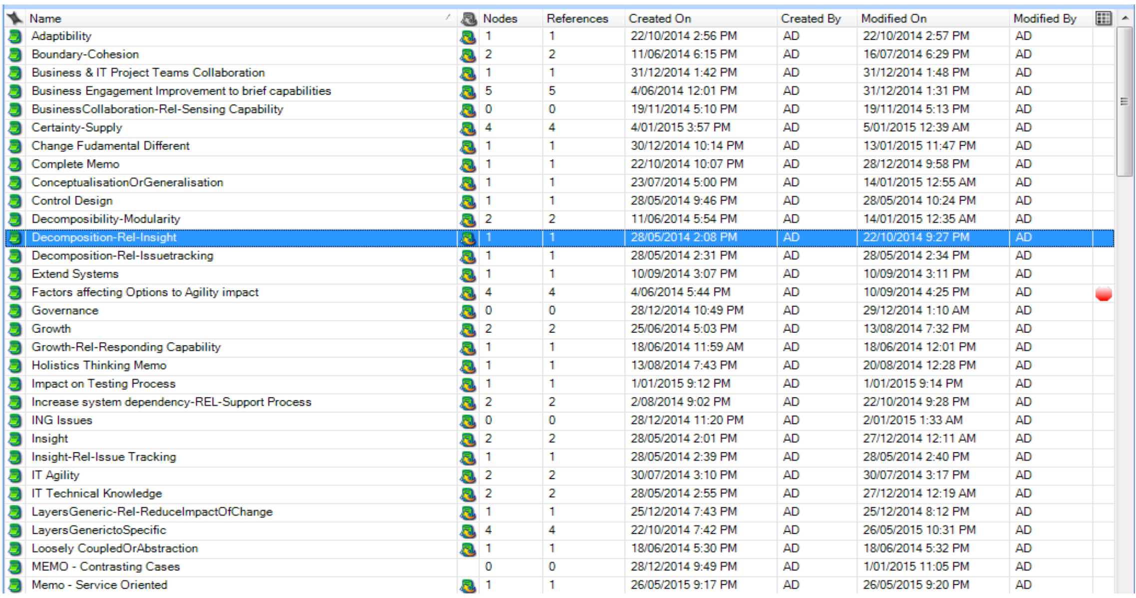
Relationships

From Name	Type	To Name
SOA	Not Affected	Shared insight\Knowledge sharing\Peripheral knowledge
Governance-Control	Negatively As	Development Process - Shortened Cycle
Governance-Control	Negatively As	Development Process - Shortened Cycle\Duplicate Work
increase system dependency & impact of change	Negatively As	Maintenance Faster\Support Process Streamlined
Structure-centric\Hierarchical Layer Driven\LayersGenerictoSpecificBased	Negatively As	increase system dependency & impact of change
Structure-centric\Autonomy\Modular deployable\Autonomy	Negatively As	Deployment Process\Outage Coordination
Structure-centric\Autonomy\Modular deployable	Negatively As	increase system dependency & impact of change
increase system dependency & impact of change	Negatively As	IT Agility\Responding capability
PeopleOrService Life Continuity-Rel-Reuse Option Execution	Moderated By	Governance-Control
Modular-Rel-Reuse Option Execution	Moderated By	PeopleOrService Life Countinuity
Modular-Rel-Reuse Option Execution	Moderated By	Governance-Control
Shared Platform-Rel-Reuse	Moderated By	Governance-Control\Centralised Management & Planning
Options Effectiveness-Rel-IT Agility	Moderated By	Governance-Control\Centralised Control - review specifications & de
SOA-Rel-ProcessImprovement (Scope)	Moderated By	KnowledgeOrAcceptance of SOA
Options Effectiveness-Rel-IT Agility	Moderated By	Adaptive Layered Governance
Options Effectiveness-Rel-IT Agility	Moderated By	Level of alignment of Required Change requirements to existing opti
SharedORCentralised Platform	Associated wit	Reuse
Portfolio based Iterative Service Product Delivery	Associated	Holistic Thinking-Strategic\Anticipation of Future Option Potentials
Shared insight\Capability Visibility\Visibility on IT capabilities\on Perform	Associated	Holistic Thinking-Strategic\Anticipation of Future Option Potentials
Flexibility-centric\Integration\Connectivity	Associated	Faster Growth\Build new systems through reuse
Faster Growth\Shorter delivery cycles	Associated	IT Agility\Responding capability
Structure-centric\Service cohesion	Associated	Shared insight\Knowledge sharing\IT Technical Knowledge
Shared insight\Capability Visibility\Visibility on IT capabilities	Associated	Operational\Optimise
Reuse	Associated	Scoping process\Initial Scope Increase
Structure-centric\Granularity	Associated	Maintenance Faster\Improve Issue Tracking
Shared insight\Knowledge sharing\IT Technical Knowledge	Associated	Maintenance Faster\Improve Issue Tracking
Shared insight\Capability Visibility\Visibility on IT capabilities	Associated	Maintenance Faster\Improve Issue Tracking
Shared insight\Capability Visibility\Visibility on IT capabilities	Associated	IT Agility\Sensing capability
Maintenance Faster\Improve Issue Tracking	Associated	IT Agility\Responding capability
Reuse	Associated	Faster Growth

Figure 4-2 - Relationships in NVivo 11

Figure 4-3 shows some of the memos out of sixty-four memos developed in the course of data analysis.

Following Yin's (2009) explanation building technique, the researcher compared the findings of each case to the initial proposition. The major refinement to the proposition, however, occurred when the data analysis revealed the main themes of the SOA characteristics and the options. Comparing the relationships captured in NVivo for each case with the refined propositions provided the opportunity to further refine the propositions and to develop and refine the conceptual model.



Name	Nodes	References	Created On	Created By	Modified On	Modified By
Adaptibility	1	1	22/10/2014 2:56 PM	AD	22/10/2014 2:57 PM	AD
Boundary-Cohesion	2	2	11/06/2014 6:15 PM	AD	16/07/2014 6:29 PM	AD
Business & IT Project Teams Collaboration	1	1	31/12/2014 1:42 PM	AD	31/12/2014 1:48 PM	AD
Business Engagement Improvement to brief capabilities	5	5	4/06/2014 12:01 PM	AD	31/12/2014 1:31 PM	AD
BusinessCollaboration-Rel-Sensing Capability	0	0	19/11/2014 5:10 PM	AD	19/11/2014 5:13 PM	AD
Certainty-Supply	4	4	4/01/2015 3:57 PM	AD	5/01/2015 12:39 AM	AD
Change Fudamental Different	1	1	30/12/2014 10:14 PM	AD	13/01/2015 11:47 PM	AD
Complete Memo	1	1	22/10/2014 10:07 PM	AD	28/12/2014 9:58 PM	AD
ConceptualisationOrGeneralisation	1	1	23/07/2014 5:00 PM	AD	14/01/2015 12:55 AM	AD
Control Design	1	1	28/05/2014 9:46 PM	AD	28/05/2014 10:24 PM	AD
Decomposibility-Modularity	2	2	11/06/2014 5:54 PM	AD	14/01/2015 12:35 AM	AD
Decomposition-Rel-Insight	1	1	28/05/2014 2:08 PM	AD	22/10/2014 9:27 PM	AD
Decomposition-Rel-Issuetracking	1	1	28/05/2014 2:31 PM	AD	28/05/2014 2:34 PM	AD
Extend Systems	1	1	10/09/2014 3:07 PM	AD	10/09/2014 3:11 PM	AD
Factors affecting Options to Agility impact	4	4	4/06/2014 5:44 PM	AD	10/09/2014 4:25 PM	AD
Governance	0	0	28/12/2014 10:49 PM	AD	29/12/2014 1:10 AM	AD
Growth	2	2	25/06/2014 5:03 PM	AD	13/08/2014 7:32 PM	AD
Growth-Rel-Responding Capability	1	1	18/06/2014 11:59 AM	AD	18/06/2014 12:01 PM	AD
Holistics Thinking Memo	1	1	13/08/2014 7:43 PM	AD	20/08/2014 12:28 PM	AD
Impact on Testing Process	1	1	1/01/2015 9:12 PM	AD	1/01/2015 9:14 PM	AD
Increase system dependency-REL-Support Process	2	2	2/08/2014 9:02 PM	AD	22/10/2014 9:28 PM	AD
ING Issues	0	0	28/12/2014 11:20 PM	AD	2/01/2015 1:33 AM	AD
Insight	2	2	28/05/2014 2:01 PM	AD	27/12/2014 12:11 AM	AD
Insight-Rel-Issue Tracking	1	1	28/05/2014 2:39 PM	AD	28/05/2014 2:40 PM	AD
IT Agility	2	2	30/07/2014 3:10 PM	AD	30/07/2014 3:17 PM	AD
IT Technical Knowledge	2	2	28/05/2014 2:55 PM	AD	27/12/2014 12:19 AM	AD
LayersGeneric-Rel-ReduceImpactOfChange	1	1	25/12/2014 7:43 PM	AD	25/12/2014 8:12 PM	AD
LayersGenerictoSpecific	4	4	22/10/2014 7:42 PM	AD	26/05/2015 10:31 PM	AD
Loosely CoupledOrAbstraction	1	1	18/06/2014 5:30 PM	AD	18/06/2014 5:32 PM	AD
MEMO - Contrasting Cases	0	0	28/12/2014 9:49 PM	AD	1/01/2015 11:05 PM	AD
Memo - Service Oriented	1	1	26/05/2015 9:17 PM	AD	26/05/2015 9:20 PM	AD

Figure 4-3 - captured memos

Additionally, the approach of the current study in selecting similar and contrasting cases improved the validity of the researcher's findings by highlighting both the comparability and variability of the cases. The NVivo Matrix coding report and the NVivo project map facilitated the comparison between the cases.

Regarding the data display, the constructs and their definitions are presented in arrays and matrix of categories (Miles & Huberman, 1994). Similarly, the interrelationships between concepts are presented in matrix display and high level network displays (model), while the focus of the network remains on the associations between the presented concepts rather than describing the process of events and activities happening overtime (Miles & Huberman, 1994)

The developed model reflects constructs and their relationships. To verify the research findings, the researcher's findings are contrasted with the extant literature at the end of the data analysis.

4.5 Validity and Reliability

Case study research has been criticised for high level of subjectivity and concerns about generalisability (A. S. Lee, 1989; Yin, 1994). Scholars (Dubé & Paré, 2003; Eisenhardt, 1989; Patton, 1999; Yin, 1994) have suggested that validity and reliability in case study research involves using clearly defined methodological guidelines for ensuring construct validity, internal validity, reliability and external validity. Table 4-9 lists a set of actions that the current research has taken to address the rigour requirements of the case study research.

Table 4-9 - Rigour in the case study research

Test	Tactics in case study research	Phase of research	How to be addressed in the current study
Construct Validity	- Use multiple sources of evidence (A. S. Lee, 1989; Yin, 1994) - Establish chain of evidence (Yin, 1994) - Have key informants review draft case study report (Yin, 1994)	1. Data collection 2. Data collection 3. Report	- Interviews and documents are used. Multiple individuals are interviewed within an organisation. - Evidence is coded and analysed using NVivo. - Key informants will review the transcribed interview documents. - Use of Pilot case - Use of a-priori constructs

Internal Validity	- Do explanation building (Yin, 1994) - Use natural controls (A. S. Lee, 1989)	1. Data analysis 2. Research design	- Explanation building has been applied. - Cases and embedded cases are selected to provide the natural controls.
External Validity	- Use replication logic in multiple-case studies (Eisenhardt, 1989; Yin, 1994) -	1. Research design	- Both theoretical and literal replication considered when cases selected.
Reliability	- Use case study protocol (Miles & Huberman, 1994; Yin, 1994) - Develop case study database (Miles & Huberman, 1994; Yin, 1994)	1. Data collection 2. Data collection	- An interview protocol developed and tested using a pilot. - NVivo software holds the case study database. Case study reports, transcripts, construct table, relationship tables are used.

As listed in Table 4-9, construct validity is the first concern that needs attention. As per Yin (1994), construct validity concerns the issue of whether a concept has been measured correctly and if empirical data in multiple situations leads to the same conclusions. The construct validity is improved by using multiple sources of evidence (to essentially provide multiple data points for the same phenomenon), having key informants review the case study report (to improve the accuracy of case study data) and establishing a chain of evidence so a reader can trace the chain of evidence (A. S. Lee, 1989; Yin, 1994).

The current study has accordingly applied the above tactics and collected data from multiple informants and project documentation in each studied case. Asking the same question from multiple informants usually provided similar responses and confirmed that the measured concepts were accurately captured. In the cases that the responses were inconsistent, further clarification was sought and further probing clarified the inconsistencies or resulted in the development of new concepts.

Additionally, all interviews were recorded, transcribed and coded in NVivo to improve the traceability of the information back to the individual sources. The transcribed interviews were sent back to the informants for review and further follow up meetings were arranged when clarifications were required.

On the chain of evidence, the next chapter presents the constructs and their relationships in form of tables with references to the informants' comments. All concepts are also clearly defined based on the data analysis. The adopted approach establishes a chain of evidence on how the captured data led to the developed constructs or relationships.

Furthermore, the current study took two additional steps to improve the construct validity: use of a priori constructs in construct development and verification of the data collection questioner in a pilot study. These two steps improved the validity of measured concepts further.

The second concern is the internal validity. Internal validity concerns the issue of alternative unidentified variables impacting on the dependent variable and is achieved by using pattern matching to ensure that case study data cannot be explained by rival theories with different independent variables (Yin, 1994). The current study improved its internal validity by combing a deductive and inductive approach and comparing the initial propositions retrieved from the theory to the data collected from the field. Application of pattern matching particularly from multiple cases in different organisations would have presented if there were alternative independent variables involved in the study. The clear research boundary also limited the interference of other independent factors.

The other concern that the current study has considered is the external validity which involves the generalisability of the findings of the study. Selection of multiple case studies and replication logic improved the generalisability of this research.

Finally, reliability concerns the stability and consistency of the study over time and is ensured by creating and maintaining a case study database and developing a clear case study protocol. To address the above concern, the current study followed a formal process sourced from the literature and captured its evidence in the NVIVO database.

In addition to validity and reliability, it is important to review the bias involved in this research. Warwick and Lininger (1975) noted four interviewing mistakes:

- Reshaping questions to match the participant's role in the project.
- The social desirability bias occurring when participants constructed answers to conform to the norms of their location or professional group.
- The self-presentation bias occurring when participants describe their role in past events in a more favourable or important manner than actual.
- The plausibility bias occurring when portions of an event had been forgotten and reconstructed with plausible explanations that differed from the actual events.

The current study reduced the impact of these issues by deep probing to obtain the full explanations, tape recording to identify any potential bias risks and comparing explanations of the same events with answers provided by other participants to piece together the most likely sequence and explanation of events. Furthermore, all studied cases were transcribed and coded in NVivo to guide the analysis of the data. For instance, to identify the strength of SOA constructs in each case, the researcher used the NVIVO database to identify the frequency of the observed factors in the studied cases and if they are reported by multiple participants.

4.6 Summary

This chapter reviewed several research methods and justified a suitable research method and design for the current study.

Firstly, it reviewed different epistemological and theoretical perspectives and justified the adoption of post-positivist perspective in the current study. Use of post-positivism in the current study provides the opportunity to reveal pre-existing relationships between SOA and IT agility as well as being open to new data emerging from the field.

Considering the selected epistemology (post-positivism) and that SOA is a contemporary phenomenon with its boundaries and effects not well understood, a positivist case study was selected as a suitable research strategy in the current study (Benbasat et al., 1987; Eisenhardt, 1989; Yin, 1994). Case study is a well-recognised research method for hypothesis-generating (Benbasat et al., 1987), which is the aim of the current research. In this study, case study provides an opportunity to develop a middle-range theory (Eisenhardt, 1989) by generating propositions which are based on constructs developed in the current study. Such a theory-building (Eisenhardt, 1989), hypothesis-generating (Benbasat et al., 1987) approach provides insight on the SOA factors that are involved in the creation of IT agility through the options that they provide.

The research design follows the procedure suggested by Eisenhardt (1989), Huberman (1994) and Yin (1994). The adopted procedure involves definition of research questions and a set of a-priori constructs and theoretical framework from the literature before entering to the field.

The unit of analysis for the current study has two levels as per the Yin's (1994) embedded case study design. The main unit of analysis is set at the IT department level to provide an opportunity to assess the agility of the IT department in dealing with events (change factor) that trigger a change in the information systems landscape (Oosterhout et al., 2006; Yusuf et al., 1999). While the unit of analysis is at the IT department level, a change occurs at a lower level such as ‘software development using SOA capabilities’ or a ‘SOA infrastructure change’. This level, called ‘system change level’, captures the context and the nature of a change, which needs to be studied and analysed in the review of a case. This embedded case is typically a software development or infrastructure project.

Considering the above design, the current study selected three organisations as three cases, each with four system change implementations as embedded cases (mix of projects and infrastructure changes). The three cases and their consequent embedded cases provided both literal replication (homogeneous case selected to predict similar results) as well as theoretical replication logic (heterogeneous sample selected to predict contrasting results) (Yin, 1994). The contrasting cases were at the level of SOA infrastructure and SOA services that the organisation has already deployed and which were available for use. Having contrasting scales of service orientation allowed the researcher to identify different SOA characteristics, their generated options and to finally extend the merging theory (Eisenhardt, 1989; McCutcheon & Meredith, 1993). The use of replication logic in the current study improved the generalisability of the emerging theory.

Regarding the data collection, this research used semi-structured interviews and documentation as its two main methods of data collection. The project documentation provided the background knowledge to prepare the researcher for the interviews. It also provided information concerning SOA adoption and how SOA affected the delivery of projects. The interviews also followed an interview protocol, which was tested in a pilot study. During the data collection, the researcher conducted twenty-nine interviews with the IT executives, business managers who sponsored SOA projects, architects and project delivery teams in the three selected organisations.

Having the data through the documentations and interviews, they had to be analysed to extract patterns of SOA characteristics which contributed to the IT sensing and responding capabilities and their relationship with the IT agility components. Since many of the SOA concepts applicable in the current study have already been recognised in the SOA literature, the current study took a deductive approach by reusing the existing codes for the SOA (as presented in Appendix B – A Priori Constructs and Mapping to SOA Characteristics) to refine and provide a solid definition and conceptualisation for the

SOA. On that basis, it used the deductive coding methods suggested by Miles and Huberman (1994) which consists of descriptive coding and pattern coding.

The researcher used Nvivo to capture all interview transcripts and the extracted codes. Codes related to the case demographics, SOA characteristics, SOA-enabled options and IT agility were maintained in their own folders in NVivo. While the interviews reflected the options that SOA created, the data analysis also revealed a set of impacts on the organisation across the four dimensions of knowledge, process, structure and technology. Additionally, there were other influencing factors identified in the data analysis that could somehow be important for the study. The identified factors were coded and maintained in the 'Other influencing factors'. The descriptive codes captured above facilitated the next stage of coding, which was pattern coding. The created patterns represented themes and theoretical constructs for SOA and their created options. In addition to the above constructs, the researcher identified a set of relationships between the developed concepts. Following Yin's (2009) explanation building technique, the findings of each case was compared to the initial proposition to refine the propositions and develop a new conceptual model. Additionally, having both cases with similar results and cases such as the Bank Digital and the Airline Loyalty projects with contrasting results, improved the validity of the researcher's findings by highlighting both the comparability and variability of the cases. The NVivo Matrix coding report and the NVivo project map facilitated the comparison between the cases.

Finally, case study research has been criticised for high level of subjectivity and concerns about generalisability (A. S. Lee, 1989; Yin, 1994). The current study follows a clearly defined methodological guidelines suggested by scholars such as (Dubé & Paré, 2003; Eisenhardt, 1989; Patton, 1999; Yin, 1994) to ensure construct validity, internal validity, reliability and external validity of the findings.

Chapter 5. Analysis of Data

This chapter reviews the data collected from the selected cases, and analyses them to respond to the following four research questions of this study. The discussions and comparison of the findings to the literature are then covered in the next chapter.

> RQ1: What are the SOA characteristics that affect the sensing component of IT agility?
>
> RQ2: What are the SOA characteristics that affect the responding component of IT agility?
>
> RQ3: How do the SOA embedded characteristics facilitate the sensing component of IT agility?
>
> RQ4: How do the SOA embedded characteristics facilitate the responding component of IT agility?

This chapter follows the Perry's proposed structure (Perry, 2008) which suggests structuring the data analysis around the research issues to maintain the focus of analysis on the main research enquiries. The table below summarises the data analysis approach discussed in section 4.4.

Table 5-1 - Summary of Data Analysis Approach

Step	Method	Purpose	Outcome	Examples
1	Descriptive coding (Iterative process started from a priori codes)	Identify initial sets of concepts and their strength in the observed cases	Identification of SOA Characteristics, Options and IT Sensing and Responding capabilities.	Standardisation, business-oriented Change Detection IT Sensing Capability
2	Pattern coding (Iterative)	Identify common themes and causes	Identification of higher order groupings for SOA characteristics, Options & the relationships between concepts	Information Repository Structure-centric
3	Comparison with initial propositions	Refine the propositions	Refined and new propositions	The IT-sensing capability is positively affected by the 'Change Detection' knowledge option that

	(Iterative for each case and cross case)			SOA offers.

Following the above proposed structure, section 5.1 reviews the case environments and the case demographics. The case demographics provide contextual information associated with the selected cases.

Section 5.2 and 5.3 address the first two research questions (RQ1 and RQ2) by analysing the data and reporting the SOA characteristics that influence the IT agility. These sections rely on the a-priori constructs extracted from the IT and SOA literature. The a-priori constructs and their revised SOA characteristics are also summarised in Appendix B and C.

To answer the last two questions (RQ3 and RQ4), section 5.4 and 5.5 analyse the data to discover the mechanisms (Avgerou, 2013) involved in the impact of SOA on the IT Agility. The data analysis relies on the initial conceptual framework defined in Chapter 3 to guide the discovery and refinement of constructs and their relationships. Section 5.4 and 5.5 conclude with a set of refined theoretical propositions and theoretical models.

The data analysis revealed several non-SOA factors that contributed to the impact of SOA on the IT agility. Section 5.6 reports these non-SOA factors that were identified from the cross-case comparisons.

Finally, section 5.7 summarises the findings and provides an integrated conceptual model for the study.

5.1 Cases Environments / Demographics

The three selected cases and their embedded cases provided an opportunity to replicate and contrast the researcher's findings beyond one case. As presented in this section, two of the cases are from the aviation industry and the third case is from the banking industry. All cases have adopted the SOA for several years, however with different focuses and objectives. Additionally, a few business divisions had higher needs for agility, which made them different from the other divisions and projects.

Table 5-2 provides the description of each organisation and the studied projects within each case.

Table 5-2 - Organisations Descriptions

Organisations (Cases)	Sector	Assets	Revenue	Size	Size of IT	SOA Adoption Year	Projects (Embedded cases)
International Bank – Australian Entity (BK)	Banking	$50,635 million (2014)	$660 million (2014)	1000	200	2005	Digital channel, DGW, BSL, Living Super, Saving Maximiser, Bank in the box
Australian Airline (AL)	Aviation	$17,500 million (2015)	$15,800 million (2015)	30000	200	2007	A380, Blackberry, Emirates Integration, Loyalty
Australian Airport (AP)	Aviation	$11,328 million (2014)	$1,163 million (2014)	400	30	2008	AOS, PA, Internet, CUTE

The cases consist of one Australian Airline (abbreviated as AL), one Australian Airport (abbreviated as AP) and one International Bank (abbreviated as BK). The Airline and the Airport were in a similar industry, expected to predict similar results (literal replication) and the bank was selected to provide contrasting results in order to extend the emergent theory (theoretical replication) (Yin, 1994). Both airline and airport cases had relatively small IT units compared to their revenue and size of the organisations.

All three organisations had adopted SOA for over eight years. As shown in Table 5-3, the Bank applied SOA to expose its core banking functionality to other systems. The bank relied on an in-house developed SOA platform called DGW to providing business services and core system functionalities such as security, auditing, and logging. Development activities on the DGW are managed internally by a centralised IT development team. The Bank's internal IT operation team provides the operational management of the DGW, and the Bank's internal architecture team governs the evolution of the DGW within the organisation.

Table 5-3 - SOA Adoption in the selected cases

Organisation (Cases)	SOA Adoption Driver	SOA Use	SOA Platforms	SOA Governance	Sourcing – SOA Development	Sourcing – SOA Operation Management
International Bank – Australian Entity (BK)	Security, Agility	Service Orientation	In-house built running on JAVA JBoss Application Server Monitoring: JBoss Operations Network	Internal – Architecture Team (Digital Team – further control on BSL)	Internal – Centralised (Digital Team: Internal and External)	Internal – Centralised
Airline (AL)	Real-time data integration / Agility	Service Orientation and Data Integration	Oracle / BEA ESB, Oracle SOA Suite (BPEL), Excel & Sharepoint for Registry, Monitoring: built to apps.	Internal – SOA Centre of Excellence (Exception: Loyalty – further control)	Outsourced – Centralised (Exception: Loyalty)	Outsourced – Centralised (Exception: Loyalty)
The Airport (AP)	Provide Real-time operation with real-time data	Service Orientation and Data Integration	IBM Integration Bus, MQ Series, Monitoring: IBM Tivoli	Internal – Architecture Team	Outsourced – Centralised	Internal - Centralised

In the past few years, the Bank has increased its focus on its Digital channels such as mobile application and its website. Due to the access from the public network, the Bank introduced a new service layer on top of DGW, called BSL. Additionally, the Digital team had to deal with a higher rate of change and faster time to market in its Digital projects, which justified their control over the governance and development of the BSL. The control allowed the Digital team to engage external suppliers in the development of their projects from time to time. Additionally, they adopted an agile software development process in the digital projects. Examples of Digital projects are the Bank iPhone and Windows Mobile Applications for consumers and automation of Rate Change on the online channels.

The Airline, however, used SOA for data integration and exposure of the available services in the systems through their Oracle Integration platform. The development and operational management of the platform and services were outsourced to a supplier. The Airline SOA Centre of Excellence team provides the governance of the SOA in the organisation. The exception was the Airline loyalty team, which built a separate service layer for its use and engaged different suppliers for the development and operational management of the services.

Similar to the Airline, the Airport used SOA for data integration and service orientation. The Airport relied on an IBM Enterprise Service Bus as its SOA platform. The development on the platform was outsourced to external suppliers, whereas the operational management and system support were kept in-house. The SOA governance remained in-house with the Airport architecture team.

A few projects from each case were selected. Table 5-4 provides information concerning the projects reviewed for the current research. In the table below, the Airline is abbreviated as 'AL', the Airport abbreviated as 'AP' and the Bank abbreviated as 'BK'. The '_P' appended at the end of the case name indicates a 'project'.

Table 5-4 - Projects Description

Organisations (Cases)	Projects (Embedded cases)	Description
International Bank – Australian Entity (BK)	DGW, BSL (BK-P)	Infrastructure projects to create a platform for service orientation. The DGW exposed the core banking internally and the BSL exposed the services to external consumers on the online channels.
	Digital Channel Projects (BK-P)	Projects sponsored and managed by the Bank Digital team to expand the Bank online channels. Some of the projects were the interest update on digital channels (Bank public Web site, Bank mobile applications on different devices), development of iPhone, Android and Windows Mobile applications. The Digital team applied Agile methodology (Scrum) in the last 6 month.
	Living Super (BK-P)	The program launched the Bank superannuation product to retail customers in the market. The second phase of the project was to introduce advisors who can sell the superannuation products to their clients. The program had multiple external vendors involved. The project slipped over 1 year compare to its original Go Live date.
	Saving Maximiser / Loyalty Bonus (BK-P)	An extension to one of the existing Bank product. The project was delivered on-time despite its challenges and short project period (8 weeks). The project had a great involvement from all stakeholders with representation from all groups.
	Bank in the box (BK-P)	An infrastructure project to expedite building a new environment which consists of all major components and systems of the bank, including its core banking. The project aimed at reducing the dependency on the number of environments available for development. The bank in the box project increased the limit of 3 parallel active projects to 40 active projects in parallel.
The Airline	A380	The project aimed at improving the passenger experience by

(AL)	(AL-P)	providing relevant information to the passengers on their A380 in-flight entertainment system. The development and design done internally.
	Blackberry (AL-P)	A follow-on project to deliver similar information exposed in the A380 project on the Blackberry and iPad. The Airline outsourced the development and operational management of the SOA platform after the A380 project, which resulted in the services created in the A380 project not being utilised.
	Emirates Integration (AL-P)	Integration of the Airline and Emirates booking processes such as chauffeur pickup.
	Loyalty (AL-P)	Projects related to the management of frequent flyers on different channels (web, mobile). These projects were managed by the loyalty team independent of the group IT.
The Airport (AP)	AOS (AP-P)	Project to replace the flight scheduling, flight display and flight management systems in the airport. The project involved integration of these systems to all other airport systems such as the web sites and Baggage handling system.
	PA (AP-P)	Project to implement the passenger announcement system in the terminal. The project involved integration to the AOS.
	Internet (AP-P)	The upgrade of the airport web site. IT involved in the integration to the AOS system.
	CUTE (AP-P)	The integration to the Common Use Terminal Equipment in the terminal.

Relevant information related to each project is provided in the next sections.

5.2 SOA characteristics affecting IT Sensing Capability

This section answers the first research question of this study, which involved identification of the SOA characteristics that affect the sensing component of the IT agility. The data analysis revealed six SOA characteristics that impacted the sensing capability of the IT. Pattern coding of these six characteristics revealed a similar theme for them which was ‘Information-centric’.

Table 5-5 lists these six characteristics and the evidence from each case on their effect on the IT sensing capability.

The table also shows the strength of each SOA characteristic in the studied cases with one of the following values (Miles & Huberman, 1994): ‘Absent’, ‘Weak’, ‘Average’, ‘Sufficient’ and ‘Strong’. Each value shows the level of observed characteristic in the studied case, with 'Absent' reflecting non-presence of the characteristic and 'Strong' full availability of the characteristic. 'Sufficient' shows that the characteristic was adequately present in certain cases or groups, however not consistently in all projects or groups to provide its full benefits.

Table 5-5 - Information-centric characteristics case evidence

Category (parent node)	SOA Characteristics (child node)	Airline (AL) (Organisation & Projects)	Airport (AP) (Organisation & Projects)	Bank (BK) (Organisation & Projects)
Information Repository (Subsection 5.2.1)	Standardisation	*Strong* at both levels, e.g. design patterns, standard technologies and consistent processes specially in deployment.	*Strong* at both levels, e.g. design patterns, standard data models (CIM & AIDX) and consistent processes adopted. *Exception*: CUTE project (*Average*)	*Strong* at both levels, e.g. DGW & BSL enforced consistent architecture to all projects and consistent processes in deployment and testing adopted.
	Business-oriented	*Strong* Majority of services were business-oriented and some technical.	*Average* A mix of business-oriented and technical services	*Strong* Majority of services were business-oriented, with some technical.
	Service Definition	*Strong* While not all services, many of the services were simply and clearly documented.	*Average* The service definitions mainly targeted the technical resources. *Exception*: AOS project (*Strong*)	*Average* The service definitions mainly targeted the technical resources (Java documentation)
Information Discovery (Subsection 5.2.2)	Service Discovery	*Sufficient* Available to business and IT on Sharepoint. Business use at early stage.	*Weak* Not available centrally. The information captured in separate documents.	*Average* Available to business and IT online. Mainly used by IT.
Information Disseminati on (Subsection 5.2.3)	Service Monitoring	*Sufficient* No monitoring of business processes KPI	*Sufficient* No monitoring of business processes KPI	*Average* No real-time service monitoring. The behaviour of services analysed through log analysis.
	Event Driven	*Strong* Many events exposed and available.	*Strong* Many events exposed and available.	*Average* Less number of events exposed.

As listed in the table, the 'Information Repository' category emerged from SOA characteristics that contributes to capturing information related to the definition and design of services. The standardisation, business-oriented, and service definition characteristics each hold information such as design patterns, business context and definition of services which will contribute to the knowledge within the organisation.

'Information Discovery' emerged from service discovery characteristic which facilitates the discovery and identification of the captured information. Lastly, 'Information Dissemination' emerged from 'Service Monitoring' and 'Event Driven' characteristics which contribute to the notification and distribution of information suitable to sense a change in an information system.

Further review of the above categories also revealed a common theme among them in that they are all concerned with information. The information-centric characteristics are involved in the capture, discovery, and distribution of information related to a service.

This section has four subsections. As shown in Table 5-5, the first three subsections explain the three identified categories ('Information Repository', 'Information Discovery' and 'Information Dissemination') and their corresponding SOA characteristics.

The last subsection, 5.2.4, summarises the discussions in this section by listing the SOA characteristics that affect the IT sensing capability. Summary of the results and their nominated measures have also been tabled in Appendix C.

5.2.1 Information Repository category

Information repository category emerged from SOA characteristics that capture information related to the design of a service. Each of the three characteristics (standardisation, business-oriented, and service definition) in this category contributed to knowledge within the organisation which improved the IT sensing capability.

All cases at both project and organisation levels recognised the importance of 'Standardisation' in SOA. The initial coding of the interviews identified five individual codes concerning consistency of processes, design and technologies used in SOA. Further review of the codes, however, showed that the five codes are concerned with 'consistency' in activities or adopted technologies, which suggested the merging of these five codes to 'Standardisation'. Therefore, the 'Standardisation' was defined as formulation and implementation of consistent rules, guidelines and specifications in design and use of services. The emerged definition from the analysis extends what available in the literature (Oh et al., 2007; Ross & Weill, 2005; Tiwana et al., 2010) by considering standardisation as:

- Consistency in ways to design services and solutions to solve a particular problem (pattern-oriented)
- Consistency in service development and management processes and policies such as documentation, testing or operationalization of services
- Consistency and adoption of industry standards in underlying technologies used in the system development or the SOA platform, such as XML, SOAP and HTTPs
- Consistent service architecture, e.g. the number of service layers
- Consistent data semantics and interface definitions, particularly when they are consistent with the industry-wide notations and standards, e.g. IATA AIDX

Standardisation provided the teams with *consistent* knowledge on how to develop and manage services, or as one of the architects put it *"…consistent way of doing things"*.

While standardisation was not specific to SOA, all interviewees considered it an important characteristic of SOA. One of the Bank architects reflected on how SOA, and particularly the standardisation of architecture and patterns, have brought predictability on the IT supply. He considered predictability on the IT supply a factor that improves the IT sensing capability in an environment with unpredictable system development demands: "*We know exactly how we are going to - where the business logistics done, [you know] exactly where it going. So, we don't have to have that debate. And also we know largely because architecture is very simple one, every change largely is in these three layers in some various form. So, from that point of view, it's where the skillset becomes predictable, so people start - people process technology, they do all three, very predictable. The people we know exactly top skills we need, we can - so from my point of view, the way I look at it is, if you can't get certainty out of your demand, you got to get it out of the supply. So, our demand is variable, we don't know, the business don't know what they want to do and we really do not know what they want to do because the organisation change…"*. The Airport head of IT also reflects on the clarity that SOA standardisation and consistency provides to the organisation: "*They will have that benefit from a deployment perspective but I think it goes deeper than that in terms of this consistency across the organization so we are going to have a dictionary that's sort of more universal. So reduces the duplication, improves clarity…"*.

The Bank's development manager recognised the importance of standardisation by explaining that the use of design patterns allowed his team to quickly share knowledge and create a good support network within the team. The shared knowledge allowed the team to collaborate on the issues quickly. The standardisation also provided the development manager with the flexibility to swap and/or introduce new resources to expedite the development. On the consistent processes, one of the IT managers at the Airport explained that the use of consistent service development and management processes, such as the standard process in system monitoring and logging, provided them with shared knowledge to identify and sense the system issues quickly.

The opposing case was the CUTE project at the Airport which had to use a non-standard queuing system. Review of project documentation showed that the Airport had to integrate to a Microsoft Queuing system (MSMQ) which was not standard for the Airport and the IT did not have the necessary knowledge to develop and maintain it. The lack of knowledge on the MSMQ affected the IT sensing to adequately size the development work and the ongoing management of the system.

The other identified SOA characteristic in the 'Information Repository' category was 'Business-oriented'. 'Business-oriented' was considered as the extent the function and definition of service was aligned with business concepts and meanings. The Airline IT manager highlighted this by stating "*Grouping things in a thing that makes sense from our business as supposed to from the IT perspective".* A business-oriented service can be high-level with a meaning close to a business process or it can be close to a business data.

The analysis of cases showed that services aligned with business concepts improved the IT's knowledge of business and ability of business to understand the services, which in turn created a shared understanding of IT capabilities and the potential gaps in the new system development or system change. The business and IT alignment are reflected in the comment from one of the Airline architects: "*traditionally people said something like 'I need a PNR' (that's Passenger Name Records). But PNR actually it doesn't mean a lot with non-technical persons, instead currently with the services, we get the name like retrieve booking, so people realises actually it is related to the flight booking. So with that, if they look at the SOA service now always is named under business context, yes, that improves the alignment of between business and IT."*

Such business orientation of services and consistency in how business and IT communicated provided the IT and business with a sense of change required in the systems at the initial planning or idea formation phase. One of the Bank architects reflected on the above point: *"... So these services definitely help us for agility. ... So for example they [digital team] want to build a page, on that page they [digital team] want to get the customer information and they want to do something else as well, which they [digital team] say you know what, 70% of services are already there. We need to build this page. Create another service in DGW which is again reusable to other implementation and here you go, you have done that piece of work in the reusability sense. You have done that piece of work very quickly."*

The last SOA characteristic that captures information related to services is 'Service Definition'. The service definition characteristic captures information regarding the service interface and its function, which helps the business and IT to understand what capabilities are available. The Airline SOA governance manager highlighted the above in the Airline context: "*The service contracts are pretty simple to understand so you don't have to talk a lot technical to understand what that service provides, it's very simple that this services use for this purpose. So that can be understood from business and IT so business doesn't have to go into detail ... it is already extracted as a service, is already clearly documented yeah it can be reused, or if some services are actually enterprise related requirements that will also help the project manager to understand ...".* The above confirms the service definitions need to be simple and clear in explaining the definition and purpose of the services to the business and IT. Clear service definitions improve the business and IT knowledge of services and their capabilities.

The data analysis of the current study showed that all cases achieved the documentation of their service definitions, however with difference degree of simplicity and clarity for business users. The Airline had a better success in documenting the service definitions clearly, consumable by business. The Airport and the Bank mainly focused on the technical resources in the documentation of their services. The exception was the Airport AOS project which created a business data dictionary to complement their service definitions. The data dictionary defined the data elements used in the service definitions in a business language usable by both IT and business units. The data dictionary created a shared understanding of terms such as 'PAX' (Passenger numbers) and how they were populated in the system.

To summarise, the cases revealed the three characteristics of 'Standardisation', 'Business Oriented' and 'Service Definition' improve the IT and business knowledge by capturing information concerning the services.

5.2.2 Information Discovery category

Information discovery category emerged from SOA characteristics that contributed in discovering information concerning services. Ability to discover and locate services improved the IT knowledge and better sensing capability. The identified SOA characteristic in this category was ‘Service Discovery’.

‘Service Discovery’ involved classifying the services and publishing their information in a way that the services can be found and used easily when needed.

In the studied cases, each organisation used different tools and methods for publishing information. The Airline used excel spreadsheet published on their SharePoint (Intranet website), and the Bank used a custom developed website which published the JAVA documentation of the services. The Airport, however, did not have a central publication of the service definitions.

The analysis revealed that the service discovery improved the IT knowledge on the available services and consequently to sense what is required in the development of a new system. The business knowledge improvement from service discovery was however found to be limited. The Airline was the best case, which business did access the site for consumption. The Airline governance manager reported this as: *“And it helps you go to find where the services are and compare the existing services with what the product needs to do and yes so having a repository is a must. Sometimes business teams ask for [you know] to look into the repository, but still is relatively early days and you need to get yeah, more visibility with business need to look into it.”.*

In other cases, the reason for the business not accessing the discovery services was reported as: *“..., so it's difficult, they are not plain English enough, that's quite confronting for a non techie.”* which emphasizes the need for services definitions to be business oriented and simple and clear, as previously identified.

The other finding was related to the classification of the services. In both cases of the Airline and the Bank, all services were classified based on a specific taxonomy. The classification improved the access and use of services by developers, hence reduced confusion and possibly the duplication of services. The Bank classified services based on their functionality, whereas the Airline adopted a more sophisticated classification based on the business capabilities that the services deliver. The Airline architect reflected this: *“And also all of our services are also classified by different taxonomies, for example* *if there is a data service it's categorized by the information area. So the developers can find them very easily. For example, I need an operational type data then I know okay under that category what kind of service available for me to use. And for the business activity process it's categorized based on the business capabilities. So if I want to do something for example, I want to hire someone so what kind of services is available and I just go to that business capability categorization to see what service is available there.".*

The 'Information Discovery' publishes and makes service information available and easily accessible to the service consumers. Thin will improve the knowledge within the organisation. The studied cases however showed that the business knowledge improvement through the information discovery was best at early stagees.

5.2.3 Information Dissemination category

The 'information dissemination' category is concerned with the SOA characteristics that participate in distributing information suitable to sense a change in an information system. This category emerged from two SOA characteristics observed in the cases: 'Service Monitoring' and 'Event Driven' characteristics.

'Service Monitoring' characteristic involved exposure and availability of access to the behaviour of services and their quality measures in a usable form for the IT and business. All the studied cases had some form of service monitoring. The Airport had a monitoring platform, which allowed them to monitor the health of their services and got notified when there was an issue with the system. The implementation relied on a commercial monitoring platform (IBM Tivoli), whereas the Airline and the Bank both had custom-built solutions to monitor their service platform and services. The Bank had less sophisticated monitoring for the services. The platform monitoring relied on JBoss Operation Network with Splunk aggregating the system logs centrally. All the cases reported awareness of issues and ability to sense any potential change required in the system as the main benefit of 'service monitoring'. One of the Airport developers explained how the monitoring and the subsequent analysis helped them to identify the improvements required in the system: *"it's monitoring ... ... this month we have 10 defects, 10 issues with for example I think like Dwell, Dwell has some issues. So then we say this month we had 10 defects, with sort of maybe we should then think about doing some optimization ... and that's actually is the sense of change.".* Similarly, one of the Airport architects explained the importance of reporting the service usage metrics to create awareness and demonstrate the potentials of the service reuse (service option execution): "*And you don't really get metrics without monitoring. So, I think monitoring is a key path of an SOA. Because it's pretty useless if you can promote reuse without saying that the service is actually being used in the first place.*" As reported in multiple cases (the Airport and the Airline), such metrics allowed the organisation to decide if the service should be decommissioned, extended, or improved. In none of the cases, the monitoring was extended to monitoring of the business processes and their key performance indicators (KPIs), though the Airline suggested they are planning to implement this level of monitoring in the coming year.

The last SOA characteristic that was identified in the cases is 'Event Driven' characteristic. 'Event Driven' is a style of architecture that involves notifying changes detected in data or system states to interested parties promptly. This characteristic was reported in both the Airline and the Airport cases. Such architecture was considered for real-time decision-making, and was implemented by providing events to the consumer in a timely manner. One of the Airline IT managers explained this: *"... when the change happened (delay, etc.), the delay information and its knock-on effect was communicated to the inflight system during the flight, informing the particular passengers on their next flight due to them missing their original flights and what they need to do. Information emerging, such as gate number that the people need to go to or information that is not know at the time when the aircraft departs*". Similarly, from one of the Airline architectures on the importance of 'Event Driven' for them: "*Yeah, absolutely, especially SOA contain… one of the important part of a SOA is event driven architecture. That greatly improve the sensing capability of the IT especially how… the thing is we… actually create quite a number of business process react with real time business events, such as the one flight schedule has changed, you know based on that event, the SOA process can basically cancel a flight booking, cancel chauffeur driver services for the business class passengers, so they actually greatly improve the customer experience.*"

In short, discussions in this section revealed that 'Information Dissemination' as the combination of 'Service Monitoring' and 'Event Driven' provides IT and business with a timely awareness of changes that might be required in the systems.

5.2.4 Summary of SOA characteristics affecting IT Sensing Capability

This section after analysing the data collected from different cases uncovers the SOA characteristics which contribute to the IT sensing capability. This finding provides answer to the first research question of this study:

- What are the SOA characteristics that affect the sensing component of IT agility?

The Pattern coding process revealed 'information-centric' as a common theme for SOA characteristics which contributed to the IT sensing capability. The 'information repository' characteristics capture information concerning the service design and definitions, and the information discovery characteristic facilitates the discovery of this information. Similarly, information dissemination notifies the potential system changes and consequently improves the sensing capability.

Figure 5-1 presents the above three categories and the SOA characteristics that emerged from the data graphically, with their description listed in Table 5-6.

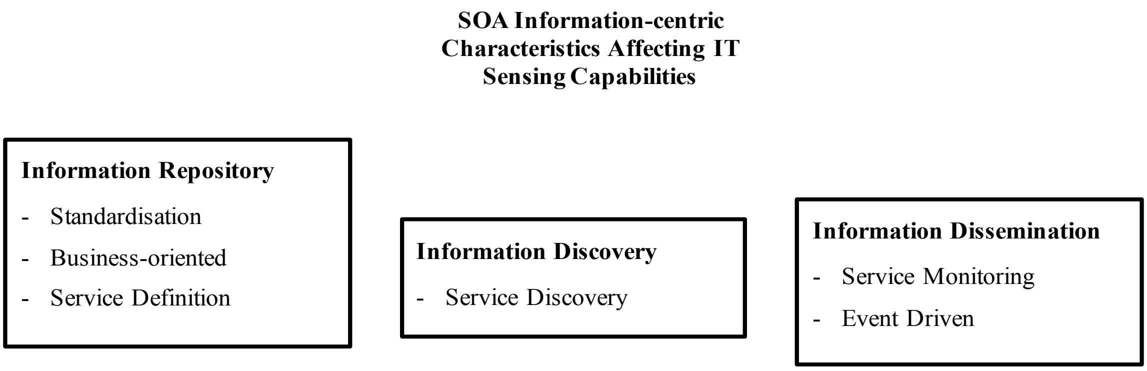


Figure 5-1 - SOA Information-centric Characteristics

The description of the SOA characteristics was based on the a priori construct definitions, refined through the data analysis as reported in Appendix C .

Table 5-6 - SOA Characteristics affecting IT sensing capability

Theme	Category (parent node)	SOA Characteristics (child node)	Description
Information-centric	**Information Repository**	Standardisation	Formulation and implementation of consistent rules, guidelines and specifications in design and use of services.
		Business-oriented	The extent the function and definition of service is aligned with business concepts and meanings
		Service Definition	The extent information regarding the service interface and the functionality of the service are captured and published
	Information Discovery	Service Discovery	The degree services are managed to become discoverable, involving classification and publication of services in a way they can be

			found easily when needed.
	Information Dissemination	Service Monitoring	Level of exposure and availability of access to the behaviour of services and their quality measures in a usable form for IT and business.
		Event Driven	The extent changes detected in data or system states are exposed and notified to interested parties in a timely manner.

The next sections provide answers to the other research questions, particularly how these characteristics impact the sensing capability.

5.3 SOA characteristics affecting IT Responding Capability

This section answers the second research question of this study by identifying the SOA characteristics which affect the responding component of IT agility. The studied cases revealed a number of SOA characteristics that contributed to the responding capability of the IT, mainly through providing 'reuse' and 'flexibility'. Pattern coding of these characteristics revealed two themes including 'Structure-centric' and 'Flexibility-centric' for these characteristics. The emerged themes were based on the role each category provided in the option generation for IT.

Table 5-7 lists the identified characteristics and the evidence from each case on their effect on the IT responding capability. The table also shows the strength of each SOA characteristic in the studied cases with one of the following values: 'Absent', 'Weak', 'Average', 'Sufficient' and 'Strong'.

Table 5-7 – Structure-centric and Flexibility-centric characteristics case evidence

Themes	SOA Characteristics (node)	Airline (AL) (Organisation & Projects)	Airport (AP) (Organisation & Projects)	Bank (BK) (Organisation & Projects)
Structure-centric **(Subsection 5.3.1)**	Granularity	*Sufficient* The services were selected with suitable granularity. There was, however, evidence of services which did not follow the guideline.	*Strong* Majority of services were selected with a suitable granularity, especially in flight related services.	*Sufficient* The services were selected with suitable granularity. There was, however, evidence of services which did not follow the guideline.
	Loosely coupled	*Strong* The design rule consistently applied.	*Strong* The design rule consistently applied.	*Strong* The design rule consistently applied.

	Generalisable	*Strong* Majority of services in the lower layers were generic. The rule applied in the design and governed by SOA CoE.	*Strong* Majority of services were generic. The rule applied in the design, governed by architecture team.	*Average* Services in the DGW layer were generic, however there were also duplicate services due to generalisability issues for some. The rule applied in the design, however loosely governed.
	Service cohesion	*Strong* The services focused on related functionality.	*Strong* The services focused on related functionality.	*Strong* The services focused on related functionality.
	Autonomy	*Strong* Particularly by distinct deployment packaging of services based on their dependency to provide autonomy at run-time.	*Strong* Achieved by creation of separate deployable packages for related services.	*Sufficient* All services deployed in one deployable package, however 'Bank in the box' project provided multiple environments to the bank to isolate changes to services in multiple projects.
	Hierarchical Layering	*Strong for Loyalty projects only* Observed with loyalty team only. Provided them with options for service orchestration and service development.	*Weak* While there was an additional layer (ECG) to service the external parties, the distribution of services was mixed.	*Strong for Digital projects only* Observed with Digital team only with BSL and Digital channel projects. Provided them with options for service orchestration and service development.
	Composability	*Sufficient* Service orchestration done using their SOA technology, which provided rapid development.	*Absent* Minimum service orchestration observed.	*Weak* Service orchestration done programmatically by the development team. No benefits from the toolsets.
Flexibility-centric (Subsection 5.3.2)	Loosely coupled	*Strong* Use of ESB provided loosely coupling.	*Strong* Use of ESB provided loosely coupling.	*Strong* DGW provided loosely coupling.
	Integration	*Strong* Use of an ESB provided multiple options for integration.	*Strong* Use of an ESB provided multiple options for integration.	*Weak* Due to custom built platform, the DGW provided limited capabilities in adaptability and connectivity.
	Standardisation	*Strong* also facilitated by the SOA platform (ESB).	*Strong* also facilitated by the SOA platform (ESB).	*Strong* also facilitated by the SOA platform (DGW).

	Framework Driven	*Strong* facilitated by the SOA platform (ESB).	*Strong* facilitated by the SOA platform (ESB).	*Sufficient* facilitated by the SOA platform (DGW). Functionality based on the services built in the system.

The 'Structure-centric' theme emerged from the characteristics that focused on the underlying structure, boundary, and composition of services to promote reuse. While Granularity, Hierarchical Layering and Composability define the characteristics associated with the arrangement of services, Generalisability, Service Cohesion, Autonomy, and Loosely coupling define the boundary of services.

The second emerged theme in the SOA characteristics was 'Flexibility-centric' that is concerned with the ability to introduce changes in SOA systems. Each of these two themes is captured as a high-order node in NVivo with their corresponding characteristics as their children nodes. The analysis of the cases revealed that the above two categories influence the IT responding capability, through the options they provided.

The next three subsections present more findings related to each of these two categories both within and across various cases. As shown in Table 5-7, the first two subsections explain the two identified themes ('Structure-centric' and 'Flexibility-centric') and their corresponding SOA characteristics. The last subsection, 5.3.3, summarises the discussions in this section by listing the SOA characteristics that affect the IT responding capability. Summary of the results and their nominated measures have also been tabled in Appendix C.

5.3.1 Structure-centric theme

The 'Structure-centric' theme emerged from SOA characteristics that were concerned with the design of a service and the underlying structure of the services together. The data analysis revealed that the structure-centric characteristics impacted the IT responding capability by reducing the development effort through options such as reuse. The SOA characteristics with 'Structure-centric' theme include 'Granularity', 'Generalisability', 'Service Cohesion', 'Autonomy', 'Loosely coupling', 'Hierarchical Layering' and 'Composability'. Each of the above characteristics is explained below, with evidence from the cases.

‘Granularity’ as the first emerged characteristic is the degree of functionality embedded in a service based on the number of tasks it handles, the amount of data it processes and the number of external interactions it has. The data analysis in all cases revealed that granularity of services was a measure that required an optimum value. If all services were designed to be highly granular (fine-grained services), there would have been operational implication such as performance issues for the system. Similarly, a service with low granularity (course-grained) limited the possibility of future reuse. The above is more clearly visible in the quote from one of the Airline IT managers: “*...so the design approach of having services with reasonably sensible boundaries around them and reasonable granularity helped us with reuse and as did the tool itself.”.*

One of the Airport ESB developers explained the correct level of functionality in each service: *“…We focus a lot of time making sure each service doesn’t do too much and doesn’t do just a little bit and that’s the key actually of, for this sort of architecture, and basically each service had to do something specific but isolate it to other services”.* One of the Bank architects complemented the above comment by elaborating on how making a service course-grained adds value to the service consumer: *“we have looked at, saying for example, everyone goes through login. And the login is where we grab a lot of data. So, what we do, is that we - because everyone goes through login, we put an orchestration, or just one call, we will do the work, and goes back to the one call. But the other ones, that for example, the paging, the third level down to some page, that makes ten calls, that doesn't get called very often, and we will just leave it like that.”*

Above comments indicate that granularity of services creates future value in the form of reuse, while it can also cause operational issues such as additional complexities and performance degradation if the granularity of services is not selected appropriately.

The next service characteristic revealed from the data was ‘Generalisability’. As defined in Appendix C - Emerged SOA characteristics affecting IT Agility, generalisability referred to the ability of a service to address a number of use cases rather than a specific use case. The data analysis showed that the generalisability of a service assists the future service reuse. This is visible in the comment from the Bank developer manager: *“That’s right if you look at the most used services been in the DGW you know there is it drops of very quickly so there is 20 or 50 that are very heavily used and then it starts dropping off you know they are the ones that are the most generic as supposed to getting customer information getting account details move money they have slightly different names for* *that but that's the functions they are performing."*. Similar observation was made in the Airline Delivery and Support manager's comments: "*We had to build a data service for an operational data source… So when we started building that, the requirement for the system was very specific to a set of data. .... So what we decided was, because this is a data which is going to be reused by multiple systems let's …try to understand what sort of data that is, that would be required by other users .... 20% of the data which is required by the application .... 80% of it is not being used by that application. But by building that we actually catered for any new applications which comes later. we changed the data services to make that generic."*.

The other service characteristic that was in the a-prior constructs and reported in the data was 'Service Cohesion'. The SOA literature reported Service Cohesion as the degree the operations of a service having related functionality (Baskerville et al., 2005; Erl, 2004; Newcomer & Lomow, 2004; Papazoglou & Yang, 2002). The initial coding recognised this as service boundary, however as the analysis progressed and the codes were reviewed, the boundary and service cohesion were merged. The service cohesion is one of the characteristics that impacts the level of service reuse in future. The Airline IT manager described this as: "*Sensible boundary around them refer to ... It is the discrete functionality is what I am thinking about than granularity.*". Similarly, this can be observed in the discussions with one of the architects at the Airport: "*So, things do discreet activities. So, I have got in mapping that, it's a message, it's very discreet activities, it sounds very easy to understand what goes on in five years time, I guess time when someone comes to a place what we have done, that can re-lift the rate, okay, I get it, it's easy. So, self-contained."*.

The other characteristic that was identified in the data and confirmed by the literature was 'Service Autonomy'. The SOA literature defines Service Autonomy as the degree the logic governed by a service resides within an explicit boundary. The service has complete autonomy within this boundary and is not dependent on other services for the execution of this governance (Erl, 2005; L. O'Brien et al., 2007). The service autonomy reduces the run-time dependency on other services and can operate independently, as explained by one of the Airline architects: *"So actually we decouple the service a lot because one of the core principles SOA is loose coupling and autonomous services, so that reduce the dependency between the different component of an end to end business process, so that makes software a lot easier"*. By analysing the dependency between services and creating autonomous deployable packages, the Airline reduced the overhead of testing and change management processes: *"one of the major learnings we had is with the deployment architecture, so we used to have a single package that everything goes into that. ...we always had a lot of pain since SOA has a lot of moving parts you will always have issues within one particular application impacting the entire SOA stack, so the key learnings ... we understood the dependencies and we set it up in a way we have actually done a logical segregation of all those services and also we have deployed them accordingly so services don't impact the others".*

The creation of independent packages explained above was initially coded as 'Modular deployable', however after further review of the codes, this was merged with Autonomy due to its focus on the creation of boundaries with minimum dependency on other services.

Another characteristic in the structure-centric theme is 'Loosely coupled'. This characteristic is frequently reported in the IT literature and is usually associated with modularity. It is the degree to which changes within a subsystem does not create a ripple effect in the behaviour of other parts of the ecosystem (Tiwana et al., 2010). A loosely coupled module hides its implementation from the service user, encapsulating its implementation (Erl, 2004; Legner & Heutschi, 2007; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003; Schulte et al., 2008). The analysis of data across all cases reflected the 'loosely coupling' as a key characteristic of SOA and services, as shown below by one of the Bank architects: *"So, I would say that largely it's - in terms of reuse, it's the fact that our services are largely decoupled.... So, that's why we get lot of reuse in the sense that, the coupling is not hard, it's loose, so we can reuse these things over and over"*. Similarly, from one of the Airline IT managers: "*here there's a system called departure control which manages the check in and boarding of the passengers to the aircraft. And actually at that time we were changing from one system to a totally different one but by using an SOA approach, ... we were able to adapt between the two systems as we migrated from one to the other... it does abstract what you actually trying to do from particular source systems or ways of doing things. So some of the backend systems could change without impacting the frontend systems... that you haven't changed the contract with the consumers from the providers*".

The five characteristics described above, that is, Granularity, Generalisability, Service Cohesion, Autonomy and Loosely coupled, reflected the design characteristics of a service. The next characteristic describes how services together must be structured to increase the services reuse and IT responding capability. ‘Hierarchical Layering’ characteristic emerged from the data collected from the Airline and the Bank. In both of these cases, the services have been grouped into different layers with highly reusable services positioned at the lower layers, and those services, which serve the clients and different customer engagement channels such as web and mobile, were positioned at the higher layers.

While the creation of such structures was not initially by design, these structures proved to increase the level of reuse of core services as well as providing the business with the flexibility to cater for different requirements of the customer channels. At the Bank, the underlying layer was called DGW (Direct Gateway), which exposed the core banking services to the rest of the applications and systems in the organisation. Access to core banking system was only available through services exposed on the DGW to apply security, auditing and other essential services. Any developments on DGW was through a central team governed by the architecture team. On top of the DGW, a new layer called BSL (Business Service Layer) was developed whose task was to mainly serve the digital initiatives. Due to the nature of their projects, the digital team required additional agility and consequently, a decentralised development capability. The digital team engaged external suppliers and other teams in order to expedite their software development efforts. The role and level of agility DGW and BSL introduced can be observed in the comments below from one of the Bank architects: “*A very interesting finding that we found over the years. We started off with putting everything centralized and we even put in facade and subsystems in DGW but we found that when everything was in one system, everything just got done one way and there is a proliferation of service operations or service method calls. And what we found was actually the right way of doing things was to have a separate layer up on top and then the core service layer. So one service operation it does one thing and then have a orchestration layer up on top, ideally a separate system to forces developers, to actually make the distinction, to reuse those core services, and what we found was that combination is the best - has the best characteristics or best possibility of reuse.”*.

Considering the high level of reusability of the core services, the core services were designed to be fast and interacted possibly with one backend system only: “*The characteristic is for core services is- does one thing at the time. Does something really* *fast, so ideally under two seconds ... it doesn't involve a lot of data returning from it, ideally it only talks to one system and one system only, and that's pretty much it, so something really simple in terms of the core service.".*

The next layer orchestrated and called the services in the DGW. It facilitated what the front-end or the channel required: *"The next layer up is basically where we - what it does at, the only thing it can do is call other services, ... So, the services is the orchestration layer is used specifically targeted more for each channels, for example, a phone channel like do things, by orchestrate things a little bit different, from a web channel. So, these combinations, these orchestrations will be slightly different. Now what we found in the past was, if everything was collapse to one layer, you have a proliferation services but what you do is that if you extract that layer out, then yes you might have different, so you might have a different of those services but underneath you still reuse is very, very high when you have that centrally."*

The separation of layers is also based on the rate of change applicable to the services in each layer. The core services are closer to the fundamental business services, in this case, core banking services, which change at a slower rate. The layer that services the channels, however, goes through much faster rate of change due to technology evolution. The above can be observed in the following comment: "*...the layers that changed quickly and the layers that don't change as much so, classic examples, move money, move money really, really, really doesn't change all that. Okay. And however, the way to access move money, does. So, five years ago, smartphones didn't even exist. But now, we are talking Google Glass and watch and all that".* The Digital project manager at the Bank made similar comments: "*So business layer, everyone changes and reuse becomes less, whereas the reuse of the DGW is much higher but the level of change is less".*

Similar observation was made at the Airline. In this case, the loyalty group created their service layer to expedite their delivery through engagement of different teams. They, however, had to call the corporate services for core services, the latter was managed and governed centrally. One of the Airline project managers explained this: "*...typically all the services are consumed by websites and mobile applications…. with our first release of the mobile application for frequent flyer we built effectively an additional services layer on top of it [corporate service layer] that all the mobile channels now use, and that was to facilitate logging and mention of tokens because we didn't have anything to do that so we've got like another service layer that is totally managed by loyalty and… it's* *not part of TCS [vendor managing corporate service layer] it's meant for our house."*

The Airline CTO reflects the differences between the services of each layer: "*You do, you do, you have to because when your heavyweight transactional services typically residing in enterprise you know in-house, they're big to have you their large amounts of data that are you know make work intensive, whereas you know mobile demand like the protocols you want small and more fine-grained services, you want to be able to combine those and aggregate them and you want to do that without having those business rules embedded in the mobile applications. So we've engineered a different set of service, we need to think about different and you know so the different, they're certainly different but importantly they achieve, and you need to do that if you want to achieve the business like…"*.

The analysis of the above data reveals that:

- The layering of services based on their proximity to the core systems, such as core banking or underlying data repositories, is associated with their expected level of reuse.
- The lower layers will contain the more generalised services whereas the higher layers will be more specific to the channels and the specific business use cases.

    Lower layers are expected to go through less change than the higher layers that carry business rules and channel specific requirements.

The opposing case was the Airport, which had an additional layer (ECG) for external communication, however the distribution of the services did not follow the above structure. In the case of the Airport and the other business units within the Airline and the Bank, the projects had limited options and were constrained by the existing development teams. The above constraints and their limited options affected their responding ability.

Finally, 'Composability' is one of the characteristics reported in the SOA literature as the ability to invoke independent services with well-defined interfaces in defined sequences with minimum effort in order to form business processes (Erl, 2004; Henningsson et al., 2007; Newcomer & Lomow, 2004; Offermann et al., 2009; Vitharana et al., 2007). The data analysis identified this characteristic in the Airline and the Bank cases. In both cases, this characteristic was associated with the ability to quickly develop orchestration of independent services with well-defined interfaces in defined sequences to form business processes. The Airline used an industry standard for service orchestration (BPEL), whereas the Bank orchestrated their services using the standard .NET programing. The value of composability is to reuse core services to build services in the layers closer to the user channels or user interface. The Airline CTO highlights this: "*you want the core services to be able to inherit and append, you also want to aggregate and combine atomic services at the right scale and so actually the non-static entities you reuse them in different ways, aggregate them…So the reality is and that change is embraced, so it gives you speed to market, so it gives you a cost saving*". This phenomenon was also observed in the case of the Bank in their Business Service Layer (BSL) where they orchestrated services to accommodate the channel requirements. Furthermore, the Airline reported additional speed benefits due to the visual presentation and automatic code generation capabilities that their SOA platform provided them with: *"A BPEL type tool that you could visually connect things together in a bit of flow and that enabled you to define a high level service that aggregate a few other services to do its job. The tool helps you to reuse, because you can drag the definition of the service and can connect to other services and get the outcome in a visual way. Appeal was simplification from development side."*. Similarly, "*we could quickly build up services is purely because it's mostly just drag and drop and then do some minor configurations*".

The above comments indicate that composability as an SOA characteristic is the ability to orchestrate services quickly and easily. Such capability expedites the delivery of new services.

5.3.2 Flexibility-centric theme

The 'Flexibility-centric' theme emerged from four SOA characteristics 'integration', 'loosely coupled', 'standardisation' and 'framework driven'. These characteristics provide flexibility to make a change in an SOA system.

The 'Integration' was the first characteristic identified in this theme. The analysis of data collected from the cases provided a similar description to what IT literature (Byrd & Turner, 2001; Papazoglou, 2008) reported for 'Integration', which is the ability and the extent of seamless access and interactions between systems and services internally and externally outside the organisation. The IT literature considers two underlying factors for Integration. These are 'connectivity' and 'compatibility' (Duncan, 1995). The analysis of data reported connectivity an important factor in facilitating integration and responsiveness to business needs. The Airline IT manager reflects on how their ESB platform facilitated the integration: "*our primary use of ESB will be in integration by a long margin, we have introduced some process automation more recently, which had to integrate to Emirates airline. We have added some new products for example chauffeur pick up for business class passengers, and there is a Dubai overnight stay hotel where it might be available to certain passengers under certain circumstanced. So that process automation around providing those products and integration with Emirates themselves … is now being done on our ESB technology.".* Similarly, there are points on some of the functionality available in the ESB to facilitate the connectivity: "*Reliable messaging is the main thing that would drive us onto some shape of infrastructure there. That's the…that's probably the big one*".

The IT literature defines compatibility as the ability to share any information across any technology component (Byrd & Turner, 2001). Such ability to share information relies on two systems being standard-base or have functionality to adapt quickly. The aspect of being standard-base has already been covered in the 'Standardisation' node. The 'Adaptability', however, was reported by the cases as a characteristic of the SOA platform, which facilitates transformation and adaptation of existing services to meet new requirements (e.g. new service definitions of consumers) and to minimise changes in the systems. This is visible in the Airport architect's comments: "A*n ESB gives you a lot of choices because you have protocol independent, you can start the process off with a new protocol you want, as you say, we can see data from one point into another, we can split data, we can do all sorts of things. I think in terms of choice points, it gives you a lot*". Similarly, the Airline delivery and support manager reflects on their capability to create adapters to accommodate the differences in system APIs: "*The Airline is using a lot of products so they have different API's … but we have a standard way of sending events… so we build a proper adaptor, so we don't have to make changes to the core systems. We build services on top of the data services... that actually saves a lot of cost and effort and time.*"

The above comments reflected that the 'Adaptability' provided by the ESB enabled the Airline and the Airport to support integration with different products or systems quickly. Such option to quickly adapt between different protocols and message formats facilitated the implementation of the required change in the systems, hence improved the IT responsiveness. The opposing case was the Bank which used a custom-built SOA platform. Their platform had limited integration capabilities, which increased the effort to integrate different systems.

The second characteristic, which provides flexibility to implement operational changes is 'Loosely coupled'. This characteristic has previously been reported under the structure-centric theme due to its structural implication on how services were designed to minimise the impact of change on other services and systems. The data analysis, however, revealed that 'loosely coupling' also provides flexibility to swap and change of one system or service with another. Not only this requires the abstraction of services by hiding its implementation, but it also requires location transparency. One of the Airline IT managers highlights this: "*Yeah because it gives you transparency of not only what the service looks like but where it is….Having a service layer around things does open up an option. Hosting things in different places or using off-the-shelf server as a service rather than an in-house."*. This option has provided the Airline with flexibility for their DR: "*Whereas in SOA if one of the services is having some issue we can quickly just move that along to a DR it may not be the whole server or the dependencies we have this isolations, so if you have some issues just turn it off and broad and turn it on in DR just to ensure that it connects to all the relevant components."*. The Bank also reported a similar experience with their DR strategy and their environment management (Bank in the box project). They explained that the SOA and DGW abstraction enabled them to implement these initiatives: "*Abstraction and decoupling. It does that beautifully for us. And it's a funny thing but it's - it has put constraints on our organization but does constraints have allowed us, so many benefits, simplify and then from that simplification do something like bank in the box."*. Similar findings were reported by the Airport, where the replacement of their major airport operational system (AOS) happened with minimal impact on the downstream systems due to the SOA and ESB abstraction. As highlighted above, the SOA platform can provide additional loosely coupling, particularly in respect to the location transparency.

The third characteristic, which provides flexibility is 'Standardisation'. The data analysis revealed that 'standardisation' not only contributes to the 'knowledge' (hence being coded under the information-centric theme), it also provides flexibility in implementing changes. This can be observed in the comment below from the Airline: "*so the schedule distribution service I talked about before. So there are a lot of consumers, and there are standard formats for that, but the next consumer that comes along will fit into one of those formats and use one of those."*. Similarly, this reflects how standardisation and patterns on the SOA platform provided consistency and faster delivery: "*SOA is based on the common infrastructure, so that includes a lot of different various business processes and business rules, so in a traditional world every business function or business process has their own application, so they have different, you know, non-consistent monitoring processes or at least we have very similar duplicate monitoring process but now we're using centralized monitoring and auditing capabilities and also because the SOA infrastructure come with a very strong and rich security mechanism, so we also standardize the security part and auditing log in part, so make more application follow the consistent in a standardized way in doing things.*".

As the above data analysis indicated the 'ability to integrate' through 'Connectivity' and 'Adaptability' capabilities, reliance on standards at all layers and loosely coupling of services provided flexibility to make swift changes to the systems. Such characteristics can be facilitated through an SOA platform as indicated in the studied cases.

The last characteristic revealed from the data analysis was 'Framework-driven'. Framework-driven, which is associated with the SOA platform, provided the development teams with a set of built-in functionalities, which were based on established design patterns. One of the Airline architects explains this: "*So, in a traditional, in a way, whenever there is a new business requirements, they have the built application to automate that, but with SOA in place is they can actually orchestrate different existing services, it's new capabilities, and the leverage existing non-functional requirements which actually take a lot of time in traditional applications that are transaction security, auditing … yeah, logging part as well*".

Framework-driven expedited the delivery of the system and centralisation of certain functions.

5.3.3 Summary SOA characteristics affecting IT Responding capability

The second research question that this section addressed was:

- What are the SOA characteristics that affect the responding component of IT agility?

The data analysis presented in this section revealed two themes of SOA characteristics that contributed to the IT responding capability. The first group was 'Structure-centric' characteristics, which described a set of characteristics for a service and their underlying structure that have impacts on the IT responding capability. There were also reports on complexities created in the IT processes due to dependencies between services.

The second identified theme was 'Flexibility-centric'. This theme, which included both SOA design and SOA platform characteristics, dealt with flexibility measures to apply changes to the systems.

Figure 5-2 presents the above two themes and the SOA characteristics that emerged from the data, with their description listed in Table 5-8.

Structure-centric

- Granularity
- Generalisability
- Service Cohesion
- Autonomy
- Loosely coupled
- Hierarchical Layering
- Composibility

Flexibility-centric

- Integration
    - Connectivity
    - Adaptibility
- Loosely coupled
- Standardisation
- Framework Driven

Figure 5-2 - SOA Characteristics affecting IT Responding capability

The description of the SOA characteristics was initially based on the a priori construct definitions, listed in Appendix C . As the data analysis progressed, these definitions were refined based on the case and cross case analysis. A few new characteristics also emerged from data, which were added to the model.

Table 5-8 - SOA Characteristics affecting IT responding capability

Themes	SOA Characteristics (node)	Description
Structure-centric	Granularity	The degree of functionality embedded in a service based on the number of tasks it handles, the amount of data it processes and the number of external interactions it has.
	Loosely coupled	The degree to which changes within a subsystem do not create a ripple effect in the behaviour of other parts of the ecosystem (Tiwana et al., 2010).
	Generalisable	Ability of a service to address several use cases rather than a specific use case.
	Service cohesion	The degree the operations of a service having related functionality.
	Autonomy	The degree the logic governed by a service resides within an explicit boundary. The service has complete autonomy within this boundary and is not dependent on other services for the execution of this governance (Erl, 2005; L. O'Brien et al.,

		2007).
	Hierarchical Layering	The degree the services are spread in different layers, with generalised and core (close to data or business core systems) services positioned in the lower layers and services more specific to channels and business requirements positioned in the higher layers.
	Composability	Ability to quickly develop orchestration of independent services with well-defined interfaces in defined sequences to form business processes
Flexibility-centric	Loosely coupled	The degree to which changes within a subsystem do not create a ripple effect in the behaviour of other parts of the ecosystem (Tiwana et al., 2010).
	Integration	The ability and extent of seamless access and interactions between systems and services internally and externally outside the organisation.
	Standardisation	Formulation and implementation of consistent rules, guidelines and specifications for common and repeated use and application of services.
	Framework Driven	Predefined sets of functionalities built based on design patterns available in the SOA platform for reuse.

It is important to note that the IT literature also reports modularity as one of the characteristics of flexible systems with its two dimensions of loosely coupled and standardisation (Duncan, 1995; Fink & Neumann, 2009; Tiwana et al., 2010). In the pattern coding, these two dimensions have been identified in the data. The modularity has not, however, been added specifically because a 'service' is a modular component with additional characteristics as outlined above in this section.

The next sections will describe how the identified SOA characteristics create IT agility.

5.4 Impact of SOA on IT Sensing Capability

Sections 5.2 and 5.3 answered the first two research questions of this study by identifying individual SOA characteristics that affect the IT agility. This section explores how the identified SOA characteristics impact the IT sensing capability.

The approach taken to develop such explanation is the Yin's (2009) technique that relies on initial theoretical statements and their refinements as the data analysis progresses.

The initial theoretical propositions as shown in the research framework, Figure 5-4, were:

Proposition (R1): *The knowledge-based options are positively affected by the SOA characteristics already embedded in the information systems.*

Proposition (P1): *The IT-sensing capability is positively affected by the knowledge-based options that SOA offers.*

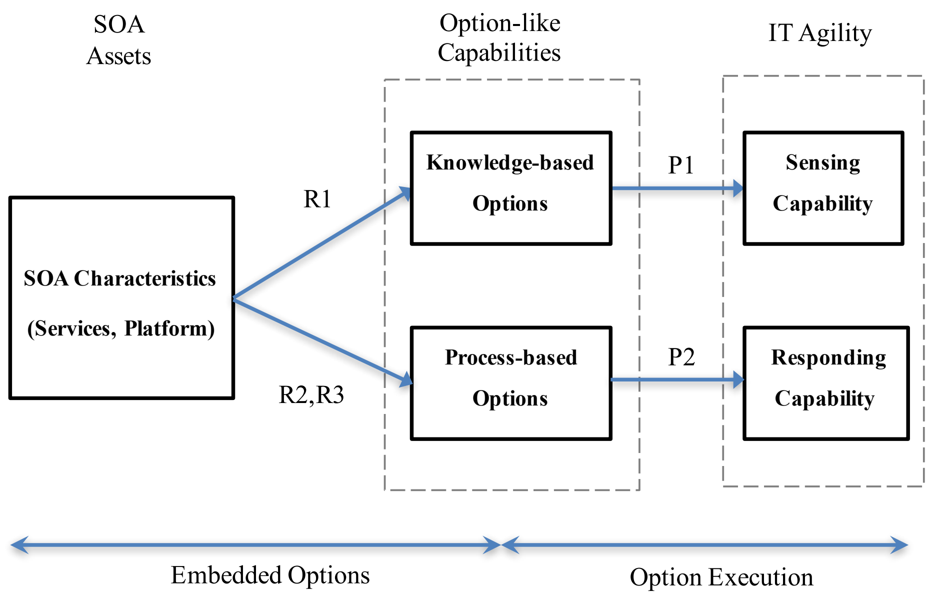


Figure 5-3 - Research framework of the study

Analysis of the data, while supporting these two initial propositions, provided further insight and mandated refinements to the above propositions.

The studied cases revealed that the SOA characteristics had an impact on the effectiveness of knowledge options, with effectiveness being the extent the option can accommodate the change requirements (Rahrovani & Pinsonneault, 2012). This impact was visible in the data analysis in the form of the range and richness of the options that SOA provided to sense changes in the information systems. The range of knowledge options was associated with the coverage of knowledge options that SOA facilitated, whereas the richness focused on the quality aspects such as timeliness of the option, its accuracy and relevance to the change event that has occurred.

Subsection 5.4.1 reports results of the data analysis on how SOA characteristics impacted the effectiveness of knowledge options (Proposition R1). Subsection 5.4.2 then reports how knowledge options improved the IT sensing capability (Proposition P1). And finally, the last subsection summarises the findings, proposes a few refined propositions and suggests a conceptual model for the impact of SOA characteristics on the IT sensing capability.

5.4.1 SOA creates Knowledge Options

The analysis of data revealed that the SOA through the 'information-centric' characteristics affects two types of knowledge options, which are ‘Change Detection’ and ‘Shared Insight’ options.

This subsection introduces the ‘Change Detection’ and ‘Shared Insight’ options. Table 5-9 summarises their definitions as well as their child nodes. After defining the knowledge options, the relationships between information-centric SOA characteristics and the identified knowledge options are discussed. Table 5-10 lists all identified relationships as the data presents them before the rest of the subsection provides evidence for each relationship in individual tables.

Knowledge options definitions

The analysis of data revealed ‘Change Detection’ in the information systems and ‘Shared Insight’ created from SOA ‘Information-Centric’ characteristics two knowledge options. Due to their potentials for use or abandoned, the ‘Change Detection’ and ‘Shared Insight’ are considered knowledge options. The creation of these options relies on the organisation making an initial investment on the SOA, such as implementations of monitoring and notification systems. However, such investment only creates choices for future use and adoption. If the ‘Change Detection’ and ‘Shared Insight’ options have been embedded effectively at the time when required, the IT and business can decide to invest further and use them. For instance, the support team will invest their time to attend the system notifications and analyse the detected issue. The decision to adopt and use the options depends on the effectiveness of options to accommodate the change requirements. Furthermore, the effectiveness of options is dependent on the range of options available and their richness: (i) their relevance to the change requirements, (ii) their timeliness and (iii) accuracy. In the case of monitoring and notification, the richness involved the notification being timely, correctly reflective of the system health, and its relevance in the form of significance.

Table 5-9 describes the ‘Change Detection’ and the ‘Shared Insight’ options with their child nodes, as identified from data reported below.

Table 5-9 - Knowledge Options created by the SOA information-centric characteristics

Category (Parent node)	Child node	Description
Change Detection	Change Detection	Notification to IT and business on an event that would possibly cause a change in the system (*Sufficient detection in all cases)*
Shared insight	Shared and clear understanding of required changes within a system to accommodate the current and future requirement	
	Capability Visibility	Shared awareness within IT and with Business on the current IT capabilities, their gaps to accommodate the current change requirements and on future required capabilities (Airline IT & Loyalty, Bank IT & Digital, Airport IT literal replication and other business units theoretical replication due to their low level of capability visibility)
	Knowledge sharing	Sharing of IT technical knowledge within the team and peripheral knowledge which participate in addressing the change requirements (Airline IT & Loyalty, Bank IT & Digital, Airport IT literal replication and other business units theoretical replication due to their low level of knowledge sharing)

The 'Change Detection' option, as emerged from the data, was associated with the notification to IT and business on an event that would possibly cause a change in the system. This option as its description suggests is outward-looking and focuses on the detection of events that can trigger a change inside the system. By monitoring the level and location of message failures, the Airline leveraged off this option and triggered an action for their support team. Similar observation was made at the Airport. Such option allowed the IT to be more proactive in sensing the change. All cases had sufficient change detection capabilities. The change detection was however limited to the health of services and the business processes were not monitored.

The 'Shared Insight' option, instead, involved in assessing the changes required inside the system after its need was detected. The data analysis revealed that ability to comprehend the change within the system was essential to preparing actions required to respond to the change. 'Shared Insight' relied on two contributing factors to assess the change: (i) the level of knowledge sharing within IT and business, and (ii) the visibility of IT capabilities. The knowledge sharing, particularly within IT was visible in all cases. The Bank benefited from this by their ability to augment additional resources to the team due to the ease of knowledge sharing created by the SOA standardisation. All the cases also reported that the IT team gained business knowledge. However, the IT knowledge within the business units was only observed in certain projects, as reported in Table 5-9.

The *capability visibility* was another aspect that was only identified within the Airline loyalty and the Bank digital teams. At the Bank, the capability visibility provided the IT and the digital teams with an opportunity to discuss the current capability options already available within IT, the capability gaps and the level of effort required to build such capabilities. Visibility on the IT capabilities allowed the IT and business to make a better decision with respect to the requirements and the options that must be taken to reduce the time to market. The Airline also reported visibility on future capabilities and particularly the potential reusable services by the emergence of common requirements in multiple projects.

Relationships between SOA information-centric characteristics and knowledge options

The data analysis revealed that the SOA information-centric characteristics impact the 'Change Detection' and 'Shared Insight' options. Table 5-10 summarises these impacts by outlining how each of the SOA characteristics has impacted the knowledge options. 'N/S' indicates that between the SOA characteristic and the knowledge option 'No Significant' relationship observed in the data. The tick (✓) indicates there was a significant relationship between the identified items, as observed in the identified cases. Each relationship is individually discussed in the referenced tables as specified in each cell. For example, Table 5-11 provides the evidence of the relationship between information dissemination characteristics and change detection options.

Table 5-10 - Impact of SOA on knowledge options

Category (parent node)	SOA Characteristics (child node)	Knowledge Options Effectiveness		
		Change Detection	Shared Insight	
			Knowledge sharing	Capability Visibility
Information Dissemination	Service Monitoring	✓ (Table 5-11)	N/S	N/S
	Event Driven	✓ (Table 5-11)	N/S	N/S
Information Repository	Standardisation	N/S	✓	✓

			(Table 5-12)	(Table 5-13)
	Business Oriented	N/S	✓ (Table 5-12)	✓ (Table 5-13)
	Service Definition	N/S	✓ (Table 5-12)	✓ (Table 5-13)
Information Discovery	Service Discovery	N/S	✓ (Table 5-14)	✓ (Table 5-14)

As Table 5-10 shows, 'Change Detection' was facilitated by the SOA 'Information Dissemination' characteristics, and the 'Shared Insight' was impacted by the 'Information Repository' and 'Information Discovery' characteristics. While all the cases reported the 'Shared Insight' as an option created in the IT, it was only a few cases that reported the 'Shared Insight' in their business units.

To better explain the interactions between the SOA characteristics and the knowledge options, the mechanisms (Avgerou, 2013) involved in these interactions are reviewed below. The mechanisms are first listed in a table, similar to Table 5-11, followed by evidence from the studied cases. When the impacts of SOA characteristics are similar and jointly, they are reported together in the below analysis, e.g. 'Service Monitoring' and 'Event Driven' in Table 5-11.

The first relationship reviewed below is the impact of 'Information Dissemination' and 'Change Detection' as listed in Table 5-11.

Table 5-11 – Empirical evidence of how 'Change Detection' in impacted by the 'Information Dissemination'

SOA Characteristic (Theme / Characteristic)	**Knowledge Option**	**Mechanism**	**Reported Cases**
Information Dissemination \ Service Monitoring and Event Driven	Change Detection	- Automatic identification and alerting to support team when an error or abnormal condition detected (option richness and range) – (*AL & AP similar, BK complemented service monitoring by log analysis to identify issues*) - Identification of common patterns over time to trigger optimisation or change (option range) (*AL, AP and BK – BK used log processing to detect the patterns*)	AL, AL-P, AP, AP-P, BK, BK-P

The cross-case analysis revealed that service monitoring and event driven architecture improved the effectiveness of 'Change Detection' by providing the IT with timely notification of an issue. The timeliness is reflective of option richness, whereas the automatic detection is a factor of a range of options. The Airline IT manager explains how the A380 project achieved this: "*Built into the application was an alerting mechanism, so it would be self detecting problems and alerts which would then call in the support teams*".

The 'information dissemination' also provided a range of 'Change Detection' options including reactive action from IT only when an error detected or more proactive pattern detection on the service behaviour and optimisation of the service accordingly.

For instance, the Airline support manager reported how they sensed the change by checking the system behaviour: *"Talking about sense of change, ... so if you see some differentiation in the pattern that a specific SOA server is suddenly having some peak usage or increased usage, it's easy for us to see which service is consuming this and why is it happening…might be costing us this much"*. Similarly, one of the Airport IT managers reported similar observation: "*Yeah, [This shows] what are the airlines doing or what are other systems doing, why is it becoming more chatty, why has it a lot more operational stuff coming through that normally it wasn't like that. Have they changed systems that we need to be aware of? are we getting more accurate information that we can share with the other business"*.

All cases showed the relationship between the SOA 'Information Dissemination' characteristics and the 'Change Detection' options. The Airline (AL) and Airport (AP) relied on their platforms and the service monitoring built into applications to monitor the services. Their 'sufficient' level of service monitoring and 'strong' focus on event driven characteristic provided them with 'sufficient' change detection capability.

The Bank was, however, different. Although the Bank had average level of information dissemination (service monitoring and event driven), they achieved similar level of change detection to the Airline and the Airport. Review of the Bank infrastructure documentation revealed that the Bank improved its 'Change Detection' capability with a non-SOA capability to analyse the system log files and identify system issues. The use of 'Splunk' system for log analysis compensated their weakness in the SOA information dissemination.

In all cases, the monitoring of the service behaviour was only at the technical level and performed by the IT support team. Discussions were raised on the ability of the business units to analyse and optimise the performance of a business process, and none of the cases confirmed such involvement and such benefit.

The next knowledge option is 'Knowledge Sharing' and its relationship with SOA 'Information Repository' characteristics. Table 5-12 explains how these characteristics impact the 'Shared Insight'.

Table 5-12 – Empirical evidence of how 'Knowledge Sharing' in impacted by the 'Information Repository'

SOA Characteristic (Theme / Characteristic)	**Knowledge Option** (Theme / factor)	**Mechanism**	**Reported Cases**
Information Repository \ Standardisation	Shared Insight \ Knowledge Sharing	- Business IT knowledge: high-level design standards which are understandable to business and relevant to their needs. Standards created a joint vocabulary between IT and Business (option richness and range) (*Only observed in AL-Loyalty & BK-Digital projects)* - IT Knowledge: creating consistency in design and development and use of well-known framework (option richness and range)	For Business IT knowledge: BK-P, AL-P For IT Knowledge sharing: AL, AL-P, AP, AP-P, BK, BK-P
Information Repository \ Business Oriented and Service Definitions	Shared Insight \ Knowledge Sharing	- Business IT knowledge: services understandable to business and relevant to their needs. This shared understanding created a shared vocabulary between the business and the IT (option range and richness) (*Only observed in AL-Loyalty & BK-Digital projects)* - IT Business knowledge: gain business insight by services having business context and the function that the services performed (option range and richness). - IT knowledge: description of the services facilitates the knowledge sharing within IT (option range and richness)	For Business IT knowledge: BK-P, AL-P For IT Knowledge & IT Business Knowledge sharing: AL, AL-P, AP, AP-P, BK, BK-P

The impact of 'Standardisation' on 'IT Knowledge Sharing' was visible in all cases. Standardisation by establishing common and consistent design patterns and standard architecture improved the knowledge sharing within the IT. Such effect is clear in a comment from the Bank: "*That's right its consistent framework …because of that* *knowledge is shared around the team. There is a good support network within the team … so you are going to have some level of consistency and if one developer leaves another one has to do his support work continue the work yeah so it's that side of agility*". On the business units gaining IT knowledge, however, the data analysis revealed only limited cases, which the business units gained IT knowledge through SOA. As shown below, the two cases involved projects with the Airline loyalty team and the Bank Digital team. Cross-case comparison of the above two teams vs. the other cases indicated that these two teams had a much closer working relationship with the IT team. The Bank Digital team leveraged off an agile software development methodology, hence the business and the IT team were actively engaged across all the daily project meetings. Also, both the Bank Digital team and the Airline Loyalty team had much shorter time to market for their products and services, compare to other business units. The impact of 'Standardisation' and 'Business IT knowledge' with the Bank digital team can be seen in this comment: "*Business like they understand how their solutions work. So, for example, there is an SME in the business now. He has retained that knowledge …So, if the next change comes along, … the business guy remembers, he knows. Yeah, it goes to DGW and blah, blah, blah*". The use of DGW as a consistent architecture and design pattern provided the business SME with the IT knowledge on how the system operates. The above statement indicates that the 'Standardisation' improves the quality of knowledge sharing (option richness) by creating consistency between the shared knowledge. It also increases the coverage (option range) of knowledge sharing when more patterns and policies are documented and standardised.

The impact of 'Service Definition' and 'Business Oriented' on 'Knowledge Sharing' was similar to 'Standardisation'. The impact on 'IT knowledge' was visible in all cases, whereas the impact on Business units was only visible in cases of the Airline loyalty team and the Bank digital teams.

On the IT knowledge, the service definitions provided the IT team with knowledge on the available services. The Airport explains this visibility: "*Yes, so for any predefined services then it's much easier because it's already articulated in the contract, so that helps the project manager to understand what he can do or he cannot do or he doesn't have to repeat.*"

The above comment indicates that the 'Service Definition' has improved the knowledge sharing quality (richness) by providing an accurate presentation of the available services.

Also, the extent the service definitions are documented impacts the range and coverage of knowledge sharing.

Similarly, ‘Business Oriented’ improves the quality of IT knowledge (richness) by creating a business context for the IT and simplifying the knowledge sharing. The Bank’s reflection on this follows: “*So having those business service there is a business function as opposed to data that no one understands, see from clarity perspective it is pretty easy to understand card management … it is, is a customer language or business language*”.

On the Business IT knowledge, the Digital and Loyalty teams supported the impact of ‘Business Oriented’ and ‘Service Definitions’ on their IT knowledge, as described in these quotes: “*once with this service we know actually what…in the granular level what IT does.*” or “*they just know that this service provides this functionality …if they are having some thought of building something then there is a service available why can't I utilize it to quickly build up this thing*”. The Service Definitions that are business oriented created a common vocabulary for knowledge sharing and discussions: “*I would say, is significant because now we have a common vocabulary to talk about”.*

To summarise, the impact of ‘Information Repository’ characteristics on ‘Knowledge Sharing’ extends across both the options range and the option richness. The data analysis revealed that cases with high level of ‘Information Repository’ characteristics achieved high level of IT knowledge sharing, which explains the role of options range. The ‘Richness’ is explained based on the quality improvement (accuracy) of having the ‘Information Repository’ on the knowledge. Similar effect on Business IT knowledge sharing was, however, observed only in two cases of the Airline Loyalty team and the Bank Digital teams. The variation between cases indicates the involvement of other factors in the above interaction. Section 5.6.1 analyses and contrasts the cases to explain the above observation.

The other dimension of ‘Shared Insight’ is ‘Capability Visibility’. Table 5-13 explains how ‘Capability Visibility’ is impacted by the ‘Information Repository’ characteristics.

Table 5-13 – Empirical evidence of how ‘Capability Visibility’ in impacted by the ‘Information Repository’

SOA Characteristic (Theme / Characteristic)	**Knowledge Option** (Theme / factor)	**Mechanism**	**Reported Cases**

Information Repository \ Standardisation, Business Oriented, Service Definitions	Shared Insight \ Capability Visibility	- Shared understanding of IT current capabilities (services and architecture), reducing the communication overheads (Option range and richness) - Shared understanding of capability gaps to achieve the requirements (Option range and richness) - Identification of common service requirements in different projects provide visibility of future services (Option range and richness) (*Observed in AL)*	AL, AP, BK *(IT departments only)* & BK-P, AL-P (*Only observed in AL-Loyalty & BK-Digital projects)*

The data analysis revealed that in the Bank Digital projects and the Airline Loyalty group projects, SOA Information Repository provided the IT team and business with a shared visibility of IT current capabilities and its limitations. The Bank architect reflects on the shared understanding and how that reduces the communication overhead: "*we can start talking IT capabilities, much easier to talk about and then basically people have a much better feel of what IT can and can't do... We are finding the digital actually - it actually helps the business much more because now they understand ... this is the stuff that SOA - this is the benefit that lot of times, is not really talked about ...once the business understands, what they could ...you start seeing things from the start, making sense and just less problem, just less communication in the first place, which is awesome, which is basically that means that understanding is there. Yes*".

Similarly, the Bank Digital project manager explains the Digital team's visibility on the available services: "*They (business) understand actually the services that are available, e.g. card management ... or money transfer or things already there .... They know what services are, what services are available, what services are not".*

As shown in the above comment, the 'Information Repository' also provided the business with visibility on the capability gaps and the effort involved to address these gaps. This allows the IT and Business to assess the available options based on their level of effort and risk involved. This argument can be seen in the following comment from the Bank: "*how long it is to build that capability, to build the service? ... 50% of the services don't even exist. ... it's going to take you forever to build this. what you find is a lot of analysis that BAs would put, gets dropped because it's just too hard. Remember they are trying to hit the targets this year, not two years or three years from now.*"

The above finding was also reflected in the Airline case. For instance, the Airline Project manager explains how the Loyalty team discusses with IT different options to achieve their goals: “*Well, they do (understand services), …, typically they will come through the IT function to say okay, we need to this. Do we have a web service that can do this? Do we need to change anything? If we want to do this function, can it be in just two days …I’ve heard there’s a service to do it. I know that there are services that should be accessed quickly and easily*”.

The last SOA characteristic, which its impact on ‘Shared Insight’ has been summarised in Table 5-14, is ‘Service Discovery’.

Table 5-14 – Empirical evidence of how ‘Shared Insight’ in impacted by the ‘Information Discovery’

SOA Characteristic (Theme / Characteristic)	**Knowledge Option** (Theme / factor)	**Mechanism**	**Reported Cases**
Information Discovery \ Service Discovery	Shared Insight \ Knowledge Sharing and Capability Visibility	- Easy access to service definitions through classification and search functions (option range and richness) (*AP was the opposing case)* - Publish of business-oriented service definitions and standards (information repository) – (option range) (*AP was the opposing case)*	AL, AL-P, BK, BK-P AP

As shown in Table 5-14, the impact of ‘Service Discovery’ on shared insight lies in its facilitation of access to the ‘Service Repository’ such as ‘Service Definitions’. Such facilitation increases the range of ‘Knowledge Sharing’ and ‘Capability Visibility’ within the organisation by making the information more accessible. This can be observed in the comment from the Airline architect: *“all of our service is also classified and categorized by different taxonomies…So the developers or solution app are very easy to find them for example, I need an operational type data then I know okay under that category what kind of service available for me to use. And for the business activity process it’s categorized based on the business capabilities”.*

The comparisons among cases showed that the Airline had a better service discovery facility with information being classified in a meaningful manner. The Bank also had their repository, limited to technical details of the services. The Airport, however, did not have a service discovery facility. In none of the cases, the service discovery was exposed and accessible to the business, even though the Airline was planning to provide such access to the wider organisation: "*Sometimes business teams ask to look into the repository, but still is relatively early days and we need to give more visibility to business*".

The cross-case comparisons reflect the difference in the access to information and consequently the level of knowledge sharing and capability visibility. The Airline IT team was able to find and analyse their services available in their service discovery (Catalogue) to identify their existing IT capabilities: *"And it helps you go to find the services and compare the existing services with what the product needs to do and yes so having a repository is a must".* In comparison, the Bank had the registry, but lacked certain functionalities which affected their knowledge sharing and capability visibility: *"So we do have this registry of services. If that was enhanced in some way people could search it better. ... because of again the way the services are exposed and we have some kind of service registry where you can see all the services and what they do to some extent".* In contrast to the Airline, the data analysis revealed that the Bank has experienced a high level of service duplication with similar functionalities. The Bank development manager explains it: "*many services that do more or less the same thing and then if there is a new front end activity you may not know which is the correct one to use if they you have a dozen services which are named in a very similar way that look like they may provide the same business function.*".

In an interview with one of the Airport architects, he referred to another organisation which he worked for. With this particular organisation, they had a 'Service Discovery' (Wiki) accessible to business and populated with relevant service information. They use the service discovery to gain 'Shared Insight' on their existing services and facilitate their planning activities. This was particularly important for them due to the number of services they had to avoid service duplication and promote service reuse: *"...a pharmaceutical company and then they have got a quite well done wiki that they used to record the services… That hit the business to my knowledge… I think we got to point to wiki and search for a particular term or search for something, and we may have a service that's 80% they are ready, why don't we reuse them. So, the agility side, we are saving the business, the 80% of the development cost.".*

These contrasting results make the proposed relationship between the 'Information Discovery' and 'Shared Insight' more generalisable.

In respect to the type of relationship between the 'Information Discovery' and the 'Shared

Insight', the data analysis suggests that the impact of 'Information Discovery' on 'Shared Insight' is dependent on the 'Information Repository'. Such dependency is visible in the described mechanisms listed in Table 5-14. As explained in the mechanisms, the 'Information Discovery' facilitates access to the 'Service Definitions'. As such, if there is no 'Service Definitions' and 'Standards' available ('Service Repository' non-existence), the 'Service Discovery' on its own does not create any Insight for the business or the IT. Whereas, if the services definitions are available and well documented, 'Service Discovery' makes their impact on 'Shared Insight' much higher due to its ability to make the 'Service Repository' more accessible. Based on these evidence, it is concluded that the 'Service Discovery' performs a moderating role for the relationships between 'Service Repository' and 'Shared Insight'.

To summarise the above findings, the data analysis revealed that:

- In all studied cases, the SOA 'Information Dissemination' characteristics positively impacted the effectiveness of 'Change Detection' options by improving the timeliness and range of events that could identify the needed change in a system.
- In certain cases, SOA 'Information Repository' characteristics positively impacted the effectiveness of 'Shared Insight' option by improving the accuracy of shared knowledge and capability visibility of services, as well as the extent the available services are known.
- In all studied cases, the impact of SOA 'Information Repository' characteristics on 'Shared Insight' effectiveness is moderated by the SOA 'Information Discovery' characteristic.

5.4.2 Knowledge options improve IT sensing capability

The previous subsection presented how SOA affected the knowledge options. This subsection focuses on the interaction between knowledge options and IT sensing capability that covers the second half of the research framework shown in Figure 5-4.

As shown before, the impact of SOA on IT sensing capability was through two options of 'Change Detection' and 'Shared Insight'. Table 5-15 summarises the impact of the two knowledge options on the IT sensing capability.

Table 5-15 - Impact of knowledge options on IT Sensing Capability

Knowledge Option	IT Agility Dimension	Mechanism	Reported Cases
Change Detection	Sensing capability	- Timely identification and notification of a change (e.g. issue) to relevant parties	AL AL-P AP AP-P BK BK-P
Shared insight	Sensing capability	- Alignment between business and IT on how IT operates, IT existing capabilities and the work that must be done to achieve the solution in a timely manner - Improved communication and shared problem-solving to agree on the required change (Reduced time) - Shared understanding and common vocabulary resulted in clear requirements understandable to IT - Focused discussions on what matters (capability than User Interface) - Improve the relationship between IT and Business (trust and empathy) - Identification of common service requirements across multiple projects	AL AL-P (*Digital projects, Saving Maximiser high sensing, Living Super opposing case)* AP AP-P (*AOS, PA, Internet & CUTE average sensing - shared insight limited to IT)* BK BK-P (*Loyalty high sensing, A380 and Emirates average sensing – shared insight limited to IT, Blackberry opposing case)*

The impact of 'Change Detection' on IT sensing capability lied in the detection and altering of the IT team. The timely identification of the change enabled the IT team to assess quickly the change and start the responding action. As previously quoted from the Airline, their service monitoring allowed their support team to be called if an issue with the services detected: "*Built into the application was an alerting mechanism, so it would be self-detecting problems and alerts which would then call in the support teams*". Similarly, the following review of service usage pattern allowed the Airline to avoid a potentially high cost by early detection of the issue: *"When you talk of sense of change, … so if you see some differentiation in the pattern that a specific SOA server is suddenly* *having some peak usage or increased usage, it's easy for us to see which service is consuming this and why is it happening…might be costing us this much".* These comments indicate that both the richness (timely and relevance) and range (coverage of services) of 'Change Detection' option positively impact the IT sensing capability.

Finally, the impact of 'Shared Insight' on IT sensing capability is on multiple fronts. The 'Capability Visibility' as discussed before, provided a shared understanding of the IT capabilities and the associated gaps involved. Such common understanding improved the business and IT alignment on the options that must be adopted in the response to a change. This alignment improved the speed of assessing the options and selecting the best path forward. The Bank architect explains how this applied in the projects with the Digital team: "*As far as I am concerned SOA is imperative, in terms of agility. ... business already has a context, to say, yep, we can do this, and if it's not in this services, we can't really do this, .. what drives them (business) is time, time to market. .. There is targets that they have to hit this year. And SOA can certainly help filter a lot of low noise. For me what SOA does, is that filters all that noise out, allows us to get to the crux of the matter, can we do it. And then, from there, then there is a natural extension to then how do we actually do it".*

The reduction of communication overhead was also another aspect reported by the Bank: *"we talked about the differences between, business unit, that's the SOA aware versus the business unit is not SOA aware… when business is more SOA aware, the conversations becomes easier, the planning becomes easier. And the understanding - demands in IT actually less is because now, ..., service is there, you can find it out yourself. So, conversations that we are having is much less. So, the maintenance side of things is much less as well. So, again and best of all, it makes IT more transparent to the business".*

The improved knowledge sharing in the IT and the Business and their insight on the IT capabilities and gaps allow them to find solutions faster: "*the thing is that by (business) knowing the problem domains better, by knowing what IT can and can't do, in their heads, when they form the solution, it's actually a solution that is optimal for the time and the cost of that they are aiming at…. whereas the other team, they do not necessarily align, as a result they make assumptions, they go and capture all the requirements and then there is this big misalignment between what they are trying to achieve and what at the end IT capabilities that exist.".* Such knowledge sharing and collaboration on the available capabilities and the future solution has built empathy in the business and improved the relationship: *"It builds empathy in the business. So, when you can empathize with someone, you can understand them, you can entrust them, you can share their pain, you can share the success. So, when we say, we have got a capability for you but it's 80% there, they are like, okay, 80% there, I understand, I got to build the next 20. These are my constraints, look, it was designed in this way, for this need, okay. They are building empathy for the architecture expense. And that in it's own right, is building much stronger trust"*.

Also 'Shared Insight' provided the IT with the visibility of the common requirements for reusable services, which assisted IT to plan their implementation early before the change occurs. The Airline achieved this by building insight into their reusable services by reviewing their project portfolio and their potential service requirements: *"so you know IT we are able to sense that there are a lot of requirements for this information… and to say okay let's make sure we build something that's reusable and extendable…and you can easily add new subscribers to that information feed"*. The Airport and the Bank achieved this by anticipating their future requirements when implementing services for projects.

In all of the above cases, the impact on IT sensing capability was dependent on the range of Insight (coverage) and its quality (timeliness, relevance and accuracy).

To summarise the discussions above, the data analysis revealed that:

- 'Change Detection' effectiveness positively impact the IT Sensing capability
- 'Shared Insight' effectiveness positively impacts the IT sensing capability.

5.4.3 Summary of impact of SOA on IT Sensing Capability

This subsection summarises the findings concerning how SOA characteristics impact the IT sensing capability as presented in this section.

From the initial conceptualisation and the research framework shown in Figure 5-4, the following theoretical propositions were suggested:

> **Proposition (R1):** *The knowledge-based options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (P1):** *The IT-sensing capability is positively affected by the knowledge-based options that SOA offers.*

As observed in this section, while the data supported the above propositions to a large extent, there is also room for their refinement. The analysis of data revealed that only a certain category of SOA characteristics ('Information-centric') impacts the IT Sensing capability.

The data also revealed that the SOA characteristics that participated in the dissemination of information, such as notification on the system health, positively impacted that effectiveness of 'Change Detection' option. Such active knowledge on the conditions that could trigger a change in the system improved the IT sensing capability by allowing the IT to identify and analyse required changes. On this basis, the following propositions are suggested:

> **Refined Proposition (RP1-R1):** *The effectiveness of 'Change Detection' knowledge option is positively affected by the SOA 'Information Dissemination' characteristics already embedded in the information systems.*
>
> **Refined Proposition (RP1-P1):** *The IT-sensing capability is positively affected by the 'Change Detection' knowledge option that SOA offers.*

The above propositions are the result of iterative refinements to the initial propositions as the cases were compared and contrasted.

Other findings from the data were the effect of SOA 'Information Repository' and 'Information Discovery' characteristics on the IT and Business 'Shared Insight'. By providing information regarding the business oriented services and SOA standards, the 'Information Repository' characteristic improved the knowledge sharing and visibility of IT capabilities within the IT and with the business units. The impact of 'Information Repository' on 'Shared Insight' was however magnified to the extent that the 'Service Discovery' facilitated the access to the information published in the 'Service Repository'. More searchable and classified the service definitions were, better the 'Service Repository' could impact the organisation 'Shared Insight'.

Also the 'Shared Insight', created by capability visibility and shared knowledge, provided the IT and the business with faster and more effective alignment on the required system changes. This alignment improved the effectiveness of the selected option from the time and option feasibility perspectives. On this basis, the following propositions are suggested:

> **Refined Proposition (RP2-R1):** *The effectiveness of 'Shared Insight' knowledge*

option is positively affected by the SOA 'Information Repository' characteristics already embedded in the information systems.

Refined Proposition (RP3-R1): *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the SOA 'Information Discovery' characteristic.*

Refined Proposition (RP2-P1): *The IT-sensing capability is positively affected by the 'Shared Insight' knowledge option that SOA offers.*

Review of the studied cases showed that the above propositions could explain the variation between a few cases including the Digital projects and Loyalty projects. In other cases, however, the effect of 'Information Repository' was limited to the IT which reduced the level of sensing capability gained through the SOA. Section 5.6 reviews the effect of non-SOA factors in the current study to provide further explanation on the above observation.

Figure 5-4 presents these propositions in the form of a conceptual model (Miles & Huberman, 1994).

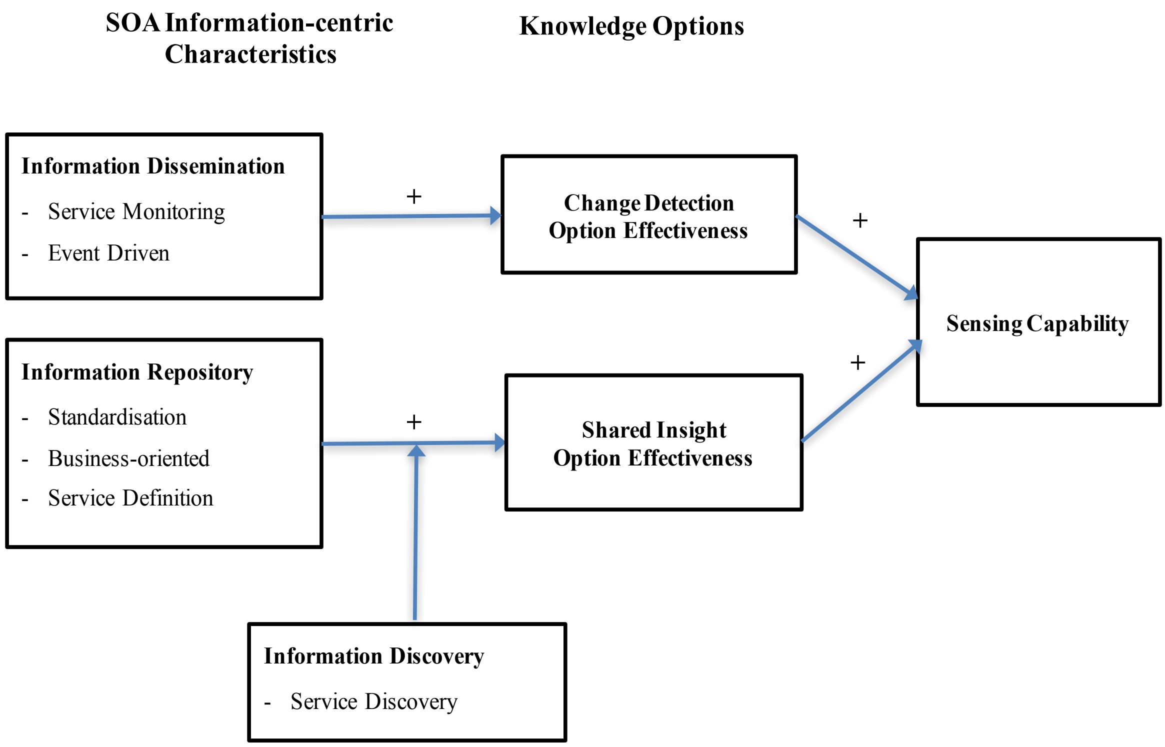


Figure 5-4 - SOA Characteristics impact on IT Sensing Capability Model

The arrows in the conceptual model show the associations between the defined concepts with their variation following the sign on the arrow. For instance, higher level of information dissemination results in more effective change detection which increase the IT sensing capability. As highlighted above, the current model does not consider the effect of non-SOA factors on the relationships between the SOA and the IT Sensing capability. Such relationships will be discussed in the future sections.

5.5 Impact of SOA on IT Responding Capability

The last research question of the study explores how SOA characteristics impact IT responding capability. This section addresses the above question by analysing the data collected from the studied cases and refining the following initial theoretical propositions shown in the research framework Figure 5-4:

> **Proposition (R2):** *The Operational options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (R3):** *The IT growth options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (P2):** *The IT responding capability is positively affected by the process-based options that SOA offers.*

The analysis of data, while supporting the above initial propositions, provided further insight and mandated refinements to the above propositions. Subsection 5.5.1 analyses the collected data and demonstrates how SOA characteristics impacted the effectiveness of process options (Propositions R2 & R3). Subsection 5.4.2 then reports how process options improved IT responding capability (Proposition P2). And finally, subsection 5.5.3 summarises the findings, proposes refined propositions and suggests a conceptual model for the impact of SOA characteristics on IT responding capability.

5.5.1 SOA creates Process Options

The studied cases revealed that the SOA characteristics impacted the effectiveness of process options in its two dimensions: (i) the operational options, and (ii) the IT growth options. In addition to the operational options and growth options, the data analysis also revealed a set of factors that mediated the effect of SOA on the process options. The above mediators are categorised as 'Options depreciators' due to the role they played in diminishing the effectiveness of the process options.

This subsection first introduces the ‘operational options’, ‘growth options’ and ‘options depreciators’. Table 5-16 and Table 5-17 summarise their definitions with a few examples as presented in the data. After defining the above terms, the relationships between SOA characteristics, the process options and options depreciators are discussed. Table 5-18 lists all identified relationships as data presents them before the rest of the subsection provides evidence for each relationship in individual tables.

Process options definitions

The operational options facilitated the management and maintenance of the existing systems. The impact on operational options was visible in the data analysis in the form of range and richness of the options that the SOA flexibility-centric characteristics created. The definitions of options effectiveness, options range and options richness are consistent with the definitions provided for knowledge options, as covered in section 5.4.1.

Similarly, the SOA characteristics: ‘structure-centric’ and ‘flexibility-centric’ impacted the effectiveness of growth options in the development activities related to service-oriented systems. The impact on the effectiveness of growth options was based on its effects on the range and richness of the options. The range of growth options was associated with the variety of options (e.g. number of reusable services) that SOA ‘structure-centric’ and ‘flexibility-centric’ characteristics facilitated. The richness, instead, focused on the quality aspects such as timeliness of the option (the time required for the option to be available for use), its accuracy (the quality of the option) and relevance (its fitness) to satisfy change events promptly.

Table 5-16 describes the growth and operational options with some examples, as extracted from the collected data.

Table 5-16 – Process options created by the SOA characteristics

Category (Parent node)	Description	Example Created Options
Operational Options	The options available to improve the response to changes required in the management and maintenance of existing service-oriented systems.	Gradual transition (Staged switch) Change service provider (Switch input) Distributed transparent sourcing (Switch) System replacement and decommissioning (Switch)

		Wrap and simulate behaviour (Maintain)
Growth Options	The options available to improve the response in processes involved in the development of service-oriented systems.	Extension by service reuse Extension by design reuse Extension by connectivity Extension of channels by reuse Extension by protecting Extension in stages

The 'Operational options' focused on the choices to maintain the existing service-oriented systems. An example of such options was 'gradual transition' of systems, as observed at the Airline when they migrated from one departure control system to another. The operational options did not involve any functionality change to the services.

The Growth options, instead, focused on the processes involved in the software development. One significant example of the growth options observed in all cases was 'extension by reuse'. All cases reported that service reuse improved the IT's ability to deliver to build or extend systems faster.

Options depreciators definition

Regarding the impact of SOA characteristics on the process options, the data analysis revealed a set of factors that mediated the effect of SOA on the process options. The above mediators are categorised as 'Options depreciators' due to the role they played in diminishing the effectiveness of the process options. The data analysis revealed high dependency between services and latency in the connectivity between services as two main 'Options depreciators' observed in the cases. The data analysis revealed certain SOA characteristics, while created future options, also created these 'options depreciators' which negatively impacted the effectiveness of process options. Table 5-17 lists the two factors identified from the cases along with their description.

Table 5-17 – Options Depreciators created by the SOA characteristics

Category (Parent node)	Description	Example observed consequences
Dynamic Dependency	The extent services are reliant on each others to complete their defined tasks in the system. A higher number of services involved in the execution of tasks reflects higher dependency.	Increased coordination effort Stakeholder impact assessment

Latency	The delay incurred in the connectivity between services to complete their defined tasks in the system.	System performance degradation

Analysis of the data revealed that there are two types of dependencies between components: (i) static dependencies and (ii) dynamic dependencies. Static dependencies are the structural dependencies between services, which cause a ripple effect in the dependent services when a change in a service occurs. As discussed in the previous sections, loosely coupling reduces the static dependency. However, the dynamic dependency is concerned with the relationships created between services when they are executing a set of tasks. For instance, the Bank card management service had to invoke a number of services to complete its function. The data analysis revealed that the dynamic dependency increases the coordination effort required for change management and stakeholder management.

The other options depreciator identified from the data was the 'latency' in the communication and connectivity between services. Latency reduced the timeliness of information and consequently degraded the system performance.

Relationships between SOA characteristics and process options

While SOA characteristics improve the effectiveness of process options, they also create options depreciators. Table 5-18 summarises the impact of SOA characteristics on the options depreciators, the process options and how options depreciators affect the process options. 'N/S' indicates that 'No Significant' relationship observed in the data. The tick (✓) indicates there was a significant relationship between the identified items as observed in the identified cases. Each relationship is individually discussed in the referenced tables as specified in each cell. For example, Table 5-19 provides the evidence of the relationship between flexibility-centric characteristics and operational options.

Table 5-18 - Impact of SOA on process options

Category (parent node)	SOA Characteristics (child node)	Process Options Effectiveness		Options Depreciators	
		Operational options	Growth options	Dynamic dependency	Latency
Flexibility-centric	Integration	✓ (Table 5-19)	✓ (Table 5-20)	N/S	✓ (Table 5-21)

	Loosely coupled	✓ (Table 5-19)	✓ (Table 5-20)	N/S	N/S
	Standardisation	✓ (Table 5-19)	✓ (Table 5-20)	N/S	N/S
	Framework Driven	✓ (Table 5-19)	✓ (Table 5-20)	N/S	N/S
Structure-centric	Granularity	N/S	✓ (Table 5-22)	✓ (Table 5-23)	✓ (Table 5-23)
	Generalisability	N/S	✓ (Table 5-22)	N/S	N/S
	Service Cohesion	N/S	✓ (Table 5-22)	N/S	N/S
	Autonomy	N/S	✓ (Table 5-22)	✓ (Table 5-23)	N/S
	Hierarchical Layering	N/S	✓ (Table 5-22)	✓ (Table 5-23)	N/S
	Composability	N/S	✓ (Table 5-22)	N/S	N/S
Options Depreciators	Latency	✓ (Table 5-24)	✓ (Table 5-24)	-	-
	Dependency	N/S	✓ (Table 5-24)	-	-

As Table 5-18 shows, the SOA flexibility-centric characteristics had a two-way impact on the effectiveness of the process options. On the one hand, flexibility-centric characteristics had a positive impact on the effectiveness of the operational options and growth options. On the other hand, it increased the latency and dynamic dependency between systems, which reduced the effectiveness of the process options.

The structure-centric characteristics had a positive impact on the growth options while it similarly increased the dynamic dependencies between services.

Flexibility-centric characteristics impact operational options

To better explain the interactions between the SOA characteristics and the process options, the mechanisms involved in these interactions were identified and analysed from the data. Table 5-19 reports how SOA flexibility-centric characteristics positively impacted the effectiveness of the operational options.

Table 5-19 – Empirical evidence of how ‘Operational Options’ in impacted positively by the ‘Flexibility-centric’

SOA Characteristic (Theme / Characteristic)	**Process Option**	**Mechanism**	**Reported Cases**
Flexibility-centric\ Integration	Operational options (Switch input)	- Faster connection recovery process through a range of connectivity and adaptability options available (Option range) *(Airline and Airport Strong, Bank weak - opposing case)* - Increase the range of options particularly with connectivity to cloud services (Option range) *(Observed at the Airline)*	AL AL-P AP AP-P BK *(Bank Opposing case)*
Flexibility-centric\ Loosely coupled	Operational options (Switch input, Staged switch, Maintain)	- Migrate: reduce migration effort to change a service provider due to protection of other systems from change (Option richness) - Simpler trouble shooting due to decoupling (option richness) - Gradual migration through adaptability and abstraction (option richness) - Simulate the backend behaviour in the test environment (Option richness) *(Observed at the Bank – Bank in the box)*	AL AL-P AP AP-P BK BK-P
Flexibility-centric\ Standardisation	Operational options (Switch input, Maintain)	- Migrate: reduce migration effort to change a service provider due to consistency between the systems (option richness) - Reuse design: Reduced system maintenance effort due to consistent design (option richness) - Reuse process: Reduced system maintenance effort and improved quality due to consistent processes concerning the system support and management (option richness)	AL AL-P AP AP-P BK BK-P
Flexibility-centric\ Framework driven	Operational options (Maintain)	- Central management: Simpler trouble shooting through central management of concerns such as security, audit and logging (option richness)	AL AL-P AP AP-P BK BK-P

The data analysis from all the cases revealed that the flexibility-centric characteristics provided the IT operations teams with options for better management and maintenance of the SOA systems. The highest reported impact was the reduced effort in systems migrations and change, which is coded as ‘Switch’ and ‘Switch input’ according to the

Trigeorgis's taxonomy (1996) of operational options. The other visible process option was 'Maintain', which is the coding for options available for system maintenance and management processes to keep a system functional. The above process options were facilitated by the SOA loosely coupling, standardisation and SOA platform integration characteristics.

The loosely coupling characteristic protected the service consumers from getting impacted when the service provider had to change. The loosely coupling particularly helped when access to the external service provider or downstream system was centralised in a service, or abstracted by an SOA platform. One of the Airline IT managers explained the effect of loosely coupling, which was provided by their SOA platform (middleware), on the range of migration options: "*The other options are ways of migrating from one application to another one. Migration can be done more easily if you have SOA or even recently abstract integration layer can enable that. So I mentioned before we changed from one departure control system to another, that's a major thing for us. But a lot of that transition and we actually cut that over airport by airport. So depending on where you were leaving from or depend on which system was processing the passengers for that. So that was by handling that in some Middleware we were able to make that transparent to the other systems involved*". In the above case, the Airline took a staged migration ('staged switch') option to migrate from one provider to another. In this case, while loosely coupling opened up the opportunity to migrate the system gradually, it did not influence the number of options available. It instead, focused on the quality of the option, particularly the timeliness of option adoption. Loosely coupling by reducing the impact of change, improved the timing of implementing the change. Higher the systems were loosely coupled; the time would have reduced further. That is the reason Table 5-19 reports the impact of loosely coupled on the process options as '(Option richness)'.

Another option identified in the data analysis was the migration to the Cloud services. Through 'Loosely coupling', 'Standardisation' and 'Integration', SOA provides a range of options to source alternative service providers. The Airline CTO highlighted the above in his comment: "*And to some extent that discipline helps with some of the other forces that are affecting us through Cloud Services, where an application or the hosting with application could be anywhere. ... Having a service layer around things does open up options for us*".

The process and design standardisation has also improved the effectiveness of the SOA maintenance and management processes, the latter coded as the ‘Maintain’ process option. The consideration of ‘Maintain’ as a process option relies on the venture that the SOA-enabled processes will not be used and add value until the IT team decides to adopt the process and takes advantage of it. Consequently, the SOA-enabled process is an option for the IT team. The higher the ‘Maintain’ effectiveness is, the better it enables the IT team to manage changes and the operational challenges. For instance, one of the Airport IT managers referred to the standardisation of their document template and how it simplified the system management and maintenance through consistency for them: “*We’ve gone through the template and ... comes down to a section where you go what audits do we need to do on this information, ... and more structures exist …improved consistency in how logs are managed*”. The above finding was consistent with observations made at the Bank: “*That's implementation, design implementation testing deployment, very predictable because of SOA. Because once you have the pattern, we have done it before, we have got it - got them existing, the test - because we know it, it because test driven, it's easy*”. Similarly, there was also evidence of troubleshooting effort getting reduced due to loosely coupling and service-orientation: “*…if everything is decoupled, you can easily find your pinpoints*”.

‘Framework-driven’ was the last identified SOA characteristic that impacted the operational options. By providing a set of reusable services, the SOA platform framework-driven characteristic centralised the management and maintenance systems. Centralisation of services, such as system logging and auditing, improved the IT system maintenance and management considerably. For instance, the Bank benefited from their centralised logging service to diagnose issues: “*SOA saved us so many times, I can't even begin to say. So, for example, something is going wrong, oh my God, it's running wrong, the first thing they look is DGW logs. And DGW logs will tell you whether it's a front-end problem or whether it's a backend problem. And basically, it so many times, so many times. Now, because that's why, it's a central source of information*”. There was a similar comment from the Bank developer manager: “*there is benefits for sure of centralizing around a service-based approach I ... it does allow all the front ends to access back end systems through a common gateway so there is already benefits there, there is benefits around security, shared security”*.

The cross-case comparison between the cases showed that the Airline and the Airport relied on their SOA platform to provide loosely coupling and integration capabilities. In these cases, the platform provided a range of options with respect to the connectivity protocols and adaptabilities to support different message formats. The access to back office services and systems were also wrapped in the SOA platform to protect the service consumers from any change that might occur in the backend systems, such as the Airline departure control system or the Airport Flight scheduling and management system. The Bank, however as an opposing case, did not utilise any commercial SOA platform. They instead relied on their internally developed platform, DGW, which wrapped all the backend services, such as their core banking, to abstract the frontend from the backend services. While they implemented all the SOA characteristics in the DGW, achieving connectivity and adaptability required software development activities. The reduced SOA integration capability in the DGW impacted their ability to recover a connection and adapt to a different service provider.

To summarise, the SOA integration capability, standardisation, and loosely coupling improved the effectiveness of operational options such as ‘switch’, ‘switch input’, ‘staged switch’ and ‘maintain’ both from the range of options available to the IT, as well as the richness of the options. The data analysis revealed that the SOA integration mainly improved the range of options available to the IT team, whereas the other characteristics focused on improving the richness of the process options.

Flexibility-centric characteristics impact growth options

Above discussions covered the impact of SOA on the operational options. Focusing on the growth options, the analysis of data revealed that SOA flexibility-centric, in addition to operational options, improved the growth options. All the cases confirmed that integration through adaptability and range of connectivity options, loosely coupling, standardisation, and framework-driven reduced the effort involved in the service development activities. Table 5-20 summarises the above impacts and the mechanisms involved in such interactions.

Table 5-20 – Empirical evidence of how ‘Growth Options’ in impacted positively by the ‘Flexibility-centric’

SOA Characteristic (Theme / Characteristic)	**Process Option**	**Mechanism**	**Reported Cases**

Flexibility-centric\ Integration	Growth options (Extension by connectivity)	- Reduction of development effort by providing a range of connectivity and adaptability features (option range).	AL AL-P *(A380, Emirates Integ and Loyalty)* AP AP-P *(AOS, Internet, CUTE)*
Flexibility-centric\ Loosely coupled	Growth options (Extension of channels by reuse, Extension by protecting)	- Provides future options to support new channels in future (option richness) (e.g. *Digital and Loyalty support of different mobile OSs)* - Provided options to developers to replace a backend service with a new one (option richness)	AL AL-P *(Emirates Integ, Loyalty)* AP AP-P *(AOS, CUTE)* BK BK-P *(Digital, Bank in the box)*
Flexibility-centric\ Standardisation	Growth options (Extension by reuse)	- Reduction of development effort by providing consistent approaches to solve similar problems (option richness) (*Use of patterns in all cases, consistent deployment processes)*	AL AL-P *(Loyalty, A380, Emirates Integ)* AP AP-P *(AOS, CUTE, Internet)* BK BK-P *(Digital, DGW, BSL)*
Flexibility-centric\ Framework driven	Growth options (Extension by reuse)	- Reduction of development effort by providing a range of pre-built services in the platform (option range) – *(Bank DGW central logging service, Airline Digital Authentication service, Airport transformation service)*.	AL AL-P AP AP-P BK BK-P *(All projects)*

The data analysis indicated that the SOA integration characteristic with reliance on the SOA platform reduced the effort involved in the system development. The integration also provided options to extend systems in future if other types of connectivity protocols or message formats are required. In NVivo, the extension of systems through integration was coded as ‘Extension by connectivity’. The above option and the impact of SOA integration on the system extension option can be observed in the comment from the Airport architect: “*Because if you look at ESB is being a pattern of a new SOA and so, an ESB gives you lots of choices because you have protocol independent, you can start the process off with a new protocol you want, we can see data from one point into another, we can split data, we can do all sorts of things. I think in terms of choice points, it gives you a lot*”. The Airline reported similar impacts: “*And the other options that creates much more flexible ways of interacting with the systems so, ... different ways of interacting with the systems*”. In the cross-case comparison, however, the data analysis did not identify any similar observation at the Bank. The lack of observation can be because the Bank does not benefit from a platform that has similar integration capabilities.

While integration provided a range of options to reduce the development effort in integration processes or data between systems, loosely coupling provided developers with options to stage their development. Loosely coupling in the context of SOA allowed the development teams to support different types of channels in the future, including mobile, interactive voice response, and websites. The data analysis revealed the support of additional channels (‘extension of channels by reuse’) in the cases of the Bank and the Airline. Both organisations supported additional channels at different stages with reliance on the loosely coupling and generic services that were independent of the channel logics.

‘Standardisation’ is the other highly reported SOA flexibility-centric characteristic that affects the growth options. In the studied cases, standardisation made the processes repeatable, leading to consistency and effort reduction in how the services designed, developed, tested and deployed. The created option, in this study, was coded as ‘Extend by reuse’ due to the reuse of consistent processes. The Airline architect explained the above impact: “*And there are repeatable processes, which means in less effort in building, in deployment processes*”. Similarly, the Bank reported similar results: “*Implementation, design, testing deployment, very predictable because of SOA. Because once you have the patterns, we have done it before, we have got it - got them existing, the test - because we know it, it because test driven, it's easy*”. Overall, the cases reported that standardisation has reduced the effort in design, development, testing and deployment processes due to the established patterns and procedures, which can be applied readily.

Standardisation improved the richness of 'Extend by reuse'. As standardisation increased, the variability in how processes performed reduced. Consequently, standardisation focused on creating consistent and limited approaches to increase the quality, repeatability and therefore timeliness of their implementations.

Finally, the *framework driven* reduced the development effort by providing out of the box libraries and services, which can handle logging, auditing, and other predefine services in the SOA platform. Airport and Airline cases relied on their commercial SOA platform to provide such services, and the Bank developed these services in its DGW project. The Airline architect explained the benefits the platform has provided to them: "*SOA there is a very important concept that we just use infrastructure as a service. So, in a traditional, in a way, whenever there is a new business requirements, they have the built application to automate that, but with SOA in place is they can … leverage existing non-functional requirements which actually take a lot of time in traditional applications that are transaction security, auditing … yeah, logging part as well*".

Flexibility-centric characteristics impact options depreciators

In addition to the positive impacts reported above, the data analysis also revealed negative effects that SOA integration has had on the system performance. Table 5-21 summarises the mechanisms involved in the interactions between the connectivity and the performance degradation.

Table 5-21 – Empirical evidence of how 'Flexibility-centric' characteristics create 'Options depreciators'

SOA Characteristic (Theme / Characteristic)	**Process Option** (Theme / factor)	**Mechanism**	**Reported Cases**
Flexibility-centric\ Integration	Options depreciators \ Latency	- Increased time for communication between services reduced the system performance. *(Airport Internet – Latency in departure & arrival time, Airline A380)*	AL AL-P AP AP-P BK BK-P

The data analysis revealed that as services were distributed over a wider network, the communication between services took a longer period; this in turn, resulted in the performance degradation for the service consumers. The performance issue was particularly the case for services which returned a large amount of data, as reported by the Bank: "*it really goes down to that level performance, can sometimes be a concern if you have a heavy service that returns a lot of information and does a lot of pricing things and now the things like workflow do you want to have a multistep core with five different services or you want to have one service that maintains the logic inside*". There was a similar comment from the Airline: "*if there was a service that got that but then also went and got all that… But something that's sort of complex to produce and produces a huge amount of information that's taken a long time to aggregate and consume the next user to come along and want to use that ... doesn't want the performance overhead of having to wait for all that to happen. And the network overhead and the slowness of large messages and so on*".

The above comments indicate that the network and connectivity overhead have implications on the performance of the system, particularly when the size of information travelled on the network grows. The performance implications reduce the effectiveness of the operational options, such as migration to cloud services. It can reduce not only the range of options, e.g. by eliminating the options that introduce intolerable performance implications, but also the quality of the options.

Structure-centric characteristics impact growth options and options depreciators

The next group of SOA characteristics with an impact on the process options is the 'structure-centric' group of characteristics. The data analysis revealed that the 'structure-centric' characteristics had a positive impact on the growth options while it still increased the options depreciators.

Table 5-22 reports the mechanisms involved in the impact of 'structure-centric' characteristics on the effectiveness of the growth options.

Table 5-22 – Empirical evidence of how 'Growth Options' is impacted positively by the 'Structure-centric'

SOA Characteristic (Theme / Characteristic)	**Process Option**	**Mechanism**	**Reported Cases**

Structure-centric\ Granularity	Growth options (Extension by reuse, Extension in stages)	- Reduce software development effort by increased reuse of services due to their fine granularity (option range) *(All projects, opposing case Airline Blackberry)* - Flexibility to extend the system or do a staged delivery (option range) *(Airport AOS, Bank Saving Maximiser, Airline Loyalty)* - Improved quality of testing due to the granularity of the services (option richness) *(Airport and Bank)*	AL AL-P AP AP-P BK BK-P
Structure-centric\ Generalisability	Growth options (Extension by reuse)	- Reduce software development effort by increased reuse of services due to the functionality being applicable to more use cases (option richness) (*Bank DGW opposing case due to duplication)*	AL AL-P AP AP-P BK BK-P
Structure-centric\ Service Cohesion	Growth options (Extension by reuse)	- Reduce software development effort by increased reuse of services due to the functionality offered by the service being related to a specific function (option richness)	AL AL-P AP AP-P BK BK-P
Structure-centric\ Autonomy	Growth options (Extension by reuse)	- Reduce the coordination effort in service development processes, such as testing, change management due to reduced dependency (option richness).	AL AL-P AP AP-P BK BK-P
Structure-centric\ Loosely coupled	Growth options (Extension by reuse, Extension in stages)	- Reduce software development effort by increased reuse of service due to the loosely coupling (option richness) - Parallel development reduce the overall time to market (option richness) - Flexibility to extend the system or do a staged delivery (option range) - Gradual migration through adaptability and abstraction (option richness) - Migrate: Increase the range of options particularly with Cloud options (Option range) *(Airline)*	AL AL-P AP AP-P BK BK-P
Structure-centric\ Hierarchical Layering	Growth options (Extension by reuse)	- Reduce software development effort by increased reuse of core services (option richness) - Reduced the coordination effort in service development processes due to reduced dependency between services through layering (option richness)	AL-P BK-P *(Airline Loyalty and Bank Digital projects)*

		Reduced change to the core services and consequent impacts on the dependent services (option richness) *(Airline Loyalty and Bank Digital Project literal replication, other projects opposing cases – theoretical replication)*	
Structure-centric\ Composibility	Growth options (Extension by combining)	Reduced development effort to compose existing services and create new services due to functionality available in SOA platform (option range). *(Airline only. Airport and Bank opposing cases)*	AL AL-P

Similarly, Table 5-23 lists the options depreciators and the mechanisms involved in the creation of the options depreciators.

Table 5-23 – Empirical evidence of how 'Structure-centric' characteristics create 'Options depreciators'

SOA Characteristic (Theme / Characteristic)	**Process Option** (Theme / factor)	**Mechanism**	**Reported Cases**
Structure-centric\ Granularity	Options depreciators \ Latency	- Reduces the system performance due to additional network connectivity created by the finer granularity.	AL AL-P AP AP-P BK BK-P
Structure-centric\ Granularity	Options depreciators \ Dependency	- Additional dependency between services due to additional number of services available and their dependencies to complete the required task.	AL AL-P AP AP-P BK BK-P
Structure-centric\ Hierarchical Layering	Options depreciators \ Dependency	- Reduces the dependency between services due to the layered structure of services.	AL-P BK-P *(Bank Digital and Airline Loyalty)*

Structure-centric\ Autonomy	Options depreciators \ Dependency	- Reduces the dependency between services	AL AL-P AP AP-P BK BK-P

As shown in Table 5-22, the data analysis identified reuse as the key mechanism that the SOA structure-centric characteristics improved the effectiveness of the growth options. Reuse was dependent on a few SOA characteristics to increase the range of options and their fitness to meet the change requirements. As shown in Table 5-23, the SOA characteristics also impacted the options depreciators by increasing dependency and latency. Below expands how each SOA 'structure-centric' characteristic impacted the growth options and the options depreciators.

'Granularity' was one of the main characteristics that had an impact on the reuse. While there were reports of finer granular services achieving higher reuse, architecture with fined grained services attracted higher performance degradation and higher dependencies between services. In all cases, the interviewees agreed that the service granularity must be at a 'reasonable' level: "*...so the design approach of having services with reasonably sensible boundaries around them and reasonable granularity helped us with reuse. So let's take that crew positioning project, if you had a service that was quite small ... you had to call it again to iterate through each crew member, that's too finely in grain it's not going to be efficient from a consumer point of view. On the other hand if there was a service that got that and all those peoples work schedules for the next month as well to see whether we are running to any fatigue issues, ... something that's sort of complex to produce and produces a huge amount of information that's taken a long time to aggregate ... doesn't want the performance overhead of having to wait for all that to happen*". The Airport ESB developer explained this further: "*We focus a lot of time making sure each service doesn't do too much and doesn't do just a little bit and that's the key actually of, for this sort of architecture, and basically each service had to do something specific but isolate it to other services*".

The above statements suggest that the service granularity, while having positive impact on the range of options available for reuse, it can also introduce additional network overhead and dependencies between finer-grained services. Accordingly, the above statements suggested a ‘reasonable’ level, which is a balance between the granularity and the options depreciators (latency and dependency).

The ‘generalisability’ and ‘service cohesion’ were the other two characteristics that had significant implications on the level of service reuse. Services that are built to address multiple use cases rather than a particular context to provide a better option fitness to the change requirements. Accordingly, more generalizable services improve the options richness. The Bank reports on such impact: “*If you look at the most used services in the DGW…, there is 20 or 50 that are very heavily used and then it starts dropping off. They are the ones that are the most generic as supposed to getting customer information getting account details move money they have slightly different names for that but that’s the functions they are performing”*. While the services need to be generic, each service needs to do a discrete and related functionality to promote its reuse: “S*o, things do discreet activities. So, I have got in mapping that, it's a message, it's very discreet activities, it sounds very easy to understand what goes on in five year time”.* Both of the above characteristics increase the option richness by providing a better relevance and fitness to the change requirements.

The next SOA characteristic is ‘service autonomy’. Autonomy in the service design and service deployment has reduced the dependency between services, hence reduced the coordination overheads involved in testing, deployment, and change management of services. The above was particularly visible at the Airline, where they packaged services based on their dependencies to reduce the impact of code change and deployment on other services: *“…issues within one particular application impacting the entire SOA stack, so the key learnings ... we understood the dependencies and we set it up in a way we have actually done a logical segregation of all those services and also we have deployed them accordingly so services don't impact the others”.* Autonomy of services improved the option richness by reducing the coordination overhead (option timeliness).

The role of ‘loosely coupling’ in improving the options effectiveness has already been discussed and elaborated. In the studied cases, ‘loosely coupling’ is not only essential for the SOA to realise the services concept but it also provides options such as parallel development and staged system delivery. For instance, the Bank had two teams engaged in parallel, one to develop the BSL services, and the other team to develop the DGW services.

The hierarchical layering and its impact on the option effectiveness was an interesting finding identified in the cross-case comparison between the Bank digital team and the rest of the Bank divisions. It was also visible at the Airline with the Airline loyalty team.

The data analysis revealed that the 'hierarchical layering' improves the effectiveness of the growth options by improving the timeliness of the options, their relevance and fitness to the requirements.

The layering of services based on their level of 'generalisability', 'service cohesion', and 'autonomy' improved the reuse of services and reduce changes to the services in the lower layers (core services). The improved reuse can be observed in the below comment: "*…what we found was actually the right way of doing things was to have a separate layer up on top and then the core service layer. So one service operation it does one thing …what we found was that combination is the best - has the best characteristics or best possibility of reuse.".*

The data analysis revealed that services at the lower layers are closer to the core systems and core business data, whereas services on the higher layers were closer to the presentation layers and user channels such as mobile. In both the Airline and the Bank, layering was aligned with the level of change to each layer, with the lower layers going through less change and the higher layers experiencing more changes, as explained below: "*…the layers that changed quickly and the layers that don't change as much so, classic examples, …move money really, really, really doesn't change all that. Okay. And however, the way to access move money, does. So, five years ago, smartphones didn't even exist. But now, we are talking Google Glass and watch and all that…So business layer, everyone changes, reuse becomes less. And the reuse of the DGW is much higher but the level of change is less".*

By making core services more generic, the *hierarchical layering* leads to a higher reuse of the core services, while the flexibility to create channel or process specific services remains for the developers in the other layers: "*it gives you a good balance between reusability and optimization. So, because there is always this natural tension if you optimize for particular channel and you can't really reuse it. So, what we want to do in this case is basically optimize for channel. But have a still encourage high reuse, at the same time.*".

By creating the 'hierarchical layering' and positioning more generalised services at the lower layers, the level of service reusability without any change increase. Such requirement fitness (option relevance) and timely availability (option timely) increase the effectiveness of the growth options.

Additionally, the cases showed that as a result of layering, the development and change coordination overhead reduced. The Bank development manager compared this additional flexibility and reduction of the effort to another bank he used to work for in the past, which did not have such layering: "*I think the ability to change, …It's just kind of the opposite what I've seen with <Another bank's name>, which was hard to change. Now digital got the business service layer, which is combining multiple DGW calls and doing more from the business perspective. I think this is giving us more flexibility because you build your service in here, and you put a bit of logic in here then people will be able to change that based on the business context…*". The data analysis revealed that centralisation of dependencies to a number of core services rather than a network of services reduces the 'options depreciators', hence reduced the coordination effort involved in the development and deployment of services.

The last SOA 'structure-centric' characteristic that was identified during the data analysis was 'Composability'. The Airline by utilising the features available in their SOA platform expedited the development and orchestration of new coarse-grained services, which reduced the development effort for their team. The Airline delivery and support manager explains how their Oracle ESB platform reduced their development effort: "*so why we could quickly build up services is purely because it's mostly just drag and drop and then do some minor configurations*". Similarly, the Airline IT manager reflected on his experience in one of his projects, which relied on standard programming language than their SOA platform: "*Exactly so when I say about this charter stuff also, it was developed in J2EE so it took…I think there is a comparable difference when you do it using any programming language from the scratch rather than using SOA so I think that make all the difference of bringing quicker you know like development.*"

In comparison, the above benefit was not reported at the Bank due to the development team using programming languages (JAVA and .NET) to orchestrate and develop new services. This would suggest that 'Composability' increases the range of development options. If the team has enough experience with the tool, they can leverage off the option and deliver the system faster.

Options depreciators impact process options

The above discussions have already covered the impact of SOA ‘flexibility-centric’ and ‘structure-centric’ characteristics on the operational options, growth options and options depreciators, as outlined in Table 5-18. The last relationship for discussions is the impact of options depreciators on the process options. As highlighted above, and despite their positive impacts on the effectiveness of the process options, a few SOA characteristics increased the connectivity latency and dependency between services. The data analysis across all cases revealed that the increased latency had implications on the system performance. Similarly, increased dependency between services created additional overheads in coordination activities such change management and deployments. These additional complexities, in fact, reduced the effectiveness of the process options. As shown in Table 5-24, the latency impacted the fitness and relevance of the option to the change requirements (system performance requirements) and the dependency impacted the timeliness of the option and its on-time availability for reuse.

Table 5-24 – Empirical evidence of how ‘option-depreciators’ affecting process options

SOA Characteristic (Theme / Characteristic)	**Process Option**	**Mechanism**	**Reported Cases**
Options depreciators\ Latency	Operational Options Growth Options	- Reduced system performance due to additional delays on the service calls. The performance degradation scaled up as per the number of service dependencies (option richness)	AL AL-P AP AP-P BK BK-P
Options depreciators\ Dependency	Growth Options	- Increased impact assessment and regression testing depending on the level of dependency (option richness) - Increased the change coordination (option richness)	AL AL-P AP AP-P BK BK-P

The data analysis revealed that additional dependency affected a number of processes. For instance, the dependency between services increased the coordination effort for system outage at the Airline: “*if you are trying to make a change and there are many dependencies, you need to get the confirmation from multiple stakeholders for the change before you can make the change*”. Over time, the Airline reduced this impact by versioning the services and maintaining different versions, in addition to creation a highly available infrastructure, to cater for service upgrades: "*it very much mitigated by versioning policy in the first place, so if something identified, there is even the slighter risks that they are going to impact our major processes. Normally what we do is we create a new version of services instead and then we take time to get the existing consumer migrated to the new version*".

The Bank also reported similar challenge in managing the list of stakeholders and coordinating the change with the stakeholders. The Bank also implemented the policy of versioning and not deprecating any of the services: "*we have a very strong belief in never deprecating anything. So, backward compatible for us is everything*".

The other process that was impacted by the dependency was the testing process. All cases had to perform an impact assessment to assess the size of impacts and the required testings, as explained by the Bank: "*So, if you change something, you need to analyse what services are impacted and then your test strategy comes from, actually testing the thing impacted. That would be then your base for regression testing*". The Airline and the Airport also reported similar findings: "*make it a bit challenging in terms of testing but with sufficient planning, it is quite easier*".

To summarise the above discussions, options depreciators negatively impact the change coordination and testing processes. Such negative impacts are translated to the reduction of effectiveness in the growth and operational options. The effectiveness, and consequently, the adoption of an option, depends on the option's overall availability, its quality and its ability to meet the change requirements with minimal modification.

5.5.2 Process options improve IT responding capability

The previous subsection explained how the SOA characteristics impacted the effectiveness of the process options. This subsection provides evidence for the effects of process options on the IT responding capability.

The data analysis revealed that, by offering a range of choices, the process options increased the speed of the IT team to address the change requirements when the offered options were effective in meeting the requirements. Table 5-25 summarises the mechanisms involved in the interactions between the process options and the IT responding capability.

Table 5-25 – Empirical evidence of how 'operational options' affect IT responding capability

Process options	Responding capability	Mechanism	Reported Cases
Operational options	Responding capability	- Availability of capabilities, which can readily be applied, leading to reduced effort for the IT operational teams. - Protection of existing systems from the impact of change - Flexibility to stage the operational changes	AL AL-P AP AP-P BK BK-P
Growth options	Responding capability	- Availability of capabilities, which can readily be deployed, leading to reduced effort for the IT development teams. - Ability to reconfigure, combine and orchestrate assets. - Flexibility to stage the system development through time - Flexibility to explore and pilot concepts - Ability to leverage off a wider range of options available through external network (e.g. Cloud) - Flexibility to replace or add new capabilities to the existing assets. - Tolerance to change in requirements.	AL AL-P AP AP-P BK BK-P

The comparison between two projects at the Airline highlights that the availability and fitness of a suitable option, such as a service reuse, has direct effect on faster system delivery: "*So we were able to improve the responsiveness for that second project, so we absolutely did that better than we would have been if we hadn't taken a SOA approach to the first project*". Similarly, the Airline architect explains the use of existing options and their orchestration in building a new capability promptly: "*So, traditionally, whenever there was a new business requirements, they had to build an application to automate that. But with SOA in place, they can actually orchestrate different existing services to build a new capability… When we talk about IT responsiveness to the business definitely improved because of the SOA, because developing the new business capability takes short time*".

The other finding was related to the tolerance to changes in the requirements. The data analysis revealed that due to its modularity, SOA tolerated changes in the system requirements. Tolerance to change was especially important in projects that changes to the system were disruptive, and business had to explore and find the best solution to embrace the opportunity. The Airline delivery and support manager expands on this: "*Main thing which I'm seeing and we go within an SOA project is the agility, so it responds quicker to changes in the project, the requirement changes. It responds pretty quickly to whatever the needs of the business, whereas when we go to non SOA the agility is less. We'll not be able to quickly make those changes and fit them into the schedule*". Similar report was observed from the Bank and the Airport: "*I have seen the design stage and the development and all the stuff, it's shorter... I think this is what we're starting to get into how SOA can lead towards being more agile".*

The other finding was about the additional flexibility to explore and identify the best solution: "*it gives them (business) a wider variety of options in terms of what they need, for example do I need to get a product or do I already have services which do part of the work, so it gives them flexibility*".

The Airline CTO finally explains the importance of reconfiguration and reuse of existing assets to build new systems quickly: "*We see our integration platform is absolutely a key capability and underpinning our agile aspirations and vision across the IT department and therefore enabling the business... and the integration team realize that through reuse of existing assets and the development of existing assets*".

The above discussions indicate that growth options improved the IT responding capability by enabling the IT team to utilise, combine, reconfigure, and orchestrate the existing services to build or extend systems quickly. Services can be sourced internally or from an external network such as Cloud service providers. The build or extension of the system can be staged and delivered as the market needs evolve.

As per the data, the IT responding capability improved by:

- Availability of a range of options, which can be used as required
- Alignment of each option to the change requirements (richness of the option)
- The ability to put the options together and combine them as required

While the range of options provided flexibility to choose the best choice suitable at the time, the richness of the option facilitated the adoption of the option in a timely manner with the required quality.

On the operational challenges as shown in the previous section, the IT responding capability relied on the flexibilities that the process options provided to the IT operational teams to deal with operational challenges such as change of a service provider, a system migration and system maintenance. The operational options through their range of options and their richness improved the IT responding capability to address the change requirements promptly.

5.5.3 Summary of impact of SOA on IT Responding Capability

This subsection summarises the findings concerning how SOA characteristics impact the IT responding capability. This section reviewed the impact of SOA flexibility-centric and structure-centric characteristics on the process options and then, on the IT responding capability. During the data analysis, the initial conceptualised relationships between the SOA characteristics, process options and IT responding capability were assessed and refined. The initial propositions assumed:

> **Proposition (R2):** *The Operational options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (R3):** *The IT growth options are positively affected by the SOA characteristics already embedded in the information systems.*
>
> **Proposition (P2):** *The IT responding capability is positively affected by the process-based options that SOA offers.*

As explained in previous sections, while the collected data supported the above propositions to a large extent, there is also room for their refinement.

As the data analysis revealed, the operational options were impacted by the SOA flexibility-centric characteristics. Integration, loosely coupling, standardisation, and framework-driven improved the effectiveness of operational options such as switch, stage, and maintain options. While integration focused on providing a range of options suitable for switching connectivity between systems (e.g., connectivity and adaptability options), loosely coupling instead concentrated on the richness of the options. By reducing the impact of change on downstream systems, 'loosely coupling' reduced the time required to adopt the options such as system migration (switch). Standardisation was similar to loosely coupling in that by creating repeatable patterns, standardisation improved the richness of options such as migration (switch) or system maintenance processes (maintain). As shown in the previous sections, the improved option richness led to faster and higher quality delivery of the change requirements. Finally, the framework-driven improved the quality of system maintenance by providing options to centrally manage the systems. For the above reasons, the first proposition is refined as below:

> **Refined Proposition (RP1-R2):** *The effectiveness of operational options is positively affected by the SOA flexibility-centric characteristics already embedded in the information systems.*

On the growth options, the data analysis showed that the process options involved in the development of Service-oriented systems were affected by flexibility-centric and structure-centric characteristics.

The integration and framework-driven characteristics provided a range of options for reuse, which in turn, reduced the development effort involved in building or extending a system. Furthermore, the standardisation and loosely coupling focused on improving the timeliness and quality of the processes.

The structure-centric characteristics, on the other hand, improved the growth options, mainly by providing reuse of the existing services and ability to recombine them. SOA structure-centric characteristics improved the range and richness of the process options such as ‘Extension by reuse’, ‘Extension in stages’ and ‘Extension by combining’.

On this basis, the second proposition can be refined as below:

> **Refined Proposition (RP1-R3):** *The effectiveness of IT growth options is positively affected by the SOA flexibility-centric and structure-centric characteristics already embedded in the information systems.*

While the flexibility-centric and structure-centric increased the effectiveness of the process options, there were also consequences. As discussed in the previous sections, the rise of network connectivity increased the network latency which resulted in the system performance degradation. Similarly, higher granularity led to higher dependencies between services and further interactions on the network. The data analysis showed that the extra dependencies between services create additional coordination overhead for processes such as change management and testing. Accordingly, a new proposition is suggested below:

> **New Proposition (RP1):** *The effectiveness of the IT process options is negatively affected by the options depreciators that SOA flexibility-centric and structure-centric characteristics create.*

Finally, the data analysis showed that the operational options and growth options improve the IT responding capability by providing options, which could readily be deployed, reconfigured and combined to meet the change requirements. By improving the effectiveness of the options in the form of range or richness, SOA characteristics increased the fitness of the option to the change requirements and consequently their prompt response to the change. On this basis, the last proposition is refined as below:

> **Refined Proposition (RP1-P2):** *The IT responding capability is positively impacted by the effective process-based options that SOA offers.*

Figure 5-5 presents these propositions in the form of a conceptual model.

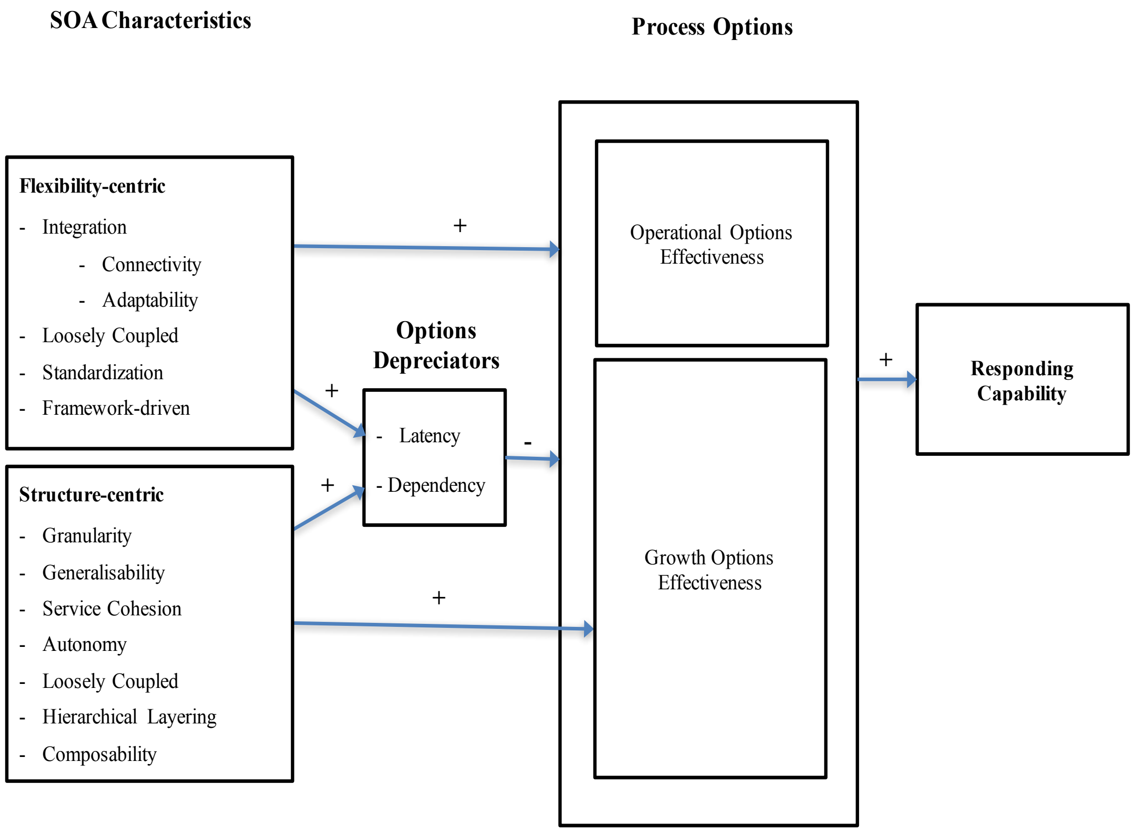


Figure 5-5 - SOA Characteristics impact on IT Responding Capability Model

It is important to note that the above conceptual model does not consider the effects of non-SOA factors on the relationships between the SOA and the IT responding capability. Such relationships will be discussed in the next section.

5.6 Complementary resources affecting the impact of SOA on Agility

Beyond the SOA, the data analysis revealed a number of complementary factors which affect the impact of SOA on IT agility. This section reviews the three identified complementary factors, which are 'Continuous IT and business engagement', 'Shared insight' and 'adaptive governance'.

The cross-case data analysis revealed that 'Continuous IT and Business Engagement' influences the extent the SOA Information Repository characteristics creates insight for business and IT. At the Airline and the Bank, the business units that had continuous engagement with the IT benefited most from the service definitions and other SOA information repository elements by gaining shared insight into the IT existing capabilities and SOA. By contrasting the studied cases, Section 5.6.1 proposed that the impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the continuous business and IT engagement.

Section 5.6.2 reviews the impact of shared insight on the effectiveness of the process options, in addition to the IT sensing capability. The gained shared insight improved the range and richness of the process options available to the IT to respond to the business needs. The above section concludes by proposing that the SOA-enabled 'Shared Insight' positively affects the effectiveness of the process options.

Finally, the data analysis revealed that the process options, even when effective, may not be adopted and executed due to factors such as lack of knowledge and lack of commitment. The cross-case analysis, however, revealed that an adaptive governance which is aligned to the hierarchical layering improved the impact of process options on the IT responding capability and addressed the above issue. Section 5.6.3 provides the evidence concerning the adaptive governance from the data and concludes that the adaptive governance moderates the impact of SOA-enabled process options on the IT responding capabilities.

5.6.1 Continuous IT and business engagement

The cross-case analysis indicated that the continuous business and IT engagement is essential for the creation of capability visibility and knowledge sharing between the business and the IT teams.

The role of continuous business and IT engagement became apparent in the comparisons between the Bank digital team and other business units within the Bank. Same thing happened in the case of Airline when comparing the loyalty team with other teams in the

Airline. As reported in section 5.4.1, the impact of SOA on ‘capability visibility’ and ‘knowledge sharing’ with business was only observed with the Bank digital team and the Airline loyalty team. The cross-case comparison highlighted that the digital team and the loyalty team had a very close and ongoing working relationship with the IT team. The above relationship becomes visible in the comment made by the Bank architects: “*The digital team is the one business unit which is different from everyone else. Because they have got largely a delivery team next door that they work hand in hand. …they are getting much more insight into architecture…. So, they understand the capabilities that exists, that helps them with their scoping and knowing what is possible from the knowledge that they have got about SOA and services that are available”.* While the information-centric characteristics were available to all of the Bank business units, the Digital team was the only unit that consistently improved their knowledge of SOA services (Business IT knowledge) and understood the available capabilities and the capability gaps to deliver their requirements (Capability visibility).

The main difference between the digital team and the other teams was the reliance on the agile software development process which provided them with a high level of engagement with the IT delivery team.

The above observation was further confirmed in the interview with one of the Bank’s business managers. He referred to one of his past projects, loyalty bonus initiative, in which the project team and business worked very closely and collaboratively to deliver the project in a very short period of time (eight weeks): *“Certainly from a business perspective that was great, how agile, how responsive IT was, awesome… I think what worked very well, in this loyalty bonus initiative was, …We literally went into the board room and got representation from across IT and across the business… just workshopped, …to flesh out the detail to articulate the document”.*

In this particular project, Agile software development was not used. Instead, the close collaboration between business and IT provided them with the opportunity to discuss the existing capabilities, such as services, the capability gaps and the timeframe required to address them. The discussion on gaps led to negotiation on requirements and options on how to address them: “*…we wanted to do these and then IT person go, yeah, okay but if you did it this way, it will be shorter, if you did your original way, it will take two weeks. If you did it this way, it will save you a week. I think great, good information to have, product will make call - let's compromise it”.* He then compared this project to other projects: "*...whereas if we did it the traditional way, we are working with the BA, we spent ages documenting it. We wouldn't know that what we require would be big or small, until a month later, two months later, when BA would go through with the developers and developers would go, actually this is too hard or this is too big"*. In the two above cases, the only common element was the close engagement between the business and the IT teams.

Similar findings were observed at the Airline with the loyalty team: "*Typically they (business) will come through the IT function to say okay, we need to this. And they know all the concept of web services they all know about it. Do we have a web service that can do this? Do we need to change anything? If we want to do this function, can it be in just two days ago? ...I've heard there's a service to do it. ... there's many business people up here who've been around for a while and who have done various projects. I know that there are services that should be accessed quickly and easily*".

The above evidence shows that the business and IT collaboration affects the impact of SOA characteristics on the shared insight options. As shown in Table 5-26, the 'continuous business and IT engagement' is defined as frequent interactions between the IT planning and delivery teams and the business users who sponsor and define the product functionality to shared information concerning the SOA capabilities.

Table 5-26 – Continuous IT and Business Engagement Definition

Category (Parent node)	Description	Reported Cases
Continuous IT and Business Engagement	Frequent interactions between the IT planning and delivery teams and the business users who sponsor and define the product functionalities to shared information concerning the current and future SOA capabilities.	AL-P BK-P

The data analysis showed that the continuous business and IT engagement has a moderating effect on how SOA information-centric characteristics affect the effectiveness of the shared insight options. The moderating effect of the continuous business and IT engagement is based on its contracting effect on the SOA information-centric characteristics. As highlighted above, while the IT can improve its SOA information repository characteristics by measures such as documentation and publication of service definitions, they will not be as effective in creating shared insight, until the IT and Business teams engage and discuss the available capabilities, capability gaps and the potential options. On this basis, the following proposition is suggested:

> **New Proposition (RP4-R1):** *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the continuous business and IT engagement.*

5.6.2 Relationship between Shared Insight and Process Options

Another relationship identified in the cross-case analysis was the relationship between the 'Shared Insight' and 'Process option' effectiveness. As summarised in Table 5-27, the data analysis revealed that high shared insight improved the range of process options and their richness.

Table 5-27 - Empirical evidence of how 'Shared Insight' positively impact the 'process' options

Shared insight	Process Option	Mechanism	Reported Cases
Shared insight	Process options	- Additional options to engage new vendors, service providers or new developers (option range) - Options to have smaller and more frequent deliveries (option range) - Align the new requirements to the existing IT capabilities (option richness) - Reduce the delivery cycle by refined scoping processing (option range)	AL-P BK-P *(Digital and Loyalty)*

The above relationship became visible when the Bank Digital team that gained high insight, was compared with other teams within the organisation. As a result of insight on the available IT capabilities and their gaps, the Digital team had increased the range of the options available to them. For instance, the Digital team divided their requirements into smaller pieces for faster delivery and faster time to market. The additional insight provided them with options to break their requirements to smaller deliverables and get the delivery team to deliver them accordingly. The Bank architect explains the process option they gained through their higher insight by saying: "*So, with Digital once they understood the impact, they have actually started asking for their own delivery capability. And basically, they started to chop their work down to bits, to smaller bits, that they can send down to this team. And this team just operates out them. They, you can actually see that, the digital side actually coming up, pushing more change through, pushing more* *work through, pushing more improvements through, whereas, the other ones all stuck in project management and trying to get the stuff out but really can't do much else of them just doing stuff".*

Similarly, the insight they gained through SOA provided them with additional options such as engagement of alternative suppliers to expedite the delivery: "*SOA hasn't traditionally been something that the businesses would understand it, ... we are slowly introducing those concepts into the business, and we are finding that once they understand, they are actually not wanting to let go. So, classic example would be Digital. ...because seeing the services are there, I [Digital team] can go and talk to my vendor, this is exposed, vendor, please call this now, IT is not part involve because, once IT is provided the service, once we have published the service, then basically it's up to the business to decide what they want to do with the service. And that has strongly allowed them to actually move a lot quicker, open up options for them*". As highlighted by the Bank developer manager, the knowledge sharing capability provides an opportunity to swap or increase the resources engaged in the development: "*Due to consistent framework in the middleware, new developers can come on-board very quickly, contractors can come in and be effective fairly quickly as well and because that knowledge is shared around the team there is a good support network within the team".*

Review of the Digital team processes also indicated that they have adopted a refined scoping exercise. While the rest of the business units still spent considerable amount of time in capturing the project scope and requirements before assessing their viability, the Digital team, due to their knowledge of available services and capability visibility, shortened the above process and focused on viable requirements. The Bank architect reflected on the above Digital team insight: "*You can quickly determine, whether you got the service or not, so ... basically with that, see I don't even need to tell them [Digital team], they just go, we got a service there, I can engage someone else [externally to do development], done. I don't have the service, I need my, get IT or I will have my own delivery team".* Similarly, the business analysts (BAs) that are aware of services can reduce the delivery cycle: "*BAs who don't know anything about services, they will just go, you are not going to get a clear answer from it. But the people who know services, they can - in fact they just go here, here is service contract, this is what we can offer today. And I can map more request to this, and it actually also makes integration much easier.*"

In addition to the range of options, the shared insight also improved the richness of the options by improving the relevance of the options and their timeliness. The data analysis revealed that when the business became aware of the existing IT capabilities, they aligned the new requirements to the existing capabilities as much as possible to increase the reuse and consequently the time to market. The following comment provides the evidence on such effect: "*the thing is that by (business) knowing the problem domains better, by knowing what IT can and can't do, in their heads, when they form the solution, it's actually a solution that is optimal for the time and the cost of that they are aiming at…. whereas the other team, they do not necessarily align, as a result they make assumptions, they go and capture all the requirements and then there is this big misalignment between what they are trying to achieve and what at the end IT capabilities that exist.".*

The data analysis also revealed an opposing case that makes the proposed relationship stronger. One of the Airline IT manager explained how the options created in one of the projects were not used in a future project (the iPad application project) due to the knowledge being lost. In this particular case, the outsource of internal development and support capabilities after the first project and lack of governance led to the loss of knowledge on what existed and consequently the services not being used: "*yeah the lack of a mechanism to communicate effectively what was there …. So at that time we did not have a SOA centre of excellence or anything like that, we also at the time we transitioned from an internal developers and support people to outsourced…So that could be part of the why we lost that continuity we missed out some of the value years later… but I think we did that through the loss of the knowledge thus a lot of the stuff already existed"*. He then considers having enough people with knowledge of the services can resolve the issue: "*there is enough critical mass of people that do understand that this thing exists and what it does, enough to get people pointing in the right direction."*

The above evidence indicates that the shared insight has a direct impact on the effectiveness of options. With additional shared insight, the business and the IT teams can explore new options that expedite the delivery of the options. They can also refine the requirements to make them more aligned to the existing IT capabilities and consequently improve their richness. The data also revealed that lack of knowledge and shared insight impacts the effectiveness of the options by diminishing the effectiveness of the options. On this basis, the researcher suggests the following proposition:

New Proposition (RP4-R4): *The SOA-enabled 'Shared Insight' positively affects* *the effectiveness of the process options.*

5.6.3 Adaptive governance

The previous sections explained how SOA characteristics create process options and improve their effectiveness. The data analysis also highlighted cases which effective process options were not adopted and used, while existed. Different factors such as lack of knowledge about the options, and lack of developer's commitment to reuse were the reasons for the options not being adopted. The Airline IT manager highlighted the lack of governance and its impact on the iPad project: "*yeah the lack of a mechanism to communicate effectively what was there …. So at that time we did not have a SOA centre of excellence or anything like that…. now that we have an SOA and integration centre of excellence, we will avoid that sort of problem in the future".* He also indicated that after the establishment of the SOA centre of excellence team, which was the central governance for SOA at the Airline, the option waste will be prevented and the time to market will be improved. The reduction of duplication and waste of options is also reflected in the comments from the SOA architect: "*And most importantly we can manage there's no duplicate work across projects. For example, we see two projects requires restful services or Jason so instead actually we just ...called people from two projects together and we said we can develop a centralised Jason adapter for both of you guys so every project saves money*".

The Airline SOA centre of excellence presented centralisation of IT specification and IT implementation decisions. The Airline SOA governance manager explains the role of her team: "*We set up standards for the suppliers or for* the *Airline integration as a whole, then looking at the reference architecture, looking at SOA governance in the SOA space, and also hopefully management. So looking at what projects, how do we prioritise our service delivery in terms SOA services, budgeting criteria*". In addition to setting up the standards, the SOA centre of excellence team, in conjunction with the Airline Architecture Review Board reviewed and assessed the solution design options and decisions against the standards. They also control the delivery priorities of the projects and the IT sourcing decisions such as the delivery vendor. Accordingly, they outsourced the development and management of their SOA infrastructure to one vendor. The Airline SOA architect explains how this works: "*Because SOA platform is essentially managed by SOA centre of excellence … we maintain a list of projects that requires our advices* *and also we maintain the timeline for each project the contact person with the project, we make sure that we don't delay project. … been able to manage the quality for sure of the work*".

Similarly, the Airport IT manager reflected on how they had to enforce the control and ask the business to use the integration options available rather than make a point to point integration: "*we actually sort of had to pull the business back and say hey hang on, don't just do this point to point here*". Similarly, the Airport architect reflects the impact of not having governance on the service development: "*without the governance, then people just redevelop the same service over and over again and then, …it will cost more money*".

While the centralised governance implemented by the Airline centre of excellence increased the uptake of the options, there were also reports of additional overheads in projects. The Airline IT manager who worked with the Loyalty team considered the contractual arrangements and relationships with the development vendor as the main reasons for the additional overhead: "*Because of the relationship and the contract, I talk about the management layers, the TCS and the time it takes, it's much worse …*". She explained that the contract with the vendor was aligned with the objectives set by the Airline Group, whereas it was not aligned with the loyalty agility requirements: "*TCS is managed by the group and of course the contract was I don't know how many pages, it has cover everything to do with* the *Airline IT, whereas, these are the benefits that Loyalty is trying to establish by being a bit more agile and independent. They (Loyalty) are establishing relationships with other vendors who would do what we want, and we don't have to go through this …*".

Consequently, the Airline loyalty team implemented and controlled an additional layer of SOA platform on top of the corporate SOA platform. The additional layer combined with their control and governance on this layer improved their time to market and agility. The Airline IT manager explained this impact: "*From this additional service layer. So we as I said we've got TCS here and we've got another layer here. So we manage this layer, and we can actually change it quickly, because we have the relationship with the vendor. We can very quickly enable new channels to use the functions that we have here because we don't need to involve TCS…. Just flexibility and the level of change that they want*". The Airline Loyalty team still relied on the options available on the corporate SOA platform for all corporate services, while they controlled and developed the additional services such as logging and security token required for the Loyalty website and mobile application on their layer.

The above discussions demonstrate the importance of the governance in gaining agility through options execution. However, as shown above, the centralisation of governance can inhibit the agility by increasing the coordination overheads and reducing flexibility. An adaptive governance structure can reduce such overheads. In such structure as explained in Table 5-28, the options created by lower SOA layers are governed centrally, whereas the governance of higher layers will be decentralised to other teams and business units. Considering the fact that lower layers contain core and highly reusable services, the centralised governance improves the evolution of core services and ensure the backend services are protected. Similarly, the decentralised governance for the higher layers provides an improved decision making to expedite the delivery. Such decisions can include IT sourcing decisions such as engagement of multiple vendors for development or applying changes to the services. Considering the higher layers are closer to the consumer channels such as mobile and web, the services in the higher layers have a higher rate of change and less reusable due to their channel logics. Therefore, evolution of the services can benefit from the additional flexibility and decentralisation in the decision making for the higher layer services.

Table 5-28 – Adaptive Governance Definition

Category (Parent node)	Description	Reported Cases
Adaptive Governance	The degree the SOA decision making is aligned with the SOA hierarchical layering, in which the decision-making for the SOA lower layers is centralised in the IT team and the decision-making for the SOA higher layers are decentralised to relevant business line functions.	AL-P BK-P

The cross-case comparison between the Airline and the Bank confirms the above finding. At the Bank, the architecture team was responsible for the governance of the SOA and the DGW in the organisation, supported by an internal development team. While the architecture team established the standards and guidelines, they did not consistently validate the creation and execution of the services. The lack of validation led to a higher level of service duplication in the DGW, as explained by the Bank development manager: "*No not so much, certainly they didn't use to. I think they (architecture team governance) are more and when the new business service layer they did more yes but in the past so with the DGW services it (service definition and use) was really just with the hands of the* *development team*". The lack of strong governance resulted in the creation of service duplication: "*there is hundreds of services more than 500 services some of which are never used because they just have fallen out of use or maybe they were never ever used I don't know we actually don't need that".*

The Bank, unlike the Airline, applied a lighter level of governance on the service creation and execution, which resulted in a higher service duplication, and consequently, lower option execution.

On the service structure, the Bank, similar to the Airline, created a layered service structure governed by the Digital team rather than the central IT team. Such governance structure provided the Digital team with autonomy to consider their suitable options and decide how they must meet their required time to market. The Bank architect explained how the Digital team got involved in the BSL: "*When they (Digital team) develop the mobile, ... we said, look, we seriously recommend that you put it in business service layer which is publicly available. And basically we educate them in that particular process. And once the mobile app was in, the first advantage they could see was, the Windows phone app (could reuse the same service) and then a second one was ... update all our interest rates and it's actually fairly error-prone and manual as well. So, …they actually use the same APIs that the mobile app use and without any IT involvement. …they saw the power and then basically now that's only more to the point where they feel they own it - which is great."*

The split of governance mainly applied to the core and highly reusable services. The core services, which are in the lower layers, are managed and governed by the Bank architecture team, whereas the services on the higher layers are governed and controlled by the Digital team. The Bank architecture and strategy manager explained how they apply the governance: "*We believe the problem is governance of services, so what we've done is we've said what are the core entities that make up this bank. They have a customer, they have a product, they have accounts, …. that is all we govern, …. so as long as fundamentally at some point, they go to those core services, and those core services are what we govern. Now, we've done that in our designs today and we are implementing that as we speak".*

Since the lower layers covered the core services and abstracted the service consumers from the backend systems, a certain level of control was built into the DGW. Similarly, implementing auditing and security as mandatory services in the DGW enforced some of these control points, which reduced the overhead associated with the governance: "*And we are trying to design the system in such a way, we don't have layers and layers of bureaucracy and layers of people reviewing it. So, the only way we can do this is to build it into the system, so that you can't do it any other way. And that's what we are trying to do. ... that's the reason why we have a business service layer*".

Such layering allowed decentralisation of the governance, particularly the IT sourcing decisions, from the bank internal development team to external suppliers, which reduced the cost and time to market: "*So the classic examples would be Windows Phone application, that was completely done by third party, which was using our publicly - public facing APIs, so that wasn't done internally at all, the digital [team] engaged an organization in the states and they manage to create the app in less than two, three months, for amazing cost, almost half of the cost we can get locally".* Similarly, the Digital team has improved their responsive capability by using their in-house team to orchestrate and create required services for the mobile and the Web channels, without engaging the IT.

Both cases above indicate that the SOA process options such as reuse of services can only improve the IT responding capability if an appropriate governance structure is in place to encourage the adoption and execution of the options. The data analysis revealed that the IT must govern the lower layers of SOA to improve the option execution of core services and implementation of concerns such as security. Additionally, the higher layers of SOA, which are closer to the consumer channels and specific business processes, must be governed by the appropriate business units. Such decentralised decision-making allowed the business units to adopt the options, which are more aligned to their needs and improve their responding capability. Such adaptive governance structure improves the effect of SOA process options on the IT responding capability by enforcing an alignment with the IT and business objectives. While the IT can protect and nourish the highly reusable services and ensure main IT requirements such as security and auditing are addressed and governed, the business can focus on options that provide them with the fastest time to market. On this basis, the following proposition is suggested:

> **New Proposition (RP4-R5):** *The adaptive governance moderates the impact of SOA-enabled process options on the IT responding capabilities.*

5.7 Conclusion of Data Analysis

This chapter reviewed the collected data from the studied cases and reported the results of the case and cross-case analysis with the intention to answer the original four research questions of this study:

> RQ1: What are the SOA characteristics that affect the sensing component of IT agility?
>
> RQ2: What are the SOA characteristics that affect the responding component of IT agility?
>
> RQ3: How do the SOA embedded characteristics facilitate the sensing component of IT agility?
>
> RQ4: How do the SOA embedded characteristics facilitate the responding component of IT agility?

This chapter considered the initial theoretical propositions and refined them as the data analysis progressed.

Relying on the Yin's iterative analytical strategy (2009) for explanation building, this study coded each case study to first identify the characteristics that contributes to the IT sensing and responding capabilities. With the analysis of each case, the researcher reviewed and refined the initial codes extracted from the SOA and IT literature to reduce their duplication and provide a more precise definition. This study with reliance on the broader IT infrastructure literature and the analysis of the data collected from the cases identified six characteristics which impacted the IT sensing capability and twelve characteristics with impact on the IT responding capability. The outcome was the result of multiple iterations of descriptive coding (Miles & Huberman, 1994) for each case, which refined the NVivo nodes (SOA characteristics) captured in the researcher's database.

As the coding progressed, the researcher reviewed the relationships between the codes and compared them to the initial theoretical propositions made in the study. To ensure the study generates more insight, the researcher took an exploratory stand beyond the initial propositions to identify any new themes that can emerge from the data. Application of this approach led to two theoretical models presented in Figure 5-4 and Figure 5-5, which reflected the propositions suggested in this study. Figure 5-6 shows the integration of the two models depicting the SOA impact on the IT sensing and IT responding capabilities in conjunction with the complementary factors discussed in section 5.6.

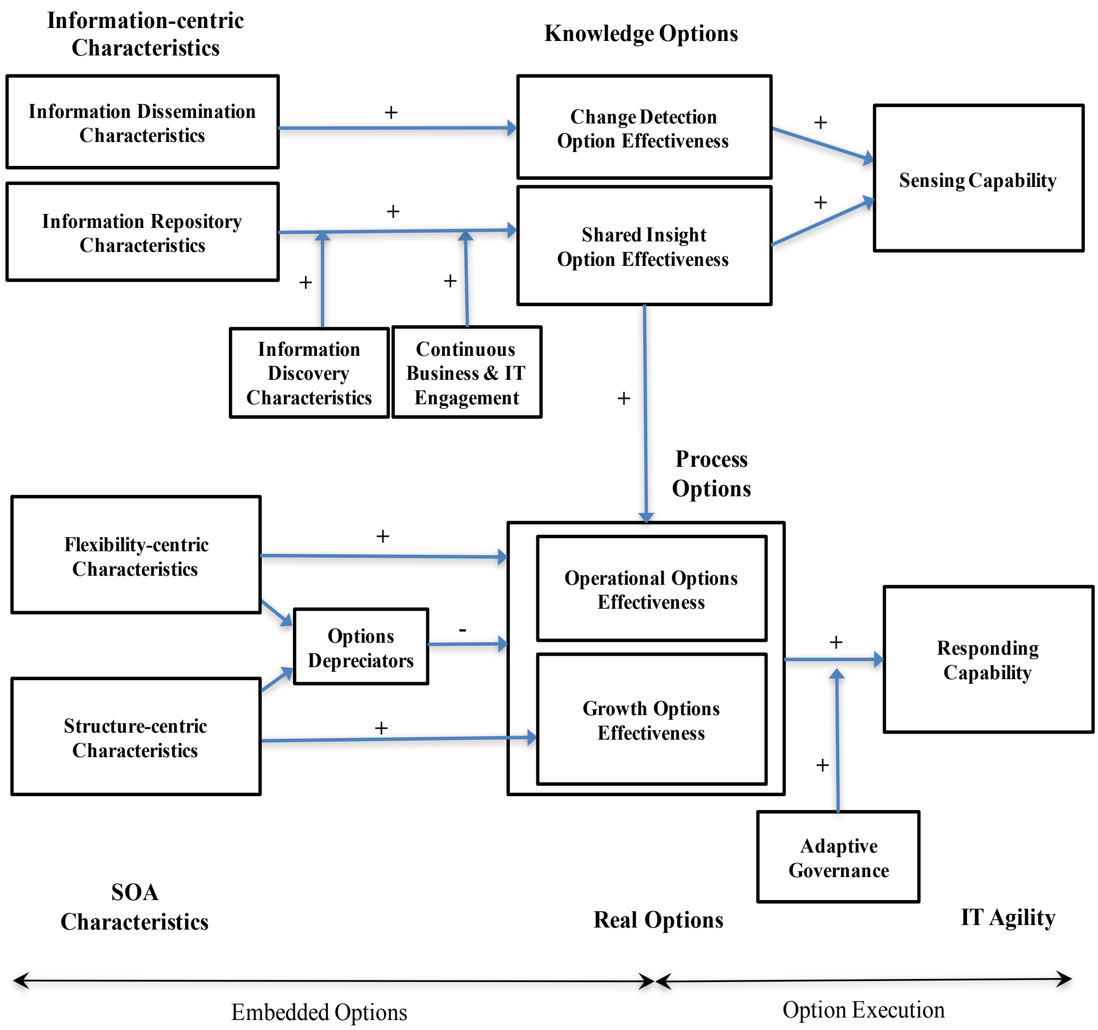


Figure 5-6 - Integrated theoretical model

The data analysis across all cases confirmed the positive impact of SOA on the IT agility. The impact on IT agility was driven by three main categories of SOA characteristics. The 'Information-centric' characteristics were the main contributor to the IT sensing capability. The data analysis revealed that the SOA 'information repository' characteristics improved the effectiveness of the IT and Business insight. Such impact was, however, contingent on the 'information discovery' characteristics, which facilitated the access and discovery of the information, and continuous business and IT engagement to improve the knowledge and visibility of IT capabilities. The moderation effect of 'information discovery' and 'continuous business and IT engagement' was discovered through the cross-case comparisons on cases that had different degrees of these factors and the varying level of insight gained from the same level of 'information repository' characteristics.

In addition to the 'information repository' characteristics, the data analysis showed that 'information dissemination' characteristics improve the effectiveness of 'change detection', which led to improved IT sensing capability.

The above findings led to the following propositions:

> **Refined Proposition (RP1-R1):** *The effectiveness of 'Change Detection' knowledge option is positively affected by the SOA 'Information Dissemination' characteristics already embedded in the information systems.*
>
> **Refined Proposition (RP2-R1):** *The effectiveness of 'Shared Insight' knowledge option is positively affected by the SOA 'Information Repository' characteristics already embedded in the information systems.*
>
> **Refined Proposition (RP3-R1):** *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the SOA 'Information Discovery' characteristic.*
>
> **New Proposition (RP4-R1):** *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the continuous business and IT engagement.*
>
> **Refined Proposition (RP1-P1):** *The IT-sensing capability is positively affected by the 'Change Detection' knowledge option that SOA offers.*
>
> **Refined Proposition (RP2-P1):** *The IT-sensing capability is positively affected by the 'Shared Insight' knowledge option that SOA offers.*

On the IT responding capability, the data analysis revealed that two groups of SOA characteristics affect the IT responding capability. These are: flexibility-centric characteristics and structure-centric characteristics. Each of these characteristics improved the effectiveness of the process options which improved the IT responding capability. The data analysis indicated that the effect of structure-centric characteristics is, however, limited to the options that are specific to developing and extending SOA systems. While the above characteristics improved the effectiveness of process options, they also increased factors which depreciated the value of options. Dependency and latency were the main options depreciators that were identified from the data.

Additionally, the data analysis confirmed that the process options improved the IT responding capability by facilitating the reconfiguration and combination of resources such as services and development resources to gain faster time to market. In addition to the SOA, the cross-case comparison revealed the role of shared insight in improving the effectiveness of process options. SOA-enabled Business and IT shared insight improved the effectiveness of process options by enabling additional options and better alignment of business requirements with the existing IT requirements. The other factor identified from the data, which affected the impact of process options on the IT responding capability was the adaptive governance. A governance structure, consistent with the SOA hierarchical layering, improved the execution of options. Such structure provided the IT with control and ownership to maintain and foster the core services that are the fabric of the enterprise. Similarly, it provided the business with autonomy to decide and choose options that improved their responding capability. The above leads the following propositions:

> **Proposition (RP1-R2):** *The effectiveness of operational options is positively affected by the SOA flexibility-centric characteristics already embedded in the information systems.*
>
> **Proposition (RP1-R3):** *The effectiveness of IT growth options is positively affected by the SOA flexibility-centric and structure-centric characteristics already embedded in the information systems.*
>
> **Proposition (RP1):** *The effectiveness of the IT process options is negatively affected by the options depreciators that SOA flexibility-centric and structure-centric characteristics create.*
>
> **New Proposition (RP4-R4):** *The SOA-enabled 'Shared Insight' positively affects the effectiveness of the process options.*
>
> **Proposition (RP1-P2):** *The IT responding capability is positively impacted by the effective process-based options that SOA offers.*
>
> **New Proposition (RP4-R5):** *The adaptive governance moderates the impact of SOA-enabled process options on the IT responding capabilities.*

The above discussions address the four research questions by identifying the SOA characteristics that improved the IT sensing and IT responding capabilities. The 'how' questions on the effect of SOA on IT agility were also addressed through the options theory and dynamic capabilities lenses by identifying the mechanisms involved in the effect of SOA on the IT agility.

Chapter 6. Conclusions and implications

This chapter provides a discussion on the findings of this research. Section 6.1 reviews the major findings of the current research including the conceptualisation of SOA, the effect of SOA on sensing and responding capabilities and the role of ‘continuous business and IT engagement’, ‘shared insight’ and ‘adaptive governance’ on the above effects. While the above findings are contrasted to their relevant literature, they are also reviewed from the Options theory perspective and how they contrast to the researcher’s understanding of options theory, particularly in a context that there are competing and interdependent options available.

Section 6.2 provides a summary of the research gaps, the four research questions and how the current study addressed them. The above section depicts an integrated conceptual model extracted from the data, with twelve refined or new theoretical propositions.

This leads to the discussions on the theoretical contributions of this research as well as the implications for the practitioners in section 6.3. This study makes a number of contributions to the SOA, Agility and Options theory literature. The contributions can be categorised in five high-level groups:

- Conceptualisation of the SOA by three themes of ‘information-centric’, ‘structure-centric’ and ‘flexibility-centric’, which captured the SOA behaviour at a level suitable for theorisation. The identification of the above themes and their operationalisation address the issue reported in the SOA literature (Joachim, 2011) on how SOA should be conceptualised in the study of SOA business value.
- A novel contribution to the SOA and IT governance literature. The findings in the current study provide a different view on how SOA governance must be implemented to nurture agility as well as future option creation. By focusing on the governance structure for an agile IT, and by decomposing systems to different service layers, this study extends the existing IT literature by proposing an adaptive governance structure that aligns to the software layers structured based on the rate of change and their level of reusability. The above finding provides an alternative view on what the SOA literature proposes, which is typically central and dedicated governance structure for the SOA.

- Theory building by unboxing the monolithic black box (Orlikowski & Iacono, 2001) of systems and bringing the two SOA and IT literature closer. The extant IT literature typically limits its focus on IT infrastructure or broader IT capabilities and how they create a business value. This study by decomposing a system to its underlying characteristics provide insight into the characteristics that contribute to the agility and their intermediate organisational impacts. This study takes preliminary steps to present SOA using familiar constructs defined in the IT literature and then show the impact of system characteristics on its business value.
- Identification of antecedents of IT sensing and responding capabilities in the context of SOA and their interplay. While there are a few studies that reviewed the IT agility and its antecedents, this is the first study that considers both sensing and responding capabilities of IT agility. As already reported in the broader agility literature (Roberts & Grover, 2012), the sensing capability is essential in achieving agility. In addition to the responding capability, the current research findings shed light on the antecedents of the sensing capability and their effect on the IT responding capability.
- A novel contribution to the options theory and to Sambamurthy et al.'s (2003) theorisation of IT as a digital options generator. To the researcher's knowledge, this is the first study that operationalises the concept of digital options and considers SOA from the options theory perspective. The application of options theory in this study extends the extant knowledge by showing the conditions required to improve the effectiveness of embedded options and the role of adaptive governance in the execution of options. Finally, options could co-exist and could impact one another. This study made a novel contribution by identifying and conceptualising the interplay between the options.

Similar to many other studies, this research has a few limitations. Section 6.4 describes the four limitations of this study, which are caused by its selected research methodology and research boundaries chosen in the current research.

Finally, Section 6.5 outlines the future research opportunities in service design, agility and options theory and section 6.6 concludes the current dissertation.

6.1 Discussion of Results

This section contrasts the research findings against the extant SOA and IT literature and identifies similarity and differences among them.

Subsection 6.1.1 discusses how this study conceptualised the SOA compared to the extant studies in the SOA literature. Subsection 6.1.2 then discusses SOA information-centric characteristics and how they affect IT sensing capabilities. The impact of 'Continuous Business and IT engagement' on the above relationship is then discussed and compared to the literature in subsection 6.1.3.

The impact of SOA characteristics on the IT responding capabilities is discussed and contrasted to the current SOA and IT literature in subsection 6.1.4, with the effect of Shared insight being covered in 6.1.5.

The subject of subsection 6.1.6 is adaptive governance and its role in the execution of embedded options. Finally, subsection 6.1.7 reviews the options theory and explicates the contribution of this study to the options theory.

6.1.1 Conceptualisation of SOA

The existing SOA literature provided limited conceptualisation of SOA (Beimborn & Joachim, 2010; Daskalakis & Mantas, 2008; Joachim, Beimborn, Schlosser, et al., 2011; Oh et al., 2007), which led to a call on how SOA should be conceptualised in the assessment of SOA value (Joachim, 2011; Richter & Basten, 2016; J. Schelp & Aier, 2009).

This study took a few steps to answer the above call and conceptualised the SOA based on:

- (i) Adoption of a theoretical perspective,
- (ii) Use of SOA characteristics as the fundamental building block of the SOA concept,
- (iii) Comparison of SOA concepts to the IT literature to sharpen the definitions and reduce duplication,
- (iv) Inclusion of design characteristics as well as characteristics of the SOA platform, and
- (v) Grouping of characteristics based on their common themes, and

The researcher's conceptualisation of SOA based on the above considerations extends the extant effort (Beimborn & Joachim, 2010; Daskalakis & Mantas, 2008; Joachim, Beimborn, Schlosser, et al., 2011; Oh et al., 2007; Richter & Basten, 2016) in measuring and conceptualising SOA in the study of SOA business value.

By adopting a resource-based theoretical perspective, this study conceptualised the SOA as an intangible IT asset (Ross et al., 1996). This is based on the assumption that SOA is concerned with rules governing the overarching structure and properties of systems (Tiwana & Konsynski, 2010), which included the rules embedded in both the services and the SOA platform that served the services. The rules are knowledge assets that can be captured in information repositories (A. S. Bharadwaj, 2000). Compare to the existing studies (Beimborn & Joachim, 2010; Daskalakis & Mantas, 2008; Joachim, Beimborn, Schlosser, et al., 2011; Oh et al., 2007), review of characteristics of services and SOA platform together provides a more holistic picture on the role of each, particularly in achieving the IT agility.

Similarly, the conceptualisation of SOA in the current study relies on the SOA characteristics that define the SOA behaviour, and excludes implementation technologies. This makes the findings more generalizable by being independent of technologies that evolve over time, such as XML, SOAP and JSON.

This study initially used the existing SOA characteristics reported in the SOA literature (Abelein et al., 2009; Baskerville et al., 2005; Erl, 2004, 2005; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003, 2008) as reported in section 2.3.6.2, however contrasted them to the IT infrastructure literature to reduce the duplication and sharpen the definitions. This process provided eleven SOA characteristics, as listed in Appendix B – A Priori Constructs and Mapping to SOA Characteristics. The researcher's data analysis, however, extended this list and identified four more SOA characteristics, in addition to refinement to the existing concepts.

By considering the nature of the characteristics, this study grouped and conceptualised these fifteen SOA characteristics to three main themes of *information-centric, structure-centric* and *flexibility-centric* characteristics. The above grouping provided an opportunity to shift the discussions from individual fifteen characteristics to higher order constructs to study their effects on the IT agility or its mediators.

As the novel contribution of this study, the data analysis showed that the SOA information-centric characteristics are the main contributor to the IT sensing capability and the SOA structure-centric and flexibility-centric characteristics are the main contributor to the IT responding capability. These findings answered the first two research questions of the current study:

RQ1: What are the SOA characteristics that affect the IT sensing capability?

RQ2: What are the SOA characteristics that affect the IT responding capability?

The above three SOA themes and their underlying characteristics are further discussed next.

6.1.2 Discussion on SOA characteristics impacting Knowledge options and Sensing capability

As shown in the data analysis chapter, the SOA information-centric characteristics were the key contributors to the IT sensing capabilities. The information repository and discovery improved the IT and business shared insight by capturing and publishing technical and business-oriented design information embedded in the services, whereas the information dissemination provided insight on the behaviour of the system when it was operating and when the system required attention. Figure 6-1 shows the above relationships.

As discussed in section 2.3.5 of the literature review, despite the importance of the IT sensing capability, the existing SOA studies are silent on the impact of the SOA on the IT sensing capability. The above findings take steps to shed light on the impact of SOA on the IT sensing capability.

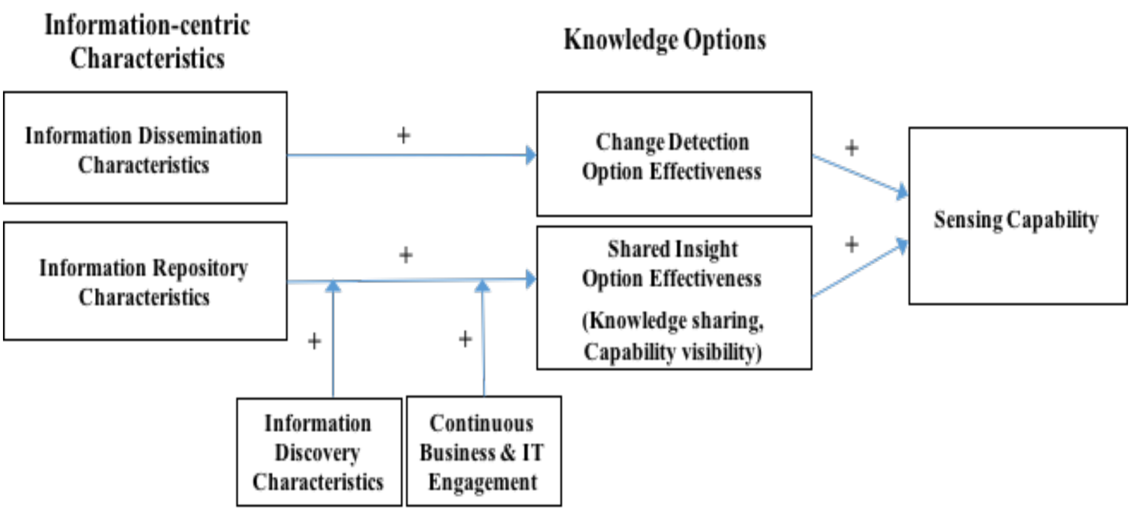


Figure 6-1 - Impact of SOA Information-centric characteristics on IT sensing capability

As shown in the data analysis, the shared insight within IT and with business was based on the knowledge sharing and capability visibility that the SOA information-centric characteristics provided.

Regarding the knowledge sharing, the researcher's findings are consistent with (Legner & Heutschi, 2007) that reported improvement in the IT knowledge due to interfaces and business process modelling and their corresponding documentations. Additionally, the current study found that standardisation, because of consistency that it provides in problem solving and business orientation, improved knowledge sharing within IT and with business.

On the capability visibility between the business and the IT, the analysis showed that a shared visibility of IT capabilities, e.g. available services to address a particular requirement, provides insight on the optimum changes required in the system and the available options to meet the requirements promptly. In the studied cases, the SOA-enabled shared insight allowed the business and IT to negotiate and consider different options to address their capability gaps quickly. Such options may include reduction of requirements to minimum viable product or selection of off the shelf products.

On the effect of SOA on shared insight, the findings revealed that such impact is contingent on existence of two factors: SOA information discovery and continuous business and IT engagement. Based on the researcher's findings, SOA cannot be effective in creating shared insight between business and IT unless the above two factors exist. Without SOA information discovery characteristics such as classification of design information and publishing them in an accessible manner, information captured in information repository will not be accessible and will not create the shared insight. The continuous business and IT engagement has similar effect, which will be discussed further in the next sub-section.

The above findings differ from the existing SOA findings (Legner & Heutschi, 2007; Yoon & Carter, 2007) that suggested unconditional improvement of business and IT communication and consequently alignment between business and IT.

In the broader agility context, the above findings also confirm and extend the existing agility literature (Teece, 2007) which identified extant organisational knowledge as a requirement for creation of new knowledge and interpretation of the market activities as part of the sensing activities. While the business and the IT must constantly search and explore technologies and markets to identify and shape opportunities (Benner & Tushman, 2003; March, 1991), their clear understanding of existing IT capabilities in the form of available SOA services will assist them to understand changes required in the system and decide on the options that can realise the opportunity faster with less effort. Therefore, from the IT agility perspective, SOA-enabled shared insight between business and IT becomes essential to improve the IT sensing capabilities.

6.1.3 Continuous Business and IT Engagement

As discussed above, this study showed the role of continuous business and IT engagement in creation of organisational shared insight. The data analysis revealed that that the ongoing engagement between business and IT considerably increased the effect of SOA information repository characteristics on the shared insight.

The SOA literature has reported improved business and IT communication and consequently alignment (Legner & Heutschi, 2007; Yoon & Carter, 2007) as the benefits of SOA. The above improvement in form of close IT/business collaboration was however marginally supported in another study of SOA (Joachim, Beimborn, Schlosser, et al., 2011). As per the analysis, a close and continuous engagement between the business and IT is essential to allow the SOA to improve the peripheral knowledge and consequently the business and IT communication through a shared vocabulary. It is such close engagement that nurtures the learning, collective sense-making, and perspective sharing which are required for improved communication between business and IT.

The findings of this study showed that those business units that had close engagement with the IT gained better insight on the available SOA capabilities and services. Such insight combined with the SOA layering provided the business with recognition of the available options, and exploration of new options such as engagement of external development teams. The findings of the current study showed that such dynamic capabilities improved the organisation responsiveness to the market needs.

Another benefit reported in one of the cases was the creation of trust between business and IT through such close engagement. The business & IT collaboration facilitated the removal of walls between the teams to work as one team to reduce the coordination overhead and engage in a collective sense-making, joint option creation and option selection.

6.1.4 Discussion on SOA characteristics impacting process options and IT responding capability

As shown in the data analysis chapter, the SOA flexibility-centric and structure-centric characteristics improved the IT responding capability by creating operational and growth options. The SOA flexibility-centric characteristics impacted the effectiveness of the growth and operational options, whereas the SOA structure-centric characteristics only impacted the growth options. Additionally, the data analysis showed that both of these two groups created options depreciators which have negative effect on the effectiveness of the operational and growth options.

The findings on the positive impact of SOA flexibility-centric characteristics on the effectiveness of processes options and consequently on the IT responding capabilities are consistent with the SOA and IT literature (Abelein et al., 2009; Baskerville et al., 2005; S. S. Bharadwaj, Chauhan, & Raman, 2015; Legner & Heutschi, 2007; Luthria & Rabhi, 2009c; Moitra & Ganesh, 2005; J. Schelp & Aier, 2009). Similar to the researcher's findings, Moitra & Ganesh (Moitra & Ganesh, 2005) recognised that defining and implementing flexible business processes supported by flexible IT systems is of significance to the organizations because this would allow them to collaborate with partners in new ways resulting in inimitability of the processes; this in turn allows them to adapt to the changing environment. Similarly, the SOA literature (Baskerville et al., 2005; Joachim, Beimborn, Schlosser, et al., 2011) has also reported flexibility and adaptability as one of the SOA benefits, which improves the IT responsiveness (J. Schelp & Aier, 2009).

While the researcher's analysis reflected on the above benefits and identified the specific SOA characteristics that provide technical IT flexibility, the analysis also showed that additional integration between services, due to their distributed architecture, can increase the latency between the services, which can inhibit the benefits the flexibility-centric characteristics offer to the IT. The researcher's findings presented such negative impact in the form of its impact on the effectiveness of the options available to the IT.

In addition to the flexibility-centric characteristics, the data analysis revealed another group of characteristics with positive impacts on the growth options. Unlike the flexibility-centric characteristics, the structure-centric characteristics provide options to reuse and combine the existing IT assets to reduce the development cycles. The IT infrastructure literature (Byrd & Turner, 2000; Duncan, 1995; Knoll & Jarvenpaa, 1994; Tiwana et al., 2010) has limited focus on the structural characteristics, and usually limits its scope to the modularity of components and the flexibility achieved through modularity. The SOA literature (Erl, 2004; Joachim, 2011; Legner & Heutschi, 2007; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003; Schulte et al., 2008) reports certain structural characteristics of the services, however does not focus on their interactions. The researcher's findings extend the above and showed that there are additional characteristics for the services both individually and collectively, which influence their suitability for option creation, such as reuse and further combination to create new value propositions for the business.

The analysis showed that granularity, by its focus on decomposability, creates more reusable components by breaking a service to smaller pieces. This characteristic however needs to be considered in conjunction with generalisability to ensure the service will not be specific to one particular use case. Similarly, each service needs to be autonomous and with operations being cohesion to reduce the dependency between the services. Combination of the above characteristics together suggests a granularity for the services which improve their reuse and reduce their dependencies. While the above factors have been reported in desperate SOA studies (Erl, 2004; Joachim, 2011; Legner & Heutschi, 2007; Luthria & Rabhi, 2009b; Newcomer & Lomow, 2004; Papazoglou, 2003; Schulte et al., 2008), this study refined these concepts and showed how they interact to provide reuse.

In addition to the above service characteristics, the analysis revealed a new and important factor concerning the structure of services. Hierarchical layering as a new factor focused on the overall structure of services together in order to reduce dependency and impact of change on the system and processes.

Comparing the above results to the Simon's theory of evolutionary selection (1962) presents an interesting outlook, particularly concerning the boundary condition of the evolutionary selection theory. the Simon's theory of evolutionary selection (1962) argues that decomposable complex systems evolve faster because they require less time to evolve by recombination and will undergo more diverse evolutionary experiments. He proposed that complex systems that evolve at a faster rate and with greater diversity are more likely to evolve to achieve better fit with their environment than those that do not possess these traits (Simon, 2002). The researcher's findings showed that services with high structure-centric characteristics shared many characteristics of a decomposable complex system. Furthermore, a change in the system will require less time and there will be greater composition opportunities. With the core services being generalizable, they are more likely to align to the new requirements without going through a change.

However, the findings also suggest that achieving a nearly decomposable system with almost no dependency is not an achievable condition, at least in the context of the current study. Increase of services granularity with service orchestration and reuse, will automatically increase the dependency between the services. To recognise the omitted variable of dependency, the researcher's finding recognises options depreciators as a mediating factor for process option effectiveness, which must be considered when deciding on the granularity and layering of the services.

In the current study, the use of options theory provided an alternative lens to the evolutionary selection by explaining the characteristics that influence the evolution of the system and the options that are created for the future evolution of the system. The more effective the options are in respect to their range and richness, the faster the system will achieve its fitness with its changing environment. The use of options theory provides a better explanation of the system evolution due to the role of options depreciators and the knowledge options.

Changing the discussion to the impact of options on the IT responding capabilities, the findings of the current study support and extend the existing IT literature (1994; Dove, 2001; Malone & Crowston, 1994; Mohr & Nevin, 1990) and showed that the SOA-enabled process options provide spare capability to the IT to be used when required. The available capabilities reduce the development or operational management cycles, which in turn, improve the time to market and the IT responsiveness. The options also provided the IT management or IT development teams with flexibility to select the best option that suits the emerging requirements. This finding is consistent with the suggestion from Copeland et al. (1994) that real options provide flexibility to managerial decision making by allowing managers to use tangible and intangible assets in completely new or alternative ways in the future without having the obligation to do so. When making resource-committing decisions, assets can be arranged to enhance alternative uses. The findings also indicated that when additional options are available to the business and the IT team, their level of autonomy in respect to software development increases. Such autonomy reduced the coordination overhead between the business and IT.

On the above effect however, this study shows situations where available and effective options were not perceived or perceived, however not adopted by the development teams. The analysis revealed two additional factors that can influence the above and the effectiveness of process options and their adoption to improve the IT responsiveness. The next two sections discuss the role of shared insight and adaptive governance in the responsiveness of IT.

6.1.5 Role of Shared Insight on Option Effectiveness

The role of shared insight in improving the effectiveness of options was clearly observed during data analysis. Shared insight through knowledge sharing between IT and with the business, and shared visibility of capabilities increased the range and richness of process options.

As previously discussed, the SOA information-centric characteristics, when combined with continuous business engagement, improved knowledge sharing and visibility of IT capabilities among the business and the IT. In several instances, the shared insight between IT and business improved the process option effectiveness in form of identification of new options and reduction of coordination overheads.

The role of shared insight is consistent with the broader IT literature (Dove, 2001; Sarker & Sarker, 2009) which suggest that the firm's ability to respond to market opportunities depends on the coordination and flexibility of its products and processes. For instance, in software development the collaboration and coordination of development teams, both internally and with clients, has improved the agility of software development (Sarker & Sarker, 2009).

6.1.6 Adaptive Governance

The cross-case analysis showed that the governance is an additional factor that affects the impact of process options on the IT responsiveness. The analysis highlighted that the type of governance should be dependent on the service layering, with the lower layers requiring more centralised governance and the higher layers requiring a decentralised governance model.

The above finding is different from the extant SOA literature (Bieberstein et al., 2005; J. H. Lee et al., 2010; Schepers et al., 2008; Walker, 2007), which suggests a central governance structure for the SOA to manage the SOA in the organisation.

This study found that a centralised governance at the lower layers provided higher level of reusability for the core services and closer alignment to the business needs due to the services being more generalised and less specific to particular use cases (option richness). In all of the researcher's cases, the decision-making of the services and platform was managed by the IT central decision-making body rather than a distinct SOA body. This included the Airline, which had an SOA Centre of Excellence. The centre of excellence while promoted the service adoption and coordinated activities concerning the service orientation, still relied on the existing IT Architecture board for decisions concerning the platform and the service evolution.

Findings concerning the governance of higher layers of services are different from the above. The analysis showed that a more decentralised governance model provides flexibility and agility required by the business. On the broader IT agility literature, the researcher's findings extend the Tiwana and Konsynski (2010)'s work, for the complementarity role of IT architecture modularity and IT governance decentralization and how the harmony between the technical and organizational design of the IT enhances IT agility. The existing study (Tiwana & Konsynski, 2010) showed that a decentralised governance has both positive and negative impacts on the IT agility. While decentralised governance provides further autonomy to the line functions (the business) to make decisions faster and more aligned with their needs, it also hindered the IT agility, possibly due to the overt interdepartmental coordination which can overwhelm its advantage (Tiwana & Konsynski, 2010).

Unlike the Tiwana and Konsynski (2010)'s work, the decentralisation of governance in the current study did not hinder agility. This could be due to the fundamental decisions on standards, core services and core infrastructure have already been made available centrally by the IT, hence reduced the interdepartmental coordination.

The findings are also confirmed by a more recent study (Tiwana & Kim, 2015), which showed centralisation of the IT governance for the IT infrastructure, when the line function' technical knowledge is high, will improve IT strategic agility. Furthermore, the above study also found that IT governance decentralisation for the IT applications, when the IT business knowledge is high, again improves the IT strategic agility. The researcher's findings take an additional step by decomposing the IT applications to service layers and proposing different governance models to the layers of services. The use of options also explains the proposed relationship and the effect of adaptive governance and shared insight on the IT responding capability.

The findings from the current study therefore extend the SOA and IT governance literature by considering a layered approach for the services and IT infrastructure, and adopting an appropriate governance structure for each. Decision on standards, core services, and SOA platform require organisation-wide knowledge of IT portfolio and technical expertise, which is mainly available within the IT; and would require a central governance structure managed by the IT unit. Such structure promotes the development of core services for further reuse, adoption of suitable platform, and the development and adoption of proper standards to create rigidity in the IT architecture, which in turn will foster greater agility for the business. The paradox role of rigidity in the lower layers to foster flexibility in the higher layers has been observed and considered by other scholars before (Star & Ruhleder, 1996; Tiwana & Konsynski, 2010).

On the higher layers, the rigidity created by the design patterns, standards and core services embed a non-overt control for concerns such as security, which allows the IT to give up its central governance for further business autonomy and agility. Considering the business knowledge on the market needs and opportunities, the decentralised governance on the higher layers provide the business with the power to decide on how they can embrace the opportunities and meet their time to market. The above relies on the nexus of collocating the decision making with the party who has the knowledge to make the decision (Jensen & Meckling, 1992).

6.1.7 Contribution to the Options Theory

The premise of the real option paradigm is the flexibility in utilising resources and capabilities to benefit from the uncertainty in the business environment. In practice, however, there have been debates whether the firms can capitalise on these opportunities and convert them to value (Bennett, 2012).

The current study took a few extra steps to shed light on how options theory can be applied to the IT management. It identified the conditions that improve the effectiveness of the options and the conditions that improve the real options decision making.

The first finding was related to the options when competing and interdependent options are available. While the extant literature reviews the IT option (Fichman et al., 2005;

Tiwana et al., 2006) and its value assessment (Gamba & Fusari, 2009; R. L. Kumar, 2004) and its antecedents (Fichman, 2004), the analysis found that options do not operate in isolation and can have implications on the values that each and every one of them may generate. The researcher conceptualised this effect as options depreciators. The current study showed that dependency between options and latency work as options depreciators, and will reduce the effectiveness of the options. While the measures of options depreciators are expected to be context specific, the concept of options depreciators could be generalizable to other applications of Options theory. For instance, in the review of the real options in the project portfolio, the dependency between projects and resourcing tensions could be factors that could create options depreciators.

On the types of options, the current study assessed both the knowledge and process options and the interplay between the two. The 'Change Detection' and 'Shared Insight' were the two knowledge options that SOA facilitated to improve the IT sensing capabilities. Additionally, on the process options, the researcher found several options that are available to the organisation to maintain a system or extend the system. Gradual transition, change service provider, wrap and simulate behaviour, extension in stages, extension by service reuse, design reuse, and extension by connectivity are examples of such options. The findings showed that the knowledge options when executed could have a high payoff in sensing system changes. Similarly, the process options facilitated the responding capabilities.

On the interactions among the options, the analysis found that shared insight is a necessary condition for the effectiveness of process options. The role of knowledge option can be explained on three fronts. Firstly, the created knowledge from the SOA information-centric characteristics can be the source of identifying new options, which can be beneficial to the organisation. This is particularly the case when there is a sizable portfolio of options available that can be executed as they stand, or be combined together to address a more complex need.

Secondly, the findings show that the embedded options while can be beneficial and deliver a high payoff, they might not be fully aligned with the market needs. In such cases, the option requires further change and investment to address the market needs. This not only increases the transaction cost but also increases the time to market. The analysis showed that when the business has visibility of available options and their constraints, they can try alternative strategies which are more aligned to the available options in an attempt to minimise the time and effort in addressing the market needs.

Thirdly, the shared insight provides the business and IT with the awareness of the options that are available for execution. The additional managerial flexibility gained through the options is only viable and valuable if there are adequate levels of knowledge to dig for options prospects (Rangan, 1998).

The above findings nicely align with the real options realism reported in the strategic management literature that suggests managerial flexibility should be explored and planned before being executed. In other words, the firms must be equipped with knowledge, systems, and capabilities of detecting and implementing real options before starting exploitation (Kogut & Kulatilaka, 2004; Rangan, 1998).

An additional factor that the data analysis identified was the adaptive governance and its role in option execution. The extant literature (McGrath, 1997; Miller, 2002) already confirms that organisational design is one of the required conditions for the realisation of the option value. This can simply be justified on the basis of the initial three conditions of real options including uncertainty regarding net payoffs, irreversibility in the initial cost, and managerial decision-making flexibility regarding how options are executed (Dixit, 1994). The researcher's argument mainly lies in the managerial decision-making flexibility associated with each layer that the options reside in. The lower layers, as discussed before, facilitate the options that are associated with the core enterprise systems and IT infrastructure. In this case, the IT, due to its knowledge of the enterprise-wide IT portfolio and deep technical knowledge, is in a better position to centrally identify and execute the options in the lower layers.

However, the higher layers require the knowledge and awareness of market needs and opportunities, with specialisation required for decision-making shifting from technical to business. The business units, in this case, are in a better position to exercise the options that best meet their requirements. The adaptive governance fosters the real option and reduce the risks of shadow options, as options which are not recognised (Bowman & Hurry, 1993).

Finally, while many studies focus on the options and their benefits, this study took an additional step in assessing the effectiveness of option in dealing with unanticipated market needs and the antecedents that can affect this effectiveness before assessing the benefits of the options. By proposing a conceptual model, this study showed that information-centric characteristics can be a source of knowledge options, particularly when there is a continuous and close engagement between business and IT. The structure-centric and flexibility-centric characteristics also affect the process options, which is also improved by the knowledge options. While the SOA creates and improve options, their execution requires a suitable governance structure to foster the execution of the options.

6.2 Conclusion about the Research Problem

The current study investigated the effects of SOA on the IT agility by adopting the theoretical foundations of real options theory (Bowman & Hurry, 1993) and dynamic capabilities (Eisenhardt & Martin, 2000; Teece et al., 1997).

To that effect, this study focused on the question of 'How does SOA impact the IT agility?' Answer to this question clarified the gap in the literature concerning the conflicting reports of SOA benefits. While a number of studies reported business agility as the benefit of SOA, there were reports of SOA introducing complexity to the IT processes and systems, which in turn, can hinder agility. Answer to the above question also provided insight on how SOA improves the sensing component of the IT agility, which is essential in achieving agility.

The above gaps with opportunity to further theorise in the SOA research motivated this study. This is particularly when agility has been reported as one of the main benefits of SOA, which can in turn enable the organisations to outperform their rivals in the current competitive environment.

To address the above question, the researcher developed the following four sub questions:

> RQ1: What are the SOA characteristics that affect the sensing component of IT agility?
>
> RQ2: What are the SOA characteristics that affect the responding component of IT agility?
>
> RQ3: How do the SOA embedded characteristics facilitate the sensing component of IT agility?
>
> RQ4: How do the SOA embedded characteristics facilitate the responding component of IT agility?

To answer the above questions, the study had to take a theory building approach due to the extant literature not offering any theoretical explanation on how SOA delivers agility. As already reported in the SOA literature, the lack of theorisation and conceptualisation of SOA has affected the studies of SOA and how it creates value (Joachim, 2011). This issue became more apparent in the course of this research, which showed while SOA has been adopted and visible in the information systems and IT infrastructure landscape, the broader IT literature does not fully recognise the SOA characteristics. Similarly, the SOA literature has made limited attempt to leverage off the broader IT literature and offer a conceptualisation that can bridge the gap between the two literatures.

To answer the above research questions, this study had to first define the IT agility. This is due to the fact that current literature offers a number of viewpoints depending on its source discipline and field. After a thorough literature review, this study defined IT agility as:

> *"Ability of IT to swiftly identify needed changes in its information systems and implement required changes in an uncertain environment in order to support the business to survive and thrive."*

As reflected in the research questions, the IT agility relies on two capabilities of sensing to identify the change, and responding, to implement a change swiftly. Similarly, the SOA as a design paradigm was conceptualised as an intangible asset as per the resource-based view (Barney, 1986; Makadok, 2001; Wernerfelt, 1984).

To theorise the relationship between the SOA and the IT agility, this study took an opportunistic view by adopting the real options theory as its main theoretical stand in conjunction with the dynamic capabilities. Inspired by the Sambamurthy et al. (2003) concept of *digital options*, this study took the initiative to further develop the digital options and operationalise them in the context of SOA. The researcher's initial conceptual framework proposed a few propositions which explained how SOA through its characteristics create options that in turn, improve the two main capabilities of the IT agility: IT sensing and IT responding capabilities. The initial framework was used as a "sensitizing device" at a high level to view the SOA and its relationship with the IT agility through the options lens (Gregor, 2006). The framework assists to identify classes of relevant variables and their interrelationships (Teece, 2007).

To answer the research questions, the researcher chose a post-positive case study as its research method to develop a mid-range theory on how SOA impacts the IT agility. Three main cases with multiple embedded cases were selected to provide the empirical data required for development of a conceptual model and theoretical propositions. The three main cases included one Australian Airline, one Australian Airport, and one International Bank. The cases were selected on the basis of providing literal and theoretical replications (Yin, 1994).

The data collection method of the study was semi-structured interviews conducted based on an interview protocol consistent with the theoretical framework and the a priori constructs extracted from the literature. To gain data saturation, twenty-two interviews were conducted with the IT executives, business managers who sponsored SOA projects, and architects and project delivery teams in the three selected organisations. The collected data were then analysed and summarised using the descriptive and pattern coding methods and stored in the researcher's NVIVO software tool. The initial rounds of data analysis resulted in over 200 nodes and 179 relationship nodes, which were refined and merged as the data analysis progressed and patterns and themes emerged from the data. The researcher also captured over 60 memos describing the interesting themes or concepts as the data analysis progressed. The data analysis captured from the field resulted in an integrated conceptual model as shown in Figure 6-2.

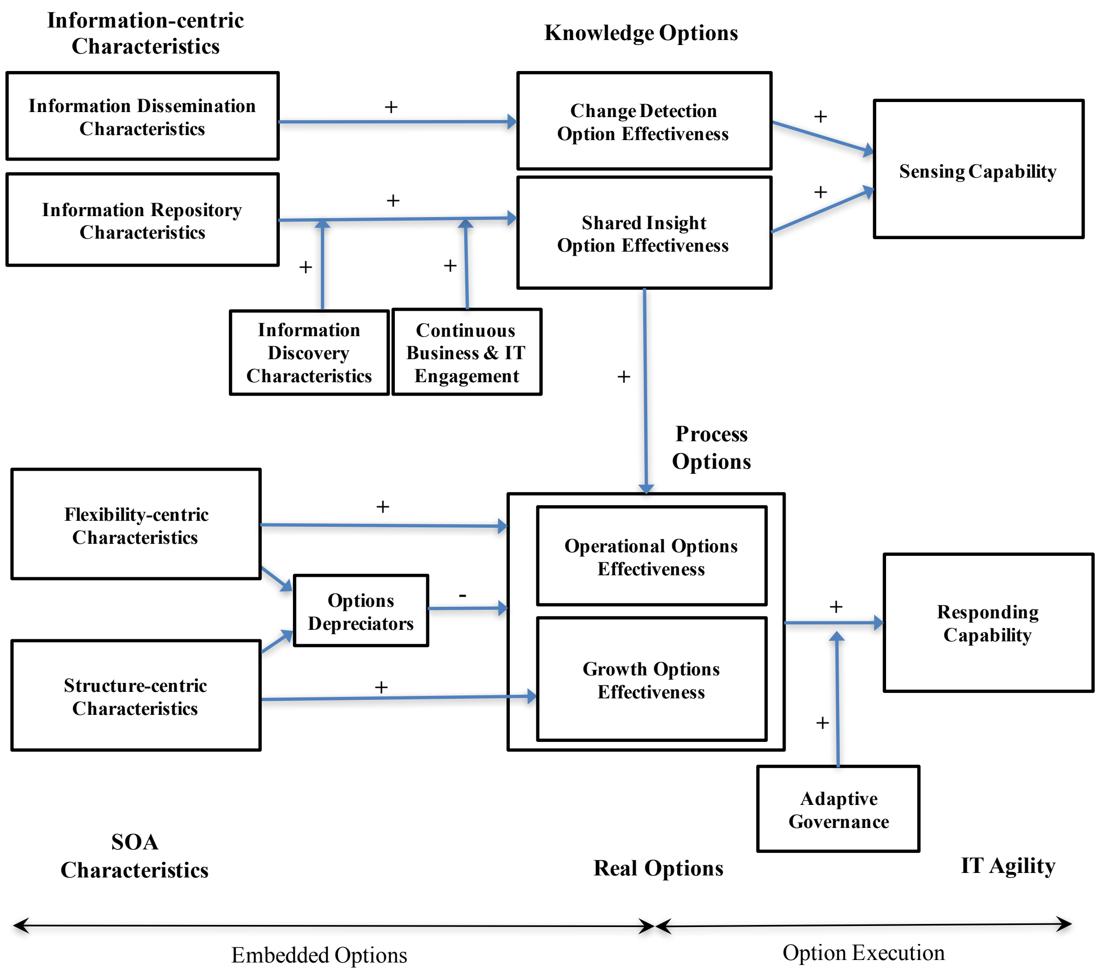


Figure 6-2 - Integrated theoretical model

The pattern coding of the SOA characteristics captured from the cases presented three common themes including information-centric characteristics, flexibility-centric and structure-centric characteristics. The SOA information-centric characteristics through their capacity to capture and discover service information and dissemination of events presented opportunities to improve the shared insight and change detection in the organisation. The shared insight was particularly in the form of knowledge sharing and shared visibility of IT capabilities when led to identification of required changes in the system and possible options to address the requirements. The cross-case comparison showed that the created shared insight is only strong when the business and IT have close and continuous engagement.

While the SOA literature (Legner & Heutschi, 2007) had desperate information available concerning the impact of SOA on the IT and business knowledge, the researcher's findings take the current state of knowledge to the next level by identifying the characteristics that contribute to the knowledge options and how they improve the IT sensing capabilities. Additionally, the researcher's findings extend the existing agility literature which focus on sensing the change events by searching and exploring technologies and markets to identify and shape opportunities (Benner & Tushman, 2003; March, 1991). While the researcher's findings do not discount the above, it revealed the role of shared insight in understanding the impact and significance of change events and the options available to deal with the change. Therefore, from the IT agility perspective, SOA-enabled business and IT insight becomes essential to support the processes involved in the IT sensing.

The above findings were reflected in the following refined propositions:

> **Refined Proposition (RP1-R1):** *The effectiveness of 'Change Detection' knowledge option is positively affected by the SOA 'Information Dissemination' characteristics already embedded in the information systems.*
>
> **Refined Proposition (RP1-P1):** *The IT-sensing capability is positively affected by the 'Change Detection' knowledge option that SOA offers.*
>
> **Refined Proposition (RP2-R1):** *The effectiveness of 'Shared Insight' knowledge option is positively affected by the SOA 'Information Repository' characteristics already embedded in the information systems.*
>
> **Refined Proposition (RP3-R1):** *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the SOA 'Information Discovery' characteristic.*
>
> **New Proposition (RP4-R1):** *The impact of the SOA 'Information Repository' characteristics on the effectiveness of 'Shared Insight' knowledge options is moderated by the continuous business and IT engagement.*
>
> **Refined Proposition (RP2-P1):** *The IT-sensing capability is positively affected by the 'Shared Insight' knowledge option that SOA offers.*

On the SOA characteristics that affected the processes, the analysis showed that the characteristics could be grouped into two categories: flexibility-centric and structure-centric. The flexibility-centric characteristics were those characteristics, which provided choices to the IT managers to adapt to a change with less effort. The analysis showed that SOA platform being a type of IT infrastructure shares similar characteristics as reported in the IT literature (Byrd & Turner, 2000; Duncan, 1995; Fink & Neumann, 2009). SOA, however, has a lot of focus on adaptability and standardisation.

The last category of characteristics was the structure-centric characteristics. Traditionally, the IT literature limits its focus to the IT infrastructure and does not normally get involved in the underlying structure of systems. With the SOA however, the internal structure of systems become visible. The study findings identified two groups of characteristics, which collectively improved the IT agility. The first group was characteristics that described the structure of individual services to improve their reuse and reduce their dependencies on each other. Characteristics such as granularity, generalizability, service cohesion, loosely coupled and autonomy together describe the main characteristics of each service for reducing its dependency on other services and increase its fitness for higher use case scenarios. The second group focused on the structure of services together and how they should be positioned to improve their resilience to change. The hierarchical layering and composability were the two identified characteristics in this group. The hierarchical layering consistent with Simon's decomposable complex systems concept (1962) structured the services with minimum dependency on one another, with core and fine-grained services in the lower layers and the services with higher granularity and logics closer to the service consumer channels in the higher layers. Such structure showed a reduction in the impact of change on the underlying and highly reusable services. It also provides opportunity to apply different levels of control and governance on each layer.

The following refined propositions reflect the above findings:

> **Proposition (RP1-R2):** *The effectiveness of operational options is positively affected by the SOA flexibility-centric characteristics already embedded in the information systems.*
>
> **Proposition (RP1-R3):** *The effectiveness of IT growth options is positively affected by the SOA flexibility-centric and structure-centric characteristics already embedded in the information systems.*

The structure-centric and flexibility-centric characteristics improved the process option effectiveness. However, they also introduced options depreciators, which had negative impact on the option effectiveness. Options depreciators captured additional overheads created in the processes such as change management and testing due to dependencies between services or latency when the services are distributed across network. A balanced selection of characteristics to manage the dependency between services and distribution of services can control and reduce the options depreciators and consequently increase the effectiveness of the options. Similarly, the following proposition summarises the above finding:

> **Proposition (RP1):** *The effectiveness of the IT process options is negatively affected by the options depreciators that SOA flexibility-centric and structure-centric characteristics create.*

Regarding the impact of process options on the IT responding capability, the researcher's findings were consistent with the agility literature (1994; Dove, 2001; Malone & Crowston, 1994; Mohr & Nevin, 1990), and showed that the SOA-enabled process options provide spare capability to the IT to be used when required. The available capabilities reduce the development or operational management effort cycles, which in turn will improve the time to market and the IT responsiveness. The options also provide the IT management or IT development teams with flexibility to select the best option that suits the emerging requirements. Therefore, the following proposition is proposed:

> **Proposition (RP1-P2):** *The IT responding capability is positively impacted by the effective process-based options that SOA offers.*

Additionally, the role of shared insight in improving the effectiveness of options was clearly observed in the data analysis. The shared insight through knowledge sharing within IT and with the business and shared visibility on capabilities increased the range and richness of process options. For example, the business did consider alterative sourcing options to meet their time to market and refined their requirements to meet the existing IT capabilities. The above finding is reflected in the following proposition:

> **New Proposition (RP4-R4):** *The SOA-enabled 'Shared Insight' positively affects the effectiveness of the process options.*

To achieve agility, the analysis also highlighted that the type of governance depends on the service layering, with the lower layers requiring more centralised governance and the higher layers requiring a decentralised governance model. The centralised governance at the lower layers provided higher level of reusability for the core services and closer alignment to the change needs due to the services being more generalised and less specific to particular use cases. The analysis also showed that the centralised governance at this layer reduced the time and the redevelopment cycles required for the service to evolve and achieve its fitness with the environment (option richness).

The findings concerning the adaptive governance are different from those in the SOA governance literature (Bieberstein et al., 2005; J. H. Lee et al., 2010; Schepers et al., 2008; Walker, 2007) which suggests a dedicated centralised governance body for the SOA. A more recent study (Joachim et al., 2013) of SOA governance also raised questions concerning a dedicated SOA decision-making body separate from the IT governance. The difference could be due to the fact that the researcher's study focused on the agility and decomposed the SOA based on its characteristics, particularly the hierarchical layering.

The researcher's findings revealed that the lower layers facilitate the options that are associated with the core enterprise systems and IT infrastructure. In this case, the IT, due to its knowledge of the enterprise-wide IT portfolio and deep technical knowledge, is in a better position to centrally identify and execute the options in the lower layers.

In the higher layers, however, the decision-making for the options require knowledge of market needs and the consumer channels. Therefore, the specialisation required for decision-making shifted from the IT to the business units, which require the business units to have managerial flexibility to exercise the options that best meet their requirements.

On the IT agility literature, the researcher's findings can explain Tiwana and Konsynski's work (2010), which showed that a decentralised governance has both positive and negative impacts on the IT agility. Based on the findings and from the options perspective, the decentralised governance provides the business line functions with autonomy to identify and execute suitable options that best achieve their goals. The negative impact observed in the Tiwana and Konsynski's work (2010), however, could be due to the core reusable process options not fully embedded due to departmentalised decision-making. According to the findings, a centralised governance structure in the reusable assets provides the mean to foster such core and fundamental capabilities in the organisation, which are typically across multiple departments and require technical knowledge and sponsorships, which are cross-departmental.

The role of adaptive governance is reflected in the following proposition:

> **New Proposition (RP4-R5):** *The adaptive governance moderates the impact of SOA-enabled process options on the IT responding capabilities.*

Finally, the researcher's study took a few steps to shed light on the options theory. While majority of the previous focus were on isolated options, the researcher found that options when coexist do not operate in isolation and can have implications on each other. On this basis, this effect was conceptualised as options depreciators. In this context, the findings showed that dependency between options and the latency created from distributed services create options depreciators and reduce the effectiveness of the options. While the measures of options depreciators are expected to be context specific, the concept of options depreciators could be generalizable to other applications of Options theory.

On the type of options, the findings extended the classification of options and identified new knowledge and process options that can be available to IT. 'Change Detection' and 'Shared Insight' were the two knowledge options which organisations can gain, especially when business and IT have continuous and close engagement. On the process options, options such as gradual transition, change service provider, wrap and simulate behaviour; extension by service reuse, design reuse and extension by connectivity are examples of options available to IT managers.

The other insight from the study is the interaction between the knowledge and process options. The analysis showed that knowledge options, particularly a shared insight, is a necessary condition for effective process options. Not only the shared insight provides the business with awareness of the available capabilities, it also provides insight to identify alternative options that can be beneficial. The business can also align their requirements to the IT capabilities, if possible, to reduce the time to market.

Finally, while organisations can make investments and embed options, the data analysis showed that the options might not be considered and executed when required. Not using the available options not only hinders the business from gaining the options potential benefits, but it wastes resources used to embed the initial options. The analysis showed introduction of adaptive governance is a prerequisite condition to enforce the option execution. The adaptive governance fosters the real option and reduce the risks of shadow options, as options which are not recognised (Bowman & Hurry, 1993).

Finally, the embedment of options benefits organisations when there is uncertainty and potential for future use. When the options are not effective, not only they do not contribute to the agility, but also, they can hinder the agility. By identifying SOA characteristics the proposed model has taken steps to identify the criteria required for embedment of effective options. While the researcher's model does not prescribe the decision criteria for option embedment, it shows the conditions where embedded options can be effective.

6.3 Implication for Theory and Practice

The current study offers multiple contributions to both theory and practice.

On the implications for theory, the study addresses the gap identified in the SOA literature concerning how SOA must be conceptualised in this study. Through the review of IT literature, SOA literature, and empirical data analysis, the study identified and reported a set of refined SOA characteristics. To reduce the duplication between these characteristics, the study offered a definition for each characteristic either sourced from the extant literature, or extracted from the data analysis. Additionally, the categorisation of the SOA by three themes is another novel contribution of this study, which in turn facilitated the opportunity to study the impact of SOA on the IT agility. These categories provided a level of abstraction required for theorisation of SOA business value and identification of how SOA creates IT agility. Future studies can adopt the proposed approach to develop new theories to explain SOA interaction with its surroundings and its business values.

The second theoretical contribution of this study is to the SOA and IT governance literature. The finding concerning the adaptive governance structure based on layering of the systems is a novel contribution to the SOA and IT literatures. Currently, there is a large number of studies in the SOA literature that suggest a centralised and separate governance body for the SOA (Joachim et al., 2013). Conversely, the IT literature has a mixed view on the issue (Almeida, Pereira, & da Silva, 2013). By focusing on the governance structure for an agile IT, and by decomposing systems to different service layers, this study extends the existing IT literature by proposing an adaptive structure that aligns to the software layers structured based on the rate of change and their level of reusability. This viewpoint can offer new ways of looking at governance, particularly to foster business autonomy and creation of sustainable assets for future use.

The third theoretical contribution of this study is to the broader IT literature. The IT literature has historically treated a system as a monolithic black box (Orlikowski & Iacono, 2001) by limiting its focus on IT infrastructure or broader IT unit capabilities. This study unpacks the black box of systems to their underlying characteristics of both infrastructure and service design characteristics; and provide insight into the characteristics that contribute to the IT agility and their intermediate organisational impacts. Such viewpoint becomes more important as concepts such as SOA makes the boundaries between the black box of applications and infrastructure blurry. While the SOA literature has taken some steps towards shedding light on the decomposition of applications to smaller services and their behaviour, perhaps lack of theorisation in the SOA literature has left gaps between the mainstream IT literature and the SOA literature. This study takes the preliminary steps to bridge this gap by reliance on the IT literature for conceptualisation of SOA, and proposing a mid-range theory to explain how a service-oriented system can create business values.

The forth contribution of this study is to the IT agility literature. The current study is the first study that focuses on the SOA characteristics and their impact on the IT sensing and responding capabilities through the Options theory. The current study complements a recent study by (Richter & Basten, 2016) who similarly raised the issue of the existence of relationship among SOA characteristics and the IT sensing and responding capabilities. The authors of the above study, however, acknowledged that the current state of knowledge suffers from lack of theoretical grounding to conceptualise and explain the relationships between SOA and agility. The current study contributes to the existing knowledge by conceptualising the SOA and explaining the relationship between SOA and IT agility using the options theory.

By identifying the SOA characteristics that contribute to the SOA enabled knowledge options and consequently to the IT sensing capability, this study extends the existing knowledge on how service orientation improves the sensing capability. The current study breaks down the service orientation construct and identifies broader role of shared insight and change detection options in improving the sensing capability. Findings also emphasise the complementary roles of a number of factors that moderate the impact of SOA on IT agility. For example, the moderating effect of *continuous business and IT engagement*, which moderates the effect of SOA information repository characteristics on the shared insight gained from the SOA.

Similarly, identification of SOA characteristics that contribute to the IT responding capabilities provides some answers to the existing conflicting reports concerning the complexity of SOA. By decomposing the SOA into its characteristics, this study untangled the complexity paradox of SOA and highlighted the factors that are important in creation of future options for the SOA and their impact on creation of option depreciators such as dependency and latency. Also, the complementary role of shared insight in the process options was another observation made in this study.

Regarding the SOA characteristics, this study stayed away from specific technology or standards, and focussed on the underlying characteristics; and attempted to offer a more generalizable result.

In the context of broader IT agility literature (Lowry & Wilson, 2016; Yousif, Magnusson, & Pessi, 2017; Yousif & Pessi, 2016), the current study contributes to the IT agility literature by identifying the role of knowledge in improving the IT agility. The current study revealed the direct and indirect effects of knowledge options on the IT agility.

Finally, the fifth theoretical contribution of this study is to the options theory literature. The current study is based on the notion by Sambamurthy et al. (2003) who conceptualise IT as a digital options generator. The current study extends the understanding of the real options theory by taking additional steps to operationalise the conceptual framework suggested by Sambamurthy et al. (2003). Use of option effectiveness and its two measures of option range and option richness expand how options theory can be used as a theoretical framework to explain future phenomena.

By identifying specific process and knowledge options applicable to the SOA and IT agility context, the current study took additional steps in the application of option thinking (Fichman, 2004; Fichman et al., 2005) and digital options (Sambamurthy et al., 2003) in the study of IT business value.

Furthermore, the current study extends the options theory when competing and interdependent options are available. While the extant literature reviews the IT option (Fichman et al., 2005; Tiwana et al., 2006) and its value assessment (Gamba & Fusari, 2009; R. L. Kumar, 2004) individually, the results of the current study found that options do not operate in isolation and can have implications on the values that each and every one of them may generate. The researcher conceptualised this effect as options depreciators. The current study also contributes to the knowledge by revealing the interplay between options and several contingent factors such as continuous business and IT engagement. Additionally, the shared knowledge option improves the effectiveness of the process options. While options can be embedded, their execution is not guaranteed. It is the adaptive governance that provides agility to the business in executing the options while maintaining the control to embed core options for future use.

On the contribution to the practice, this study offers insight into the specific SOA characteristics that the practitioners must focus on in realising IT agility. Proper service monitoring with alerting enabled by the SOA infrastructure offers the option for the IT team to detect issues quickly, before the business or end users report them. The information repository characteristics also proves to be essential in creating a shared insight. While the services must be business-oriented, the service definitions and standards such as the design patterns, processes and selected technologies must be documented in a business-oriented language. Such documentations, when are discoverable to the business and IT, can improve the IT and the business knowledge sharing. The knowledge sharing and share insight however require the business and IT to have continuous engagements to share perspective and discuss the capabilities that are available and the options that must be executed to address the system change requirements.

In addition to the information-centric characteristics, the practitioners must focus on the SOA characteristics that deliver flexibility. Characteristics such as adaptability and connectivity offered in the SOA platform, loosely coupling at the service run-time, deployment between services and interfaces at the design time, standardisation and framework-driven provide flexibility to make changes in the system with less effort.

The structure-centric characteristics group is the other group of SOA characteristics that provide insight on the design of services individually and together. While there has been a long debate on the granularity of services whether they should be designed fine-grained or coarse-grained, findings of the study propose a way to determine the right granularity for the service. As per the researcher's findings, while a finer granularity is desirable in achieving reusability, the granularity must be considered with additional factors such as level of generalizability of the service, the relatedness of the service operations, its independence from other services and the overall loosely coupling of the service. Combination of these factors can increase the options that the service can offer for future use with reduced dependency between the services. Additionally, to increase the IT agility, the services must be grouped in layers based on their rate of reusability and how close they are to the core systems or data repositories, especially when there are additional data security and auditing requirements for the core systems. Such layering, when the layers are physically separate, combined with an adaptive governance structure provide control over the IT on the capabilities which are core and flexibility to the business to meet its time to market.

6.4 Limitations of the study

This study inherits a few limitations as explained below. On the research methodology, case study research has been criticised due to the high level of subjectivity and concerns about generalisability (A. S. Lee, 1989; Yin, 1994). Scholars (Dubé & Paré, 2003; Eisenhardt, 1989; Patton, 1999; Yin, 1994) have suggested that validity and reliability in case study research involves using clearly defined methodological guidelines for ensuring construct validity, internal validity, reliability and external validity. As outlined in Table 4-9, this study applied several methods to improve the rigour of the current case study research. Regarding the generalizability, this study selected its case studies from different industries and the embedded cases included a mix of different projects from different business units to broaden its applicability.

Regarding the limitations of the findings, three limitations are notable. The first limitation is concerned with the tool view of technology (Orlikowski & Iacono, 2006) adopted in this study. Tool view usually conceives technology independent of the social or organizational arrangements within which it is developed and used (Orlikowski & Iacono, 2006). While this view avoided creation of overly complex theory that could be expected from theories developed from cases (Eisenhardt, 1989), it limited the insight on the non-technical factors that contribute to the IT agility. To reduce this issue, the researcher extended the data collection and analysis to monitor and report the vital non-technical factors that impacted the interaction between the technology and the IT agility.

The second limitation of the results relates to the process of embedding options vs. executing the options. This study focused on the option execution and assumed that options are already embedded. Such assumption was incorporated in the model by considering the effectiveness of options. An option which has not been embedded yet has low level of effectiveness, if any, due to the time required to build the option. This study however did not consider factors involved in the decision to embed the option.

The third limitation of this study relates to the type and rate of change. While the selected cases and their embedded cases presented different rate and type of change in the projects and organisation, the nature of study did not provide an opportunity to study the rate and type of change.

6.5 Future Research

The current study provides opportunities for future research in several areas, as follows:

In the Service Orientation field, results from this study create opportunities to refine the SOA design processes, especially when the IT agility is the objective of the SOA implementation. The design processes are usually silent on the importance of characteristics required in the system to achieve a particular objective. Such refinements to the design process could focus on the important characteristics leading to better adoption of the SOA in the organisations and offer richer empirical research on how SOA assists organisations to achieve their business goals. The results of the current study in form of its identified SOA characteristics and their consequent impact facilitates the follow-on work to refine the SOA design processes.

Future studies are required to test the hypothesis of the conceptual model offered by this study and its propositions. Validation of the impact of SOA information-centric characteristics on the IT sensing capability can be the subject of a separate study to confirm the proposed relationships and their applicability in a wider and broader context. Similarly, the impact of SOA on the responding capability can be validated in another study. These studies become more insightful if the scope of the SOA is extended to Cloud computing adoptions (Hoberg et al., 2012; Stieninger & Nedbal, 2014) and new emerging concepts of Microservices (Balalaie et al., 2016). The future studies can also investigate the relationships among the identified SOA concepts particularly during the construct development.

Future studies could also investigate the relationship between the adaptive governance and the service layering. Further studies can focus on the alignment between the governance structure and the layering in the services and if such structure leads to a different rate of change in each of the layers over time. The researcher's study presented interesting findings in this area, which can be expanded further in future studies. The future studies can focus on the governance of the IT artefacts as well as its contents. Additionally, the future studies can breakdown the complex construct of governance to dimensions of 'decision rights', 'control' and 'standards and policies provisioning', as suggested by Tiwana et al. (2013) to study the appropriate governance structure for each dimension. The future studies can develop new hypothesis concerning the dimensions of governance and their assignment to different layers of architecture, particularly in the

SOA context.

A promising line of future research is to conduct a process theory (Avgerou, 2013; Pentland, 1999) study on the impact of SOA on the IT agility from the options theory perspective. The researcher's findings showed that options could be refined over time in an iterative evolutionary process. The first time an option is embedded, it might not be generic enough to cater for certain likely scenarios in future. Therefore, when a use case scenario occurs and the option is assessed for execution and use, it has to be refined or skipped all together. Such refinement and extension to the service is undesirable considering its implications on the existing services and the project timeline. The less the number of refinements are and options can be adopted without change, the more responsive the IT will be to the business needs. The data analysis showed that processes such as proactive planning and competencies such as IT holistic thinking can assist to shorten the number of iterations. For instance, the airline adopted a top-down planning process based on the portfolio of the projects to assess the services required in these projects. After identifying the services, the service development went through a separate and parallel development cycle separate to the product development process.

Figure 6-3 presents information collected during the data collection in the form of events and actions which could be usable in proposing a process theory (Pentland, 1999). While the presented model is not complete and the result of such study can look different from the depicted model, it can show the opportunity for future research. Due to the process theory presenting the actions and their antecedent events over time (Avgerou, 2013), it can provide further opportunities to explain the process of embedding an option and its execution. This is particularly important for the decision to embed an option. While an embedded option can offer value in future and provide agility, the process of embedding an option, for instance a reusable service, can cost the organisation more and itself slow down the delivery. Therefore, the condition to make such investment require further attention.

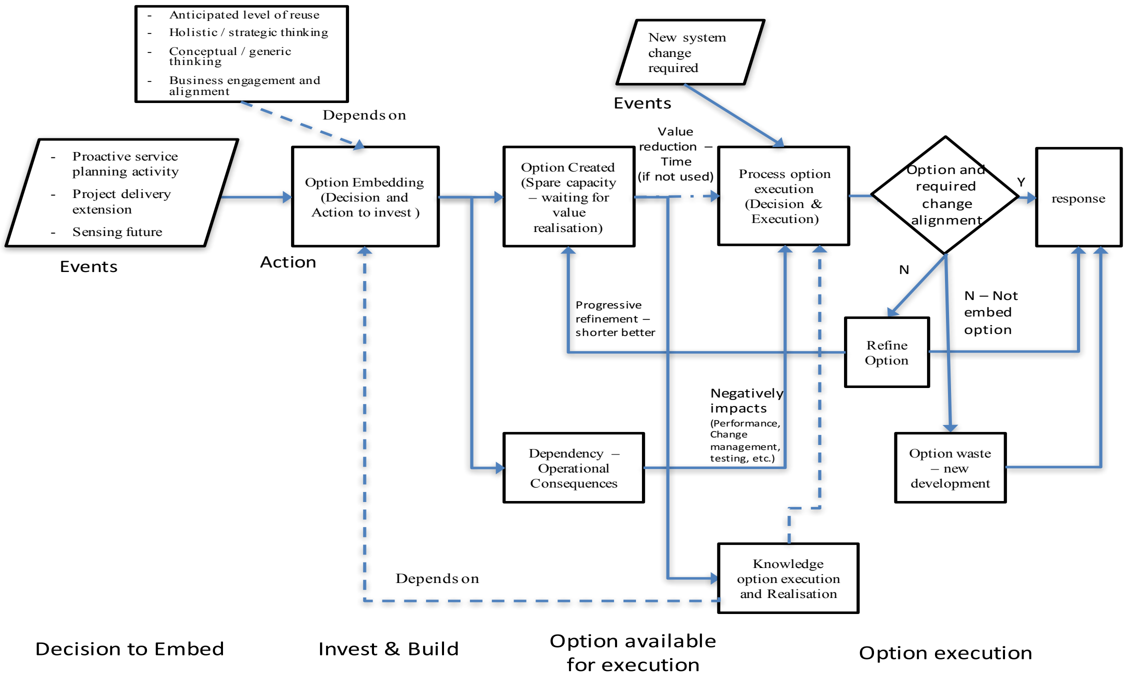


Figure 6-3 –Future research on the option refinement process

6.6 Conclusion

The current thesis aimed to develop a deep and rich understanding of how SOA affects IT Agility. Decomposition of SOA to its core characteristics and use of Options theory lens provided a different perspective to explain the SOA value generation mechanism. Not only this study identified the SOA characteristics required to achieve IT agility, but also identified a number of complementary factors important to achieve IT agility.

The conceptual model emerging from this study shows how embedded SOA characteristics affect the knowledge and process option effectiveness, which in turn affect the IT sensing and responding capabilities. The model also depicts the options depreciators which present the interplay between options and their negative impact on one another.

In doing so, this work has provided several significant theoretical contributions to the existing body of knowledge on SOA implementation and governance, IT agility and the Options theory which was the base theory applied in the current study. Conceptualisation of SOA based on its characteristics, identification of antecedents for the IT sensing and responding capabilities, and extension of options theory are the highlights of the current research.

Finally, the adopted theory building research through case studies provided me with very rich understanding of the researcher's initial research enquiry as well as many new areas for future exploration and research.

Appendices

Appendix A – Literature Review on Conceptualisation of IT Characteristics

Constructs/Measures	Definition or Descriptions	Source/IT Literature References
Scalable	Allows firms to integrate disparate and geographically distributed systems.	IT Infrastructure / (G. D. Bhatt et al., 2010)
Compatible		IT Infrastructure / (G. D. Bhatt et al., 2010)
Share information*	Yet to develop and maintain: this requires managers to adopt and implement a shared set of IT standards and policies. When standards and policies are in place, firms can share information across internal business units and external partners.	IT Infrastructure / (G. D. Bhatt et al., 2010)
Modular	Allows firms to integrate disparate and geographically distributed systems.	IT Infrastructure / (G. D. Bhatt et al., 2010)
Handle multiple business applications		IT Infrastructure / (G. D. Bhatt et al., 2010)
Business knowledge/skills	The ability of human IT infrastructure (ITI) resources to enable flexibility is reflected in the depth and breadth of the knowledge and skills of the IT personnel. Therefore, we considered the three domains of IT personnel knowledge and skills as the flexibility-enabling dimensions of human ITI. Possessing the knowledge and skills puts an organisation in a position to take advantage of new opportunities.	IT Infrastructure / IT Personnel Knowledge and skills (Fink & Neumann, 2009)
Behaviour knowledge/ skills	The ability of human ITI resources to enable flexibility is reflected in the depth and breadth of the knowledge and skills of the IT personnel. Therefore, we considered the three domains of IT personnel knowledge and skills as the flexibility-enabling dimensions of human ITI. Possessing the knowledge and skills puts an organisation in a position to take advantage of new opportunities.	IT Infrastructure / IT Personnel Knowledge and skills (Fink & Neumann, 2009)

Constructs/Measures	Definition or Descriptions	Source/IT Literature References
Technical knowledge/skills	The ability of human ITI resources to enable flexibility is reflected in the depth and breadth of the knowledge and skills of the IT personnel. Therefore, we considered the three domains of IT personnel knowledge and skills as the flexibility-enabling dimensions of human ITI. Possessing the knowledge and skills puts an organisation in a position to take advantage of new opportunities.	IT Infrastructure / IT Personnel Knowledge and skills (Fink & Neumann, 2009)
IT modularity	Reusability can be achieved by implementing independent and standardised components, implying modularity. Loosely coupled components allow greater flexibility in end configurations. It has been well-established that shared technical components enhance flexibility when they are connectable, compatible and modular	IT Infrastructure / (Fink & Neumann, 2009)
IT compatibility	Sharability involves connectivity and compatibility. For the technical ITI element to support multiple business processes and applications, the technical components should be seamlessly deployed across the organisation, allowing users to share information.	IT Infrastructure / (Fink & Neumann, 2009)
IT connectivity	Sharability involves connectivity and compatibility. For the technical ITI element to support multiple business processes and applications, the technical components should be seamlessly deployed across the organisation, allowing users to share information.	IT Infrastructure / (Fink & Neumann, 2009)
Connectivity	Ability of any technology component to attach to any of the other components inside and outside the organisational environment.	IT Infrastructure /(Duncan, 1995)
Compatibility	Ability to share any type of information across any technology component. At one extreme, only simple text messages can be shared, while at the other extreme any document, process, service, video, image, text, audio or a combination of these can be used by any other system, regardless of	IT Infrastructure /(Duncan, 1995)

Constructs/Measures	Definition or Descriptions	Source/IT Literature References
	manufacturer, make or type.	
Modularity	Ability to add, modify and remove any software, hardware or data components of the infrastructure with ease and with no major overall effect. Modularity also relates to the degree to which IT software, hardware, and data can be either seamlessly or effortlessly diffused into the infrastructure or easily supported by the infrastructure.	IT Infrastructure /(Duncan, 1995)
IT infrastructure capability	The IT infrastructure capability is operationalised by measuring four indicators: data management services and architectures, network communication services, application portfolio and services, and IT facilities' operations/services.	IT Capability / (Lu & Ramamurthy, 2011)
IT business-spanning capability	IT business spanning capability addressing the business-IT strategic thinking and partnership.	IT Capability / (Lu & Ramamurthy, 2011)
IT proactive stance	IT proactive stance covering opportunity orientation.	IT Capability / (Lu & Ramamurthy, 2011)
Technical IT capabilities	Operationalised as a second-order construct with hardware compatibility, software modularity, network connectivity and IT skills adaptability as the first-order constructs. Study: how IT capabilities and IT managerial capabilities improve agility in different *environmental volatility*.	(Tallon, 2008)
IT flexibility (second-order)	Second-order construct, consisting of: "integration," "modularity" and "IT personnel flexibility"	IT Infrastructure / (Byrd & Turner, 2000)
Integration	Transparent access into all organisational platforms and consistent IT connectivity (ability of any technology component to attach to any of the other components inside and outside the organisational environment) and IT compatibility (the ability to share any type of information across any technology component)	IT Infrastructure Flexibility / (Byrd & Turner, 2000)

Constructs/Measures	Definition or Descriptions	Source/IT Literature References
Modularity	Ability to add, modify and remove any software, hardware or data components of the infrastructure with ease and with no major overall effect. It consists of application functionality and data transparency (free retrieval and flow of data between authorized personnel in an organisation or between organisations regardless of location) for hardware, software and data	IT Infrastructure Flexibility / (Byrd & Turner, 2000)
IT personnel flexibility	Covering technology management, business knowledge, management knowledge and technical knowledge (Byrd & Turner, 2000) Technical skills, boundary skills, functional skills, technology management (Byrd & Turner, 2001)	IT Infrastructure Flexibility / (Byrd & Turner, 2000; Byrd & Turner, 2001)
Connectivity	Ability of any technology to attach to any of the other technology components inside and outside the organisational environment.	IT Infrastructure / Integration / (Byrd & Turner, 2000)
Compatibility	Ability to share any type of information across any technology components.	IT Infrastructure / Integration / (Byrd & Turner, 2000)
Data transparency*	Ability defined as the free retrieval and flow of data between authorized personnel in an organisation or between organisations regardless of location.	IT Infrastructure / Modularity / (Byrd & Turner, 2000)
Application functionality *	Ability to add, modify and remove any software applications of the infrastructure with ease and with no major overall effect.	IT Infrastructure / Modularity / (Byrd & Turner, 2000)
Technical skills	A set of measures of technical capabilities such as programming, understanding software development processes and knowledge of operating systems.	IT Infrastructure / IT Personnel Flexibility / (Byrd & Turner, 2001)
Boundary skills	Refers to the importance of IT personnel having skills and knowledge to assume roles outside their area of training or original competencies. This may include areas like project management and business process support.	IT Infrastructure / IT Personnel Flexibility / (Byrd & Turner, 2001)
Functional skills	Ability of IT personnel to understand the business processes they are to support and apply the	IT Infrastructure / IT Personnel Flexibility /

Constructs/Measures	Definition or Descriptions	Source/IT Literature References
	appropriate technical solution to a given business problem.	(Byrd & Turner, 2001)
Technology management	Pertains to the organisation's ability to deploy IT in the most effective possible manner in support of the business strategies	IT Infrastructure / IT Personnel Flexibility / (Byrd & Turner, 2001)
Loosely coupling	The degree to which changes within a subsystem do not create a ripple effect in the behaviour of other parts of the ecosystem. Dimensions: standardisation and loosely coupled.	Loosely Coupling / (Nambisan, 2002)
Modularity	the degree of decomposition of an organization's IT portfolio into loosely coupled *functionality discrete* subsystems that communicate through standardized interfaces	IT Modularity / (Nambisan, 2002)
Flexibility in functionality	Consists of robustness, scalability and slack	IT Dimensions of Flexibility / (Knoll & Jarvenpaa, 1994)
Flexibility in modification	Commensurability, feedback sensitive, goal adjusting, just-in-time adjusting, polyadjustable, self-adjusting, trialability	IT Dimensions of Flexibility / (Knoll & Jarvenpaa, 1994)
Flexibility in use	Concurrency, connectivity, modularity, multiple forms, responsiveness, reusability, spatial decoupling, temporal decoupling, transparency, versatility	IT Dimensions of Flexibility / (Knoll & Jarvenpaa, 1994)
Decomposability	How the form and function of a platform's ecosystem are broken down into constituent atomic subsystem.	Nearly decomposable system (Simon, 1962)

Appendix B – A Priori Constructs and Mapping to SOA Characteristics

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table** 2-5)	**Gap requires further construct development or extension**
Modularity SOA services characteristic	Modularity is the degree of decomposition of an organization's IT portfolio into loosely coupled subsystems that communicate through standardized interfaces (Nambisan, 2002)	N/A. Second-order formative construct. Dimensions: standardisation and loosely coupled (Tiwana et al., 2010).	'Modular'	Requires emphasis on 'discrete function', possibly in a separate construct. Also decomposability needs to be considered.
Loosely coupled SOA services characteristic	The degree to which the applications in an organization's IT architecture are designed such that internal changes in one application do not affect the behaviour of others (Nambisan, 2002)	SOA literature (Joachim, Beimborn, Schlosser, et al., 2011) adopted from (Chung et al. 2005; Tallon 2008) – Modularity: - We can add new functionality to our systems without having serious problems. - Exchanging or modifying single components does not affect our IT infrastructure. - Our systems consist of clearly separated modules.	'Loosely coupled' 'Abstract from service implementation / implementation independence'	

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table 2-5**)	**Gap requires further construct development or extension**
Standardisation SOA services characteristic	The degree to which organisation-wide standards (technology, protocols, interface, semantics for data and business tasks) and policies pre-specify how applications in an organisation's IT portfolio connect and interoperate with each other (Ross & Weill, 2005).	(1) IT standards, (2) IT policies, (3) IT architecture, (4) compliance guidelines for IT applications, (5) compliance guidelines for IT infrastructure (Tiwana et al., 2010). IT technical standards are used XML, WSDL, SOAP (Oh et al., 2007).	'Technical standardisation' 'Stable, managed service contracts', 'Comprehensive, uniform service specification', 'business standardisation', 'Interoperability using open, widely applied industry standards'	'Comprehensive, uniform service specification' and 'business standardisation' would require further indicators to reflect standardisation of data formats and protocols across all services.
Decomposability SOA services characteristic	The degree a system can be broken down into constituent granular subsystem (finer grain functions), each providing discrete function (Simon, 1962).	Possible: - (3) well-understood interdependencies, - - (4) minimal unnecessary interdependencies (Tiwana et al., 2010). The degree the operations of the service are	'high service cohesion', 'Service granularity oriented toward business concepts' and	'Service granularity' requires an indicator to cover the 'district business function'.

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table 2-5**)	**Gap requires further construct development or extension**
		related.	'autonomous'	
Integration SOA platform characteristic	Byrd (2000) considers integration as transparent access into all organisational platforms.	N/A. Second-order formative construct. Dimensions: IT compatibility and IT connectivity (G. D. Bhatt et al., 2010; Byrd & Turner, 2000; Duncan, 1995; Fink & Neumann, 2009).	SOA Infrastructure (Papazoglou, 2008)	
IT compatibility SOA platform characteristic	Ability to share any type of information across any technology component (Byrd & Turner, 2000).	- Systems and services utilise common data definitions in their data exchange. - Services (data) can be easily used across multiple platforms - Our organisation offers a wide variety of types of information to end-users (e.g. multimedia) - Our SOA platform provides transparent access to all services - Data received by our organisation from electronic links (e.g. EDI, EFT) are easily interpretable - The rapidity of IT change (e.g. revision level, release) to SOA systems in our organisation is high - Information is shared seamlessly across our organisation, regardless of the location - Our organisation provides multiple interfaces or entry points (e.g. web access)	No specific construct, however characteristics are included in connectivity (Papazoglou, 2008)	Amended the indicators to cater for SOA infrastructure. During construct development, might need to break them down to multiple constructs or refine the indicators.

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table 2-5**)	**Gap requires further construct development or extension**
		for consumers (Byrd & Turner, 2000; Papazoglou, 2008)		
IT connectivity SOA platform characteristic	The degree of seamless attachment of any technology to any of the other technology components inside and outside the organisational environment (Byrd & Turner, 2000; Joachim, Beimborn, & Weitzel, 2011).	- Dynamic configuration and binding - The SOA platform provides data and protocol adaptation to facilitate connectivity between service provider and consumers internally or externally. - Our organisation utilises open systems network mechanisms to boost connectivity (e.g. HTTP) - The SOA platform facilitates message exchange between service providers and consumers, - We can integrate additional data formats (e. g. EDI, XML) easily in our applications. - Flexible electronic links exist between our organisation and external entities - Our organisation has formally addressed the issue of data security with access to a number of protocols (e.g., HTTPS, WS-Security) - New locations or acquisitions are quickly assimilated into our SOA infrastructure	Connectivity (Papazoglou, 2008)	Amended the indicators to cater for SOA infrastructure. During construct development, might need to break them down to multiple constructs. Some of the indicators for the compatibility and connectivity are related to standardisation or others. There should be a link from standardisation to them.
IT agility	IT agility deals with how swiftly IT can detect and implement a change to the information systems to support business to survive and thrive in an uncertain environment.	Measurement through reflective measures for IT sensing and responding capabilities by adopting (Roberts & Grover, 2012) measures for this study.	N/A	N/A

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table 2-5**)	**Gap requires further construct development or extension**
Service Discovery	The degree services are managed to become discoverable.	- The service consumers can easily locate a service for information consumption (e.g. through service repository). -	Service Discovery (Papazoglou, 2008)	
Composability	The degree the services are composed together to deliver a business functionality and form a business processes.	-	Composibility (Papazoglou, 2008)	
Monitoring	Monitoring of internal behaviour and state of services to management systems (services). Also monitoring of the service level agreements for a business process.	-	Monitoring (Papazoglou, 2008)	
Modelling	The extent the service is defined including their interface (e.g. WSDL), the internal behaviour specially in business processes (e.g. BPMN)	-	Modelling (Papazoglou, 2008)	
It Governance Decentralisation	IT governance decentralization (centralization) refers to the degree to which the line functions have greater decision-making authority for IT	IT specification decision rights are defined as decision-making authority for specifying what objectives IT should accomplish, and IT implementation decision rights specify how it		

Constructs/Grouping (Service, Platform) characteristic	**Definition** (Sourced from literature)	**Proposed Indicators** (Sourced from IT literature)	**Relevant SOA Characteristics** (Sourced from section 2.3.6 – **Table** 2-5)	**Gap requires further construct development or extension**
	decisions (Brown 1997). Fama and Jensen (1983) identify two broad classes of decision rights: (1) those that define what objectives a department should accomplish and (2) those that define how it should accomplish them.	should accomplish those objectives. IT specification therefore encompasses decisions about what business processes in the line functions IT must support, the associated constraints (schedule, bud- get, quality), objectives, priorities, and performance expectations (e.g., service levels). IT implementation encompasses decisions about the methods, programming languages, platforms, definition of IT standards and policies, and IT sourcing (e.g., outsourcing, purchase, or internal development).		

Appendix C - Emerged SOA characteristics affecting IT Agility

Code	Description
Information-centric	SOA characteristics that participate in capture, distribution and discovery of information related to services.
Information Dissemination	SOA Characteristics that participant in distributing information suitable to sense a change in an information system.
Information Repository	SOA characteristics that contribute in capturing information related to the services.
Information Discovery	SOA characteristics that contributes in discovering the services and their definitions.
Flexibility-centric	SOA Characteristics that facilitate operational management and maintenance of the SOA systems through flexibility in adopting a change.
Structure-centric	SOA characteristics that describe the design and structure of services
Activity-centric	SOA characteristics that facilitate the effort and tasks required in development and commissioning of a service.

SOA Characteristic	Category (Parent Node)	Characteristics Type (SOA Design, Platform)	Definition	Analysis
Business Oriented	Information-centric / Information Repository	SOA Design	The extent the function and definition of service is aligned with business concepts and meanings	Consists of alignment of the service definition and function to Business services, e.g Business capabilities in the case of the Airline. This can be: - Business process

SOA Characteristic	Category (Parent Node)	Characteristics Type (SOA Design, Platform)	Definition	Analysis
				- Business data
Standardisation	Information-centric / Information Repository	SOA Design and Platform	The extent consistent rules, guidelines and specifications in design and use of services are formulated and implemented.	Different dimensions of standardisations have been identified as: (1) Consistent IT Technology standards, e.g. XML, WSDL, SOAP, etc. in development and system implementation (2) Consistent IT policies applied to the service design and management, (3) Consistent IT architecture, (4) Consistent Design patterns - pattern oriented as consistent approach in development or management of services to solve a particular problem. This was captured in form of established framework and libraries, many documented and known in form of team knowledge. (5) Consistent service development and management processes to develop and manage services, either explicitly documented or team knowledge. (6) Consistent data semantics and interface definitions as per the internal or external standards, such as AIDX for the aviation industry or Swift data exchange standards for the bank. This addresses existing definitions from (Ross & Weill, 2005), (Tiwana et al., 2010) or (Oh et al., 2007).
Service Definition	Information-centric / Information Repository	SOA Design	The extent information regarding the service interface and the functionality of the service are captured and published	Involved description of: - Service interface - Expected functionality of the service
Service Discovery	Information-centric / Information Discovery	SOA Platform and Design	The degree services are managed to become discoverable, involving classification and publication of services in a way they can be found	Different cases used different technologies. They however shared these aspects: - Search functionality

SOA Characteristic	**Category (Parent Node)**	**Characteristics Type (SOA Design, Platform)**	**Definition**	**Analysis**
			easily when needed.	- Classification of services as per a suitable taxonomy both for data and higher level services such as business process services. - Being in English and easy to be consumed by IT and Business
Service Monitoring	Information-centric / Information Dissemination	SOA Platform and Design	Level of exposure and availability of access to behaviour of services and their quality measures in a usable form for IT and business.	Monitoring of internal behaviour and state of services. It covered service's health and usages, from simple services to an end to end process implemented through service composition.
Event Driven	Information-centric / Information Dissemination	SOA Platform and Design	The extent changes detected in data or system states are exposed and notified to interested parties in a timely manner.	The analysis shows the 'Event Driven' characteristic can improve the business and IT sensing capability. This is through the real-time access to information and events, when they occur.
Granularity	Structure-centric	SOA Design	The degree of functionality embedded in a service reflective of the number of tasks it handles, the amount of data it processes and returns and the number of external interactions it has.	The data analysis revealed that the following factors must be considered to choose the correct granularity of services: - Level of service reuse - Security of data - Performance - Service dependency and consequently complexity. - The expected level of customisation to the functionality in future
Generalisability	Structure-centric	SOA Design	Ability of a service to address a number of use cases rather than a specific use case.	
Service Cohesion	Structure-centric	SOA Design	The degree the operations of a service	

SOA Characteristic	Category (Parent Node)	Characteristics Type (SOA Design, Platform)	Definition	Analysis
			having related functionality.	
Autonomy	Structure-centric	SOA Design	The degree the logic governed by a service resides within an explicit boundary. The service has complete autonomy within this boundary and is not dependent on other services for the execution of this governance (Erl, 2005; L. O'Brien et al., 2007).	Modular deployable is also considered part of Autonomy.
Loosely Coupled	Structure-centric Flexibility-centric	SOA Design	The degree to which changes within a subsystem do not create a ripple effect in the behaviour of other parts of the ecosystem (Tiwana et al., 2010).	The loosely coupling applied to: - The service interface to protect changes in one service affecting the rest of the services. - Hide Implementation technology, eg. JAVA, PHP, etc. - Location transparency
Hierarchical Layering	Structure-centric	SOA Design	The degree the services are spread in different layers, with generalised and core (close to data or business core systems) services positioned in the lower layers and services more specific to channels and business requirements positioned in the higher layers.	
Loosely Coupled	Structure-centric Flexibility-centric	SOA Design	The degree to which changes within a subsystem do not create a ripple effect in the behaviour of other parts of the ecosystem (Tiwana et al., 2010).	The loosely coupling applied to: - The service interface to protect changes in one service affecting the rest of the services. - Hide Implementation technology, eg. JAVA, PHP, etc. - Location transparency

SOA Characteristic	Category (Parent Node)	Characteristics Type (SOA Design, Platform)	Definition	Analysis
Standardisation	Information-centric / Information Repository & Flexibility Centric	SOA Design and Platform	The extent consistent rules, guidelines and specifications for common and repeated use of services are formulated and implemented.	Different dimensions of standardisations have been identified as: (7) Consistent IT Technology standards, e.g. XML, WSDL, SOAP, etc. in development and system implementation (8) Consistent IT policies applied to the service design and management, (9) Consistent IT architecture, (10) Consistent Design patterns - pattern oriented as consistent approach in development or management of services to solve a particular problem. This was captured in form of established framework and libraries, many documented and known in form of team knowledge. (11) Consistent service development and management processes to develop and manage services, either explicitly documented or team knowledge. (12) Consistent data semantics and interface definitions as per the internal or external standards, such as AIDX for the aviation industry or Swift data exchange standards for the bank. This addresses existing definitions from (Ross & Weill, 2005), (Tiwana et al., 2010) or (Oh et al., 2007).
Integration	Flexibility-centric		The ability and extent of seamless access and interactions between systems and services internally and externally outside the organisation.	It consists of two factors of: - Connectivity: The extent services and systems are already connected to exchange data internally and externally over a distributed network (should reflect distributed, latency). - Adaptability: Ability to transform or adapt the existing services to new requirements (new service definitions of consumers) to minimise the change in those systems

SOA Characteristic	Category (Parent Node)	Characteristics Type (SOA Design, Platform)	Definition	Analysis
Composibility	Structure-centric		Ability to quickly develop orchestration of independent services with well-defined interfaces in defined sequences to form business processes	Extent the development can be expedited through drag and drop functionality at design time.
Framework Driven	Flexibility-centric		Predefined sets of functionalities built based on design patterns available in the SOA platform for reuse.	
Standardisation	Information-centric / Information Repository & Flexibility Centric	SOA Design and Platform	The extent consistent rules, guidelines and specifications for common and repeated use of services are formulated and implemented.	Different dimensions of standardisations have been identified as: (13) Consistent IT Technology standards, e.g. XML, WSDL, SOAP, etc. in development and system implementation (14) Consistent IT policies applied to the service design and management, (15) Consistent IT architecture, (16) Consistent Design patterns - pattern oriented as consistent approach in development or management of services to solve a particular problem. This was captured in form of established framework and libraries, many documented and known in form of team knowledge. (17) Consistent service development and management processes to develop and manage services, either explicitly documented or team knowledge. Consistent data semantics and interface definitions as per the internal or external standards, such as AIDX for the aviation industry or Swift data exchange standards for the bank. This addresses existing definitions from (Ross & Weill, 2005), (Tiwana et al., 2010) or (Oh et al., 2007).

Appendix D – Interview Protocol

Considerations in the current interview protocol:

1. Project can be a system change or software development, with different team size.
2. Change can be caused by multiple factors such as requirements for development of a new system, requirements to extend an existing system, requirement to reconfigure a system (e.g. infrastructure reconfiguration).

Revised Interview Protocol

Introduction

- Name of the research: IT Agility through Service-Oriented Architecture
- Explain the purpose of the study:
- Explain the interview length: 60 mins per interview
- Explain the format and review process
- Explain the confidentiality and ethics
- Research start and finish dates
- How to contact me:
- Any questions they have before starting:

Demographic & Context (mainly collected from the project documentation)

Interviewee:

- Interviewee's job title, department, how long with organisation, level of involvement in SOA implementations and role in project (embedded level).
- Any outstanding question from the document analysis.

Interview Questions

IT ***Responding Capability*** *in Software Development / System Change Implementation & IT infrastructure change:*

- Before discussing the effect of SOA, can you pls provide a short brief about the project, the type of change introduced?
- Based on your experience with SOA in this project / organisation, did SOA impact the responsiveness of IT to implement the new system / change the system? In what way?
- What aspects of SOA (as a design principles or infrastructure characteristics) were involved in this impact?
- Can you please elaborate on how this characteristic impacted the ability of IT to be responsive now and later (positively or negatively)?
    - From perceptive of processes involved in software development (scoping, requirement, design, implementation, testing, deployment, maintenance), system maintenance or any other IT processes?
    - From knowledge perspective?
    - From project structure or team members being able to make decisions autonomously (independently)?
    - Did you notice any interplay between these factors (e.g. process, knowledge or structure) in their impact on IT responding capability?

IT ***Sensing Capability*** *in Software Development / System Change Implementation Project & IT infrastructure change:*

- Based on your experience with SOA in this project / organisation, did SOA impact the sensing ability of IT to detect new requirements, system changes, etc.? In what way?
- What aspects of SOA (as a design principles or infrastructure characteristics) were involved in this impact?
- Can you please elaborate on how this characteristic impacted the ability of IT to sense the change now and later (positively or negatively?
    - From perceptive of processes involved in software development (scoping, requirement, design, implementation, testing, deployment, maintenance), system maintenance or any other IT processes?
    - From knowledge perspective?

- o From project structure or team members being able to make decisions autonomously (project team members exercise power in decision making to set objectives and how to implement the objectives without approval)?
- o Did you notice any interplay between these factors (e.g. process, knowledge or structure) in their impact on IT sensing capability?

The Original Interview Protocol

Introduction

- Name of the research: Investigation on Impact of Service-Oriented Architecture on IT Agility
- Explain the purpose of the study
- Research start and finish dates
- Explain the interview length: 60 mins per person
- Explain the format and review process
- Explain the confidentiality and ethics
- How to contact me:
- Any questions they have before starting:

Demographic & Context

- Interviewee's job title, department, level of involvement in SOA implementations and role in project (embedded level).
- Job position and how long with the organisation?
- Industry the organisation is operating in?
- Size of the organisation?
- Size of the IT team?
- Size of project team?

Environment:

- How is the environment in which the IT is operating: stable or volatile?
- Are there many requests to change systems or introduce new systems, which are not previously planned (e.g. not in the IT project portfolio)?

SOA Background:

- How long has SOA been practised in the organisation?
- Has SOA been used to automate business processes?
- What technologies are used in the SOA implementation for service development and deployment and SOA platform (integration middleware, ESB or tool used for SOA)?
- What was the driver and objectives for the SOA in the organisation (e.g. agility)? Was that part of business or IT strategy?

SOA Characteristics

- Describe the characteristics that SOA as a design paradigm has provided in the services implemented so far (e.g. modularity) in your organisation.
- Describe the characteristics that SOA infrastructure has provided (e.g. monitoring, connectivity) to manage the systems in your organisation.

SOA Options

Note: These questions are currently at the case level. For the embedded cases (projects), they are changed to past tense (if applicable) and ‘in the organisation’ is replaced with ‘in the project’.

- Please describe the extent of each SOA characteristics in your organisation.
- What options have the identified SOA characteristics provided in sensing changes required in a system in your organisation?
- How has such SOA characteristics enabled the identified option in your organisation?
- What options have each SOA characteristics provided in responding to a system change (configuration or development) in your organisation?
- How has such SOA characteristics enabled the identified option in your organisation?

IT Agility

Note: These questions are currently at the case level. For the embedded cases (projects), they are changed to past tense (if applicable) and ‘in the organisation’ is replaced with ‘in the project’.

- Please describe if the identified SOA options have improved the IT sensing capability?
- If yes, how? If no, why?
- Which processes or routines (if any) are usually involved in your organisation to achieve this?
- Please describe if the identified SOA options have improved the IT responding capability?
- If yes, how? If no, why?
- Which processes or routines (if any) are usually involved in your organisation to achieve this?
- In the case of a project, what was the cause and trigger for the change?
- In the case of a project, what was the outcome of the project (delivered earlier than scheduled within budget, on-time and on-budget, over-time and over-budget).

Other potential factors

- Structure:
    - What is the IT structure in your organisation? How many levels?
    - What is the project structure and size?
    - Level of autonomy: What is the project's governance/control structure (central, distributed)? To what extent the project team has control and can make decisions (High, Med, Low)?
    - IT specification decentralisation: Level of IT decision making on the objectives that IT must accomplish (High, Med, Low)?
    - IT implementation decentralisation: Level of IT decision making on how objectives must be implemented (High, Med, Low)?
- Process:
    - What development methodology is used for software development?
- IT Personnel capabilities:
    - How do you rank ((High, Med, Low) the technical capabilities of IT personnel?
    - How do you rank ((High, Med, Low) the business capabilities of IT personnel?

Bibliography

Abelein, U., Habryn, F., & Becker, A. (2009, 1-4 Sept. 2009). *Towards a holistic framework for describing and evaluating business benefits of a service oriented architecture.* Paper presented at the 13th Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2009.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis*: VTT Finland.

Afshar, M., Cincinatus, M., Hynes, D., Clugage, K., & Patwardhan, V. (2007). Soa governance: Framework and best practices. *Oracle Whitepapers*.

Ahsan, M., & Ngo-Ye, L. (2005). *The Relationship Between IT Infastructure and Strategic Agility in Organizations.* Paper presented at the AMCIS 2005 Proceedings.

Allen, B. R., & Boynton, A. C. (1991). Information architecture: In search of efficient flexibility. *MIS Quarterly, 15*(4), 435-445.

Alliance, A. (2001). Agile manifesto. *Online at* http://www/. *agilemanifesto. org*.

Almeida, R., Pereira, R., & da Silva, M. M. (2013). IT Governance Mechanisms: A Literature Review *Exploring Services Science* (pp. 186-199): Springer.

Amit, R., & Schoemaker, P. J. H. (1993). Strategic assets and organizational rent. *Strategic Management Journal, 14*(1), 33-46.

Amram, M., & Kulatilaka, N. (1999). *Real options: Managing strategic investment in an uncertain world* (Vol. 4): Harvard Business School Press Boston.

Anderson, T. J. (2000). Real options analysis in strategic decision making: An applied approach in a dual options framework. *Journal of Applied Management Studies, 9*(2), 235-255.

Argyres, N. S. (1999). The impact of information technology on coordination: Evidence from the B-2 "Stealth" bomber. *Organization Science, 10*(2), 162-180.

Arsanjani, A., Borges, B., & Holley, K. (2004). Service-Oriented Architecture Components and modeling can make the difference. *Web Services Journal, 9*(1), 34-38.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., & Holley, K. (2008). SOMA: A method for developing service-oriented solutions. *IBM SYSTEMS JOURNAL, 47*(3), 377-396.

Arteta, B., & Giachetti, R. (2004). A measure of agility as the complexity of the enterprise system. *Robotics and Computer-Integrated Manufacturing, 20*(6), 495-503.

Avgerou, C. (2013). Social Mechanisms for Causal Explanation in Social Theory Based IS Research. *Journal of the association for information systems, 14:8*.

Badaracco, J. (1991). *The knowledge link: How firms compete throught strategic alliances*: Harvard Business Press.

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: an Experience Report on Migration to a Cloud-Native Architecture. *IEEE Software, 33*(3), 42-52.

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity* (Vol. 1): The MIT Press.

Bardhan, I., Demirkan, H., Kannan, P., Kauffman, R., & Sougstad, R. (2010). An Interdisciplinary Perspective on IT Services Management and Service Science. *Journal of Management Information Systems, 26*(4), 13-64.

Barney, J. (1986). Strategic Factor Markets: Expectations, Luck, and Business Strategy. *Management Science, 32*(10), 1231-1241.

Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management, 17*(1), 99.

Barney, J., & Arikan, A. M. (2001). The resource-based view: Origins and implications. *The Blackwell handbook of strategic management*, 124-188.

Barua, A., Konana, P., Whinston, A. B., & Yin, F. (2004). An Empirical Investigation of Net-Enabled Business Value. *MIS Quarterly, 28*(4), 585-620.

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M., & Virili, F. (2005). *Extensible architectures: The strategic value of service-oriented architecture in banking.* Paper presented at the European Conference on Information Systems (ECIS).

Baskerville, R., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M., & Virili, F. (2010). The strategic value of SOA: a comparative case study in the banking sector. *International Journal of Information Technology and Management, 9*(1), 30-53.

Beimborn, D., & Joachim, N. (2009) Proposing the relationship between IT business alignment and the business value of service-Oriented architectures in financial firms. *4th International Workshop on Enterprise Applications and Services in the Finance Industry, FinanceCom 2008: Vol. 23 LNBIP* (pp. 78-93). Paris.

Beimborn, D., & Joachim, N. (2010). The joint impact of service-oriented architectures and business process management on business process quality: an empirical evaluation and comparison. *Information Systems and E-Business Management*, 1-30. doi:10.1007/s10257-010-0129-1

Bell, M. (2008). *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*: Wiley.

Benaroch, M. (2002). Managing information technology investment risk: A real options perspective. *Journal of Management Information Systems, 19*(2), 43-84.

Benaroch, M., & Kauffman, R. J. (1999). A case for using real options pricing analysis to evaluate information technology project investments. *Information Systems Research, 10*(1), 70-86.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly*, 369-386.

Benbasat, I., & Weber, R. (1996). Research commentary: Rethinking "diversity" in information systems research. *Information Systems Research, 7*(4), 389-399.

Benbya, H., & McKelvey, B. (2006). Toward a complexity theory of information systems development. *Information technology & people, 19*(1), 12-34.

Benner, M. J., & Tushman, M. L. (2003). Exploitation, exploration, and process management: The productivity dilemma revisited. *Academy of Management Review, 28*(2), 238-256.

Bennett, D. J. (2012). Real Options in Management and Organisational Strategy: A Review of Decision Making and Performance Implications. *International Journal of Management Reviews, 14*(1).

Bharadwaj, A., Keil, M., & Mahring, M. (2009). Effects of information technology failures on the market value of firms. *The Journal of Strategic Information Systems, 18*(2), 66-79.

Bharadwaj, A. S. (2000). A resource-based perspective on information technology capability and firm performance: an empirical investigation. *MIS Quarterly*, 169-196.

Bharadwaj, S. S., Chauhan, S., & Raman, A. (2015). Achieving Business Agility Through

Service-Oriented Architecture in Recovering Markets. In S. Chatterjee, N. P. Singh, D. P. Goyal, & N. Gupta (Eds.), *Managing in Recovering Markets* (pp. 15-26). New Delhi: Springer India.

Bhatt, G. D., Emdad, A., Roberts, N., & Grover, V. (2010). Building and leveraging information in dynamic environments: The role of IT infrastructure flexibility as enabler of organizational responsiveness and competitive advantage. *Information and Management, 47*(7-8), 341-349. doi:10.1016/j.im.2010.08.001

Bhatt, G. D., & Grover, V. (2005). Types of information technology capabilities and their role in competitive advantage: An empirical study. *Journal of Management Information Systems, 22*(2), 253-277.

Bhattacherjee, A. (2012). *Social Science Research: principles, methods, and practices* (Vol. Book 3): USF Tampa Bay Open Access Textbooks Collection.

Bieberstein, N., Bose, S., Walker, L., & Lynch, A. (2005). Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. *IBM SYSTEMS JOURNAL, 44*(4), 691-708.

Black, J. A., & Boal, K. B. (1994). Strategic resources: Traits, configurations and paths to sustainable competitive advantage. *Strategic Management Journal, 15*(S2), 131-148.

Bostrom, R. P., & Heinen, J. S. (1977). MIS problems and failures: a socio-technical perspective, part II: the application of socio-technical theory. *MIS Quarterly*, 11-28.

Bowman, E. H., & Hurry, D. (1993). Strategy through the option lens: An integrated view of resource investments and the incremental-choice process. *Academy of Management Review, 18*(4), 760-782.

Boynton, A. C., Zmud, R. W., & Jacobs, G. C. (1994). The influence of IT management practice on IT use in large organizations. *MIS Quarterly*, 299-318.

Brauer, B., & Kline, S. (2005). SOA governance: a key ingredient of the Adaptive Enterprise. *HP & Systinet White Paper*.

Broadbent, M., & Weill, P. (1997). Management by maxim: how business and IT managers can create IT infrastructures. *Sloan Management Review, 38*, 77-92.

Buzacott, J. A. (1999). The structure of manufacturing systems: Insights on the impact of variability. *International journal of flexible manufacturing systems, 11*(2), 127-146.

Byrd, T. A., & Turner, D. (2000). Measuring the flexibility of information technology infrastructure: Exploratory analysis of a construct. *Journal of Management Information Systems, 17*(1), 167-208.

Byrd, T. A., & Turner, D. E. (2001). An exploratory examination of the relationship between flexible IT infrastructure and competitive advantage. *Information & Management, 39*(1), 41-52. doi:10.1016/s0378-7206(01)00078-7

Cabigiosu, A., & Camuffo, A. (2012). Beyond the “Mirroring” Hypothesis: Product Modularity and Interorganizational Relations in the Air Conditioning Industry. *Organization Science, 23*(3), 686-703.

Cao, G. (2010). A four-dimensional view of IT business value. *Systems Research and Behavioral Science, 27*(3), 267-284. doi:10.1002/sres.1015

Cavaye, A. L. M. (1996). Case study research: a multi,faceted research approach for IS. *Information Systems Journal, 6*(3), 227-242.

Chen, T., Zhang, J., & Lai, K.-K. (2009). An integrated real options evaluating model for information technology projects under multiple risks. *International Journal of Project Management, 27*(8), 776-786.

Clark, C. E., Cavanaugh, N. C., Brown, C. V., & Sambamurthy, V. (1997). Building change-readiness capabilities in the IS organization: Insights from the Bell Atlantic experience. *MIS Quarterly, 21*(4), 425-455.

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: a new perspective on learning and innovation. *Administrative Science Quarterly*, 128-152.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research, 20*(3), 329-354.

Conner, K. R. (1991). A historical comparison of resource-based theory and five schools of thought within industrial organization economics: do we have a new theory of the firm? *Journal of Management, 17*(1), 121-154.

Copeland, T. E., Koller, T., & Murrin, J. (1994). *Valuation: measuring and managing the value of companies* (Vol. 2nd edition): Wiley (New York).

Currie, W. L., & Parikh, M. A. (2006). Value creation in web services: An integrative model. *Journal of Strategic Information Systems, 15*(2), 153-174. doi:10.1016/j.jsis.2005.10.001

D'Aveni, R. A. (1996). Hypercompetition: Managing the Dynamics of Strategic Maneuvering. *The Academy of Management Review, Vol. 21, No. 1* 291-294.

Dai, Q., Kauffman, R. J., & March, S. T. (2007). Valuing information technology infrastructures: a growth options approach. *Information Technology and Management, 8*(1), 1-17.

Daskalakis, S., & Mantas, J. (2008). *Evaluating the impact of a service-oriented framework for healthcare interoperability.* Paper presented at the Studies in health technology and informatics.

de Neufville, R., Hodota, K., Sussman, J., & Scholtes, S. (2008). Using Real options to increase the value of intelligent transportation systems. *Transportation Research Record: Journal of the Transportation Research Board*(2086), 40-47.

Demirkan, H., Kauffman, R., Vayghan, J., Fill, H., Karagiannis, D., & Maglio, P. (2009). Service-oriented technology and management: Perspectives on research and practice for the coming decade. *Electronic Commerce Research and Applications, 7*(4), 356-376.

Dixit, A. K. (1994). *Investment under uncertainty*: Princeton university press.

Dos Santos, B. (1991). *Justifying investments in new information technologies*: Krannert Graduate School of Management, Purdue University.

Dove, R. (2001). *Response ability: the language, structure, and culture of the agile enterprise*: Wiley.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS Quarterly*, 597-636.

Duncan, N. B. (1995). Capturing flexibility of information technology infrastructure: A study of resource characteristics and their measure. *Journal of Management Information Systems*, 37-57.

Eckert, J., Bachhuber, M., Miede, A., Papageorgiou, A., & Steinmetz, R. (2010). *Readiness and Maturity of Service-oriented Architectures in the German Banking Industry*. Paper presented at the 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010).

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *The Academy of Management Review, 14*(4), 532-550.

Eisenhardt, K. M., & Martin, J. (2000). Dynamic capabilities: what are they? *Strategic Management Journal, 21*(10-11), 1105-1121.

Erl, T. (2004). *Service-oriented architecture: a field guide to integrating XML and web services*: Prentice Hall PTR Upper Saddle River, NJ, USA.

Erl, T. (2005). Service Oriented Architecture (SOA): Concepts, Technology, and Architecture: Prentice Hall.

Erradi, A., Anand, S., & Kulkarni, N. (2006, Sep 18-22). *SOAF: An architectural framework for service definition and realization.* Paper presented at the IEEE Services Computing Workshops, Chicago, IL.

Evans, P., & Wurster, T. S. (2000). *Blown to bits: How the new economics of information transforms strategy*: Harvard Business Press.

Fama, E. F., & Jensen, M. C. (1983). Separation of ownership and control. *The Journal of Law & Economics, 26*(2), 301-325.

Feeny, D. F., & Willcocks, L. P. (1998). Core IS capabilities for exploiting information technology. *Sloan Management Review, 39*(3), 9-21.

Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research, 19*(4), 475-493.

Fichman, R. G. (2004). Real options and IT platform adoption: Implications for theory and practice. *Information Systems Research, 15*(2), 132-154.

Fichman, R. G., Keil, M., & Tiwana, A. (2005). Beyond valuation: 'options thinking' in IT project management. *California Management Review, 47*(2), 74-95.

Fink, L., & Neumann, S. (2007). Gaining agility through IT personnel capabilities: The mediating role of IT infrastructure capabilities. *Journal of the Association of Information Systems, 8*(8), 440-462.

Fink, L., & Neumann, S. (2009). Exploring the perceived business value of the flexibility enabled by information technology infrastructure. *Information & Management, 46*(2), 90-99.

Gamba, A., & Fusari, N. (2009). Valuing modularity as a real option. *Management Science, 55*(11), 1877-1896.

Ganguly, A., Nilchiani, R., & Farr, J. V. (2009). Evaluating agility in corporate enterprises. *International Journal of Production Economics, 118*(2), 410-423.

Goodhue, D. L., Chen, D. Q., Boudreau, M. C., Davis, A., & Cochran, J. D. (2009). Addressing Business Agility Challenges with Enterprise Systems. *MIS Quarterly Executive, 8*(2), 73-87.

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly, 30*(3), 611.

Gregor, S., & Iivari, J. (2007). *Designing for Mutability in Information Systems Artifacts*: Canberra: ANU Press.

Guba, E. G. (1990). *The paradigm dialog*: Sage.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research, 2*, 163-194.

Haeckel, S. H. (1999). *Adaptive enterprise: Creating and leading sense-and-respond organizations*: Harvard Business Press.

Hagel, J., & Brown, J. S. (2001). Your next IT strategy. *Harvard Business Review, 79*(9), 105-115.

Hagel, J., & Brown, J. S. (2005). Flexibility for Better IT-Business Alignment. *CIO Magazine.* Retrieved from http://www.cio.com/article/print/10514

Haines, M. N., & Haseman, W. (2009). *Service-oriented architecture adoption patterns.* Paper presented at the 42nd Hawaii International Conference on System Sciences, HICSS'09. .

Helfat, C. E., Finkelstein, S., Mitchell, W., Peteraf, M., Singh, H., Teece, D., & Winter, S. G. (2007). *Dynamic capabilities: Understanding strategic change in organizations*: Wiley-Blackwell.

Helfat, C. E., & Peteraf, M. (2009). Understanding dynamic capabilities: progress along a developmental path. *Strategic organization, 7*(1), 91.

Henningsson, S., Svensson, C., & Vallen, L. (2007). *Mastering the integration chaos following frequent M&As: IS Integration with SOA Technology.* Paper presented at the 40th Hawaii International Conference on System Sciences, 2007. HICSS 2007. .

Hirschheim, R., & Klein, H. K. (1992). *A research agenda for future information systems development methodologies.* Paper presented at the Challenges and strategies for research in systems development.

Hoberg, P., Wollersheim, J., & Krcmar, H. (2012). The business perspective on cloud computing-a literature review of research on cloud computing.

Hoetker, G. (2006). Do modular products lead to modular organizations? *Strategic Management Journal, 27*(6), 501-518.

Holland, J. H. (1992). Complex adaptive systems. *Daedalus*, 17-30.

Huang, C., & Hu, Q. (2004). Integrating web services with competitive strategies: the balanced scorecard approach. *Communications of the Association for Information Systems (Volume 13, 2004), 57*(80), 57.

Iyer, B., Freedman, J., Gaynor, M., & Wyner, G. (2003). Web services: enabling dynamic business networks. *The Communications of the Association for Information Systems, 11*(1), 38.

Jensen, M. C., & Meckling, W. H. (1992). Specific and general knowledge and organizational structure.

Joachim, N. (2011). *A Literature Review of Research on Service-Oriented Architectures (SOA): Characteristics, Adoption Determinants, Governance Mechanisms, and Business Impact.* Paper presented at the AMCIS 2011 Proceedings - All Submissions. Paper 339.

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.* Paper presented at the ICIS 2011 Proceedings.

Joachim, N., Beimborn, D., & Weitzel, T. (2011). *What Are Important Governance and Management Mechanisms to Achieve IT Flexibility in Service-Oriented Architectures (SOA)?: An Empirical Exploration.* Paper presented at the 44th Hawaii International Conference on System Sciences (HICSS), 2011

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. doi:http://dx.doi.org/10.1016/j.jsis.2012.10.003

Karsten, H., Lyytinen, K., Hurskainen, M., & Koskelainen, T. (2001). Crossing boundaries and conscripting participation: representing and integrating knowledge in a paper machinery project. *European Journal of Information Systems, 10*(2), 89-98.

Kim, Y. J., & Sanders, G. L. (2002). Strategic actions in information technology investment based on real option theory. *Decision Support Systems, 33*(1), 1-11.

Knoll, K., & Jarvenpaa, S. L. (1994). *Information technology alignment or "fit" in highly turbulent environments: the concept of flexibility.* Paper presented at the Proceedings of the 1994 computer personnel research conference on Reinventing IS: managing information technology in changing organizations: managing information technology in changing organizations.

Kogut, B., & Kulatilaka, N. (2004). Real options pricing and organizations: The contingent risks of extended theoretical domains. *Academy of Management*

Review, 29(1), 102-110.

Kogut, B., & Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science, 3*(3), 383-397.

Kohli, R., & Grover, V. (2008). Business value of IT: an essay on expanding research directions to keep up with the times. *Journal of the association for information systems, 9*(1), 1.

Kokko, T., Antikainen, J., & Systa, T. (2009). *Adopting SOA-Experiences from Nine Finnish Organizations*. Paper presented at the 13th European Conference on Software Maintenance and Reengineering, CSMR'09. .

Kontogiannis, K., Lewis, G., Smith, D., Litoiu, M., Muller, H., Schuster, S., & Stroulia, E. (2007). *The landscape of service-oriented systems: A research perspective*. Paper presented at the ICSE Workshops: International Workshop on Systems Development in SOA Environments. SDSOA'07

Kumar, R. L. (2004). A framework for assessing the business value of information technology infrastructures. *Journal of Management Information Systems, 21*(2), 11-32.

Kumar, S. (2007). Impact of Service-Oriented Architecture Adoption on Electronic Supply Chain Performance. *AMCIS 2007 Proceedings*, 411.

Lavie, D. (2006). Capability reconfiguration: An analysis of incumbent responses to technological change. *Academy of Management Review, 31*(1), 153-174.

Lee, A. S. (1989). A scientific methodology for MIS case studies. *MIS Quarterly*, 33-50.

Lee, A. S. (1991). Integrating positivist and interpretive approaches to organizational research. *Organization Science, 2*(4), 342-365.

Lee, G., & Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. *MIS Quarterly, 34*(1), 87-114.

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.

Lee, O. D., Banerjee, P., Lim, K. H., Kumar, K., Hillegersberg, J., & Wei, K. K. (2006). Aligning IT components to achieve agility in globally distributed system development. *Communications of the Acm, 49*(10), 48-54.

Lee, O. K. D. (2012). IT-Enabled Organizational Transformations to Achieve Business Agility. *Review of Business Information Systems (RBIS), 16*(2), 43-52.

Legner, C., & Heutschi, R. (2007). *SOA Adoption in Practice-Findings from Early SOA Implementations.* Paper presented at the Proceedings of the 15th European Conference on Information Systems, St. Gallen, Switzerland: Association of Information Systems.

Leidner, D. E., Pan, G., & Pan, S. L. (2009). The role of IT in crisis response: Lessons from the SARS and Asian Tsunami disasters. *The Journal of Strategic Information Systems, 18*(2), 80-99.

Lim, S. L., Ishikawa, F., Platon, E., & Cox, K. (2008). *Towards agile service-oriented business systems: A directive-oriented pattern analysis approach.* Paper presented at the Proceedings - 2008 IEEE International Conference on Services Computing, SCC 2008, Honolulu, HI.

Lowry, P. B., & Wilson, D. (2016). Creating agile organizations through IT: The influence of internal IT service perceptions on IT service quality and IT agility. *The Journal of Strategic Information Systems, 25*(3), 211-226.

Lu, Y., & Ramamurthy, K. (2011). Understanding the Link Between Information Technology Capability and Organizational Agility: An Empirical Examination.

MIS Quarterly, 35(4), 931-954.

Lui, T. W., & Piccoli, G. (2007). Degrees of agility: Implications for information systems design and firm strategy *Agile Information Systems: Conceptualization, Construction and Management* (pp. 122-133): Butterworth-Heinemann, Amsterdam.

Luthria, H., & Rabhi, F. (2008). Organizational Constraints to Realizing Business Value from Service Oriented Architectures: An Empirical Study of Financial Service Institutions. *Service-Oriented Computing-ICSOC 2008*, 256-270.

Luthria, H., & Rabhi, F. (2009a). Building the Business Case for SOA: A Study of the Business Drivers for Technology Infrastructure Supporting Financial Service Institutions. *Enterprise Applications and Services in the Finance Industry*, 94-107.

Luthria, H., & Rabhi, F. (2009b). 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*, 39-56.

Luthria, H., & Rabhi, F. (2009c). Using Service Oriented Computing for Competitive Advantage. *AMCIS 2009 Proceedings, 140*.

Magnusson, J. (2013). Intentional Decentralization and Instinctive Centralization: A Revelatory Case Study of the Ideographic Organization of IT. *Information Resources Management Journal (IRMJ), 26*(4), 1-17.

Makadok, R. (2001). Toward a synthesis of the resource-based and dynamic-capability views of rent creation. *Strategic Management Journal, 22*(5), 387-401.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR), 26*(1), 87-119.

Manes, A. (2005). The elephant has left the building. *at URL:* http://eai/. *ittoolbox. com/documents/industry-articles/the-elephant-has-left-the-building-3633, Intelligent Enterprise*.

Mangalaraj, G., Mahapatra, R. K., & Nerur, S. (2009). Acceptance of software process innovations–the case of extreme programming. *European Journal of Information Systems, 18*(4), 344-354.

Manson, S. (2001). Simplifying complexity: a review of complexity theory. *Geoforum, 32*(3), 405-414.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science, 2*(1), 71-87.

Marks, E., & Bell, M. (2006). *Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology*: John Wiley & Sons, Inc. New York, NY, USA.

Markus, M. L., & Robey, D. (1988). Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science, 34*(5), 583-598.

Maurer, C. (2010). *Measuring Information Systems Agility: Construct Definition and Scale Development.* Paper presented at the Proceedings of the Southern Association for Information Systems Conference.

McAvoy, J., & Butler, T. (2009). The role of project management in ineffective decision making within Agile software development projects. *European Journal of Information Systems, 18*(4), 372-383.

McCutcheon, D. M., & Meredith, J. R. (1993). Conducting case study research in operations management. *Journal of Operations Management, 11*(3), 239-256. doi:10.1016/0272-6963(93)90002-7

McGrath, R. G. (1997). A real options logic for initiating technology positioning investments. *Academy of Management Review, 22*(4), 974-996.

McKeen, J. D., Guimaraes, T., & Wetherbe, J. (1994). The Relationship Between User Participation and User Satisfaction: An Investigation of Four Contingency Factors. *MIS Quarterly, 18*(4).

Meyer, M. H., & Curley, K. F. (1991). An applied framework for classifying the complexity of knowledge-based systems. *MIS Quarterly, 15*(4), 455-472.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook.* (Second ed.): Beverly Hills: Sage Publications.

Miller, K. D. (2002). Knowledge inventories and managerial myopia. *Strategic Management Journal, 23*(8), 689-706.

Mohr, J., & Nevin, J. R. (1990). Communication strategies in marketing channels: a theoretical perspective. *The Journal of Marketing*, 36-51.

Moitra, D., & Ganesh, J. (2005). Web services and flexible business processes: towards the adaptive enterprise. *Information and Management, 42*(7), 921-933.

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.* Paper presented at the European Conference on Information Systems, 2007.

Mumford, E. (2003). *Redesigning human systems*: Irm Press.

Nagel, R. N., & Dove, R. (1991). *21st century manufacturing enterprise strategy: An Industry-Led View* (Vol. 1): Lehigh University Press.

Nambisan, S. (2002). Complementary product integration by high-technology new ventures: The role of initial technology strategy. *Management Science, 48*(3), 382-398.

Nevo, S., & Wade, M. (2011). Firm-level benefits of IT-enabled resources: A conceptual extension and an empirical assessment. *The Journal of Strategic Information Systems*.

Nevo, S., & Wade, M. R. (2010). The Formation and Value of IT-Enabled Resources: Antecedents and Consequences. *MIS Quarterly, 34*(1), 10.

Newcomer, E., & Lomow, G. (2004). *Understanding SOA with web services (independent technology guides)*: Addison-Wesley Professional.

Niemann, M., Eckert, J., Repp, N., & Steinmetz, R. (2008, 14-17th August 2008). *Towards a generic governance model for service-oriented architectures.* Paper presented at the Proceedings of the Fourteenth Americas Conference on Information Systems, Toronto, ON,, Toronto, ON, Canada.

O'Brien, J. A., & Marakas, G. M. (2005). *Introduction to information systems* (Vol. 13): McGraw-Hill/Irwin.

O'Brien, L., Merson, P., & Bass, L. (2007). *Quality attributes for service-oriented architectures*. Paper presented at the Proceedings of the International Workshop on Systems Development in SOA Environments.

Offermann, P., Hoffmann, M., & Bub, U. (2009). *Benefits of SOA: Evaluation of an implemented scenario against alternative architectures.* Paper presented at the 13th Enterprise Distributed Object Computing Conference Workshops, EDOCW 2009.

Oh, L., Leong, Y., Teo, H., & Ravichandran, T. (2007). *Service-oriented architecture and organizational integration: An empirical study of IT-enabled sustained competitive advantage.* Paper presented at the 28th International Conference on Information Systems, Montreal.

Oosterhout, M. P. A. V., Waarts, E., & Hillegersberg, J. V. (2006). Change factors

requiring agility and implications for IT. *European Journal of Information Systems, 15*(2), 132-145.
Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research, 2*(1), 1-28.
Orlikowski, W. J., & Iacono, C. S. (2001). Research commentary: Desperately seeking the" it" in it research—a call to theorizing the it artifact. *Information Systems Research, 12*(2), 121-134.
Orlikowski, W. J., & Iacono, C. S. (2006). Desperately seeking the 'IT' in IT research: A call to theorizing the IT artifact. *Information systems: the state of the field, 34*, 19.
Orton, J. D., & Weick, K. E. (1990). Loosely coupled systems: A reconceptualization. *Academy of Management Review*, 203-223.
Overby, E., Bharadwaj, A., & Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems, 15*(2), 120-131.
Papazoglou, M. P. (2003, Dec 10-12). *Service-oriented computing: Concepts, characteristics and directions.* Paper presented at the 4th International Conference on Web Information Systems Engineering (WISE 2003), Rome, Italy.
Papazoglou, M. P. (2008). Service-Oriented Computing: A research roadmap. *INTERNATIONAL JOURNAL OF COOPERATIVE INFORMATION SYSTEMS, 17*(2), 223-255.
Papazoglou, M. P., & Van Den Heuvel, W. J. (2006). Service-oriented design and development methodology. *International Journal of Web Engineering and Technology, 2*(4), 412-442.
Papazoglou, M. P., & Yang, J. (2002). *Design methodology for web services and business processes.* Paper presented at the Proc. of the 3rd International Workshop on Technologies for E-Servcies, London, UK.
Park, Y. K., & El Sawy, O. A. (2012). *Discovering the Multifaceted Roles of Information Technologies with a Holistic Configurational Theory Approach.* Paper presented at the 45th Hawaii International Conference on System Sciences.
Patton, M. Q. (1999). Enhancing the quality and credibility of qualitative analysis. *Health services research, 34*(5 Pt 2), 1189.
Pavlou, P. A., & El Sawy, O. A. (2006). From IT leveraging competence to competitive advantage in turbulent environments: The case of new product development. *Information Systems Research, 17*(3), 198-227.
Pentland, B. T. (1999). Building process theory with narrative: From description to explanation. *Academy of Management Review, 24*(4), 711-724.
Peppard, J., & Ward, J. (2004). Beyond strategic information systems: towards an IS capability. *The Journal of Strategic Information Systems, 13*(2), 167-194. doi:10.1016/j.jsis.2004.02.002
Perry, C. (2008). A structured approach to presenting theses: notes for students and their supervisors., 1, 20 October 2009.
Philips, D., & Burbules, N. (2000). *Postpositivism and Educational Research*: Rowman & Littlefield.
Pinsonneault, A., & Kraemer, K. L. (1993). Survey Research Methodology in Management Information Systems: An Assessment. Retrieved from http://escholarship.org/uc/item/6cs4s5f0 website:
Rahrovani, Y., & Pinsonneault, A. (2012). On the Business Value of Information Technology: A Theory of Slack Resources. *Information Systems Theory*, 165-198.
Ramollari, E., Dranidis, D., & Simons, A. J. H. (2007). *A Survey of Service Oriented*

Development Methodologies. Paper presented at the The 2nd European Young Researchers Workshop on Service Oriented Computing.

Rangan, S. (1998). Do multinationals operate flexibly? Theory and evidence. *Journal of International Business Studies*, 217-237.

Ravichandran, T., Liu, Y., Han, S., & Hasan, I. (2009). Diversification and firm performance: Exploring the moderating effects of information technology spending. *Journal of Management Information Systems, 25*(4), 205-240.

Rettig, C. (2007). The trouble with enterprise software. *MIT Sloan Management Review, 49*(1), 21.

Richter, J., & Basten, D. (2016). In Search of Explanations: Conceptualizing the Relationship between Service-oriented Architecture and Organizational Agility.

Roberts, N., & Grover, V. (2012). Leveraging Information Technology Infrastructure to Facilitate a Firm's Customer Agility and Competitive Activity: An Empirical Investigation. *Journal of Management Information Systems, 28*(4), 231-270.

Ross, J. W. (2003). Creating a strategic IT architecture competency: Learning in stages. *MIT Sloan Management Review*(Working Paper: 4314 - 03).

Ross, J. W., Beath, C. M., & Goodhue, D. L. (1996). Develop long-term competitiveness through IT assets. *Sloan Management Review, 38*(1), 31-42.

Ross, J. W., & Weill, P. (2005). A matrixed approach to designind it governance. *MIT Sloan Management Review, 46*(2), 26-34.

Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. *MIS Quarterly, 27*(2), 237-263.

Sanchez, R. (1996). *Dynamics of competence-based competition: Theory and practice in the new strategic management*: Pergamon.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product organization and design. *Strategic Management Journal, 17(1) 63–76.*

Sanyal, S., & Sett, P. (2011). Applying real options theory to HRM: an empirical study of IT software firms in India. *The International Journal of Human Resource Management, 22*(01), 72-102.

Sarker, S., & Lee, A. S. (2003). Using a case study to test the role of three key social enablers in ERP implementation. *Information & Management, 40*(8), 813-829.

Sarker, S., & Sarker, S. (2009). Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research, 20*(3), 440-461.

Schelp, J., & Aier, S. (2009). *SOA and EA-sustainable contributions for increasing corporate agility.* Paper presented at the 42nd Hawaii International Conference on System Sciences, 2009. HICSS'09. .

Schelp, J., & Stutz, M. (2007). SOA Governance. *HMD-Praxis der Wirtschaftsinformatik, 42*(253), 66-73.

Schepers, T., Iacob, M., & Van Eck, P. (2008). *A lifecycle approach to SOA governance.*

Schilling, M. A. (2000). Toward a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review, 25*(2), 312-334.

Schulte, S., Repp, N., Eckert, J., Berbner, R., Blanckenburg, K., Schaarschmidt, R., & Steinmetz, R. (2008). General Requirements of Banks on IT Architectures and the Service-Oriented Architecture Paradigm. *Enterprise Applications and Services in the Finance Industry*, 66-80.

Sen, A. (2008). Capability and well-being. In D. M. HAUSMAN (Ed.), *The Philosophy of Economics* (pp. 270): CAMBRIDGE UNIVERSITY PRESS.

Sengupta, K., & Masini, A. (2008). IT agility: striking the right balance. *Business Strategy Review, 19*(2), 42-48. doi:10.1111/j.1467-8616.2008.00534.x

Sharifi, H., & Zhang, Z. (1999). A methodology for achieving agility in manufacturing organisations: An introduction. *International Journal of Production Economics, 62*(1), 7-22.

Sherehiy, B., Karwowski, W., & Layer, J. K. (2007). A review of enterprise agility: Concepts, frameworks, and attributes. *International Journal of Industrial Ergonomics, 37*(5), 445-460.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the american philosophical society*, 467-482.

Simon, H. A. (2002). Near decomposability and the speed of evolution. *Industrial and corporate change, 11*(3), 587-599.

Sprott, D. (2004). The Business Case for Service Oriented Architecture. *CBDI Journal*.

Star, S. L., & Ruhleder, K. (1996). Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research, 7*(1), 111-134.

Stieninger, M., & Nedbal, D. (2014). Characteristics of cloud computing in the business context: A systematic literature review. *Global Journal of Flexible Systems Management, 15*(1), 59-68.

Su, N., Akkiraju, R., Nayak, N., & Goodwin, R. (2009). Shared services transformation: Conceptualization and valuation from the perspective of real options. *Decision Sciences, 40*(3), 381-402.

Tallon, P. P. (2007). Does IT pay to focus? An analysis of IT business value under single and multi-focused business strategies. *The Journal of Strategic Information Systems, 16*(3), 278-300.

Tallon, P. P. (2008). Inside the adaptive enterprise: an information technology capabilities perspective on business process agility. *Information Technology and Management, 9*(1), 21-36.

Tallon, P. P., Kraemer, K. L., & Gurbaxani, V. (2000). Executives' perceptions of the business value of information technology: A process-oriented approach. *Journal of Management Information Systems, 16*(4), 145-173.

Tallon, P. P., & Pinsonneault, A. (2011). Competing perspectives on the link between strategic information technology alignment and organizational agility: insights from a mediation model. *MIS Quarterly, 35*(2), 463-484.

Taudes, A. (1998). Software growth options. *Journal of Management Information Systems, 15*(1), 165-185.

Taudes, A., Feurstein, M., & Mild, A. (2000). Options analysis of software platform decisions: A case study. *MIS Quarterly, 24*(2), 227-243.

Teece, D. J. (2007). Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal, 28*(13), 1319-1350.

Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal, 18*(7), 509-533.

Terlouw, L., & Dietz, J. (2008, Jun). *Comparing methodologies for service-orientation using the Generic System Development Process.* Paper presented at the 6th International Workshop on Security in Information Systems held in Conjunction with the ICEIS, Barcelona, SPAIN.

Tiwana, A., & Keil, M. (2007). Does peripheral knowledge complement control? An empirical test in technology outsourcing alliances. *Strategic Management Journal, 28*(6), 623-634.

Tiwana, A., Keil, M., & Fichman, R. G. (2006). Information systems project continuation in escalation situations: A real options model. *Decision Sciences, 37*(3), 357-391.
Tiwana, A., & Kim, S. K. (2015). Discriminating IT governance. *Information Systems Research, 26*(4), 656-674.
Tiwana, A., & Konsynski, B. (2010). Complementarities between organizational IT architecture and governance structure. *Information Systems Research, 21*(2), 288-304.
Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research, 21*(4), 675-687.
Tiwana, A., Konsynski, B., & Venkatraman, N. (2013). Special Issue: Information Technology and Organizational Governance: The IT Governance Cube. *Journal of Management Information Systems, 30*(3), 7-12.
Trigeorgis, L. (1996). *Real Options: Managerialflexibility and Strategy Inresource Allocation*: the MIT Press.
Trinh-Phuong, T., Molla, A., & Peszynski, K. (2010). *Enterprise Systems and Organisational Agility: Conceptualizing the Link.* Paper presented at the 21st Australasian Conference on Information Systems, Brisbane.
Tseng, Y. H., & Lin, C. T. (2011). Enhancing enterprise agility by deploying agile drivers, capabilities and providers. *Information Sciences, 181*(17), 3693-3708.
Tsourveloudis, N. C., & Valavanis, K. P. (2002). On the measurement of enterprise agility. *Journal of Intelligent & Robotic Systems, 33*(3), 329-342.
Vidgen, R., & Wang, X. (2006). From business process management to business process ecosystem. *Journal of Information Technology, 21*(4), 262-271.
Vitharana, P., Bhaskaran, K., Jain, H., Wang, H., & Zhao, L. (2007). *Service-oriented enterprises and architectures: state of the art and research opportunities.* Paper presented at the AMCIS 2007 Proceedings.
Volberda, H. W., & Lewin, A. Y. (2003). Co-evolutionary Dynamics Within and Between Firms: From Evolution to Co-evolution. *Journal of Management studies, 40*(8), 2111-2136.
Von Bertalanffy, L. (1972). The History and Status of General Systems Theory. *Academy of Management Journal, 15*(4), 407-426.
Wade, M., & Hulland, J. (2004). Review: The resource-based view and information systems research: Review, extension, and suggestions for future research. *MIS Quarterly, 28*(1), 107-142.
Walker, L. (2007). IBM business transformation enabled by service-oriented architecture. *IBM SYSTEMS JOURNAL, 46*(4), 651-667.
Warwick, D. P., & Lininger, C. A. (1975). *The sample survey: Theory and practice*: McGraw-Hill.
Webster, J. (1998). Desktop videoconferencing: experiences of complete users, wary users, and non-users. *MIS Quarterly*, 257-286.
Weick, K. E. (1976). Educational organizations as loosely coupled systems. *Administrative Science Quarterly*, 1-19.
Wernerfelt, B. (1984). A resource-based view of the firm. *Strategic Management Journal, 5*(2), 171-180.
Windley, P. (2006). SOA governance: rules of the game. *InfoWorld, v28 n4 p29 (5)*.
Winter, S. (2003). Understanding dynamic capabilities. *Strategic Management Journal, 24*(10), 991-995.
Xia, W., & Lee, G. (2003). Complexity of information systems development projects: Conceptualization and measurement development. *Journal of Management*

Information Systems, 22(1), 45-83.
Yang, C., & Liu, H. M. (2012). Boosting firm performance via enterprise agility and network structure. *Management Decision, 50*(6), 3-3.
Yin, R. K. (1994). *Case study research: design and methods*: Thousand Oaks, CA.
Yin, R. K. (2009). *Case study research: Design and methods* (Vol. 5): Sage Publications, Inc.
Yoon, T., & Carter, P. (2007). *Investigating the antecedents and benefits of SOA implementation: a multi-case study approach.* Paper presented at the AMCIS 2007 Proceedings.
Yousif, M., Magnusson, J., & Pessi, K. (2017). *IT Agility: Current State, Organizational Contingencies, and Future Research Avenues.* Paper presented at the Proceedings of the 50th Hawaii International Conference on System Sciences.
Yousif, M., & Pessi, K. (2016). *It Agility Research Review: Thematic Analysis and Categorization of Literature.* Paper presented at the publishing at: 20th Pacific Asia Conference on Information Systems, June 27-July 1, 2016, Chiayi, Taiwan.
Yusuf, Y. Y., Sarhadi, M., & Gunasekaran, A. (1999). Agile manufacturing:: The drivers, concepts and attributes. *International Journal of Production Economics, 62*(1), 33-43.
Zaheer, A., & Zaheer, S. (1997). Catching the wave: Alertness, responsiveness, and market influence in global electronic networks. *Management Science*, 1493-1509.
Zahra, S. A., & George, G. (2002). Absorptive capacity: A review, reconceptualization, and extension. *Academy of Management Review, 27*(2), 185-203.
Zahra, S. A., & George, G. (2002). The net-enabled business innovation cycle and the evolution of dynamic capabilities. *Information Systems Research, 13*(2), 147.
Zahra, S. A., Sapienza, H. J., & Davidsson, P. (2006). Entrepreneurship and dynamic capabilities: a review, model and research agenda*. *Journal of Management studies, 43*(4), 917-955.
Zhang, Z., & Sharifi, H. (2007). Towards theory building in agile manufacturing strategy—a taxonomical approach. *Engineering Management, IEEE Transactions on, 54*(2), 351-370.
Zimmermann, O., Krogdahl, P., & Gee, C. (2004). Elements of service-oriented analysis and design. *IBM developerworks*.
Zollo, M., & Winter, S. G. (2002). Deliberate learning and the evolution of dynamic capabilities. *Organization Science, 13*(3), 339-351.