An Interactive Metaheuristic Search Framework for Software Service Identification from Business Process Models

Hamzeh Jamil Aljawawdeh

The University of the West of England

Faculty of Environment and Technology

This thesis is submitted as partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Software Engineering Research Group (SERG)

September 2018

Abstract

In recent years, the Service-Oriented Architecture (SOA) model of computing has become widely used and has provided efficient and agile business solutions in response to inevitable and rapid changes in business requirements. Software service identification is a crucial component in the production of a service-oriented architecture and subsequent successful software development, yet current top-down service identification methods have limitations. For example, service identification methods are either not sufficiently comprehensive to handle the totality of service identification activities, or they lack for automation and rely heavily on the software engineer to do some activities, or they pay insufficient attention to quality checks of the service identification method or the resulting services. To address these limitations, comprehensive computationally intelligent support for software engineers when deriving software services from an organisation's business process models shows great potential, especially when the impact of human preference on the quality of the resulting solutions can be incorporated. Accordingly, this research attempts to apply interactive metaheuristic search to effectively bridge the gap between business and SOA technology and so increase business agility.

A novel, comprehensive framework is introduced that is driven by domain independent role-based business process models, and uses an interactive metaheuristic search-based software service identification approach based on a genetic algorithm, while adhering to SOA principles. Termed BPMiSearch, the framework is composed of three main layers. The first layer is concerned with processing inputs from business process models into search space elements by modelling input data and presenting them at an appropriate level of granularity. The second layer focuses on identifying software services from the specified search space. The third layer refines the resulting software services to map the business elements in the resulting candidate services to the corresponding software service components based on web service standards. The proposed BPMiSearch framework has been evaluated by applying it to a healthcare domain case study, specifically, Cancer Care and Registration (CCR) business processes at the King Hussein Cancer Centre, Amman, Jordan. The impact of software engineer interaction on the quality of the outcomes in terms of search effectiveness, efficiency, and level of user satisfaction, is assessed. Results show that BPMiSearch has rapid search performance to positively support software engineers in the identification of services from role-based business process models while adhering to SOA principles. High-quality software services are identified that might not have been arrived at manually by software engineers. Furthermore, it is found that BPMiSearch is sensitive and responsive to software engineer interaction resulting in a positive level of user trust, acceptance, and satisfaction with the candidate software services.

Acknowledgement

I would like to sincerely thank my supervisors Dr Mohammed Odeh and Dr Chris Simons for their support and guidance during this PhD journey. Thank you, Mohammed and Chris, for taking me on a wonderful journey of exploring research ideas. Your advice on research has been very enlightening, inspiring, and always motivating.

I am grateful to thank Dr Asem Mansour, Dr Abdelghani Tbakhi, Dr Imad Traish, Mr Nawras Lebzo, Mrs Enas Jazairy, Mr Mohannad Abdulmalik from the King Hussein Cancer Centre (KHCC), Jordan, who supported and facilitated the research case study and data analysis.

I am also grateful to my other PhD colleagues and friends with whom I shared the office and some of the difficulties in the way. Rawad, Ahmad Samhan, Mohammad Sabri, Sami, Shadan, Suhair, Eman, Noura, Abdullah Al-Khateeb, Abdullah Jalabneh, Gaith, Mais, Bushra, Dalia, Daniel, Barkha, and Kayode. Furthermore, I would like to thank all colleagues at UWE Graduate School for the continues support and encouragement, Marisa, Helen, Patricia, Samantha, Bridgett, Caterina, Victoria and Yasmine.

I wish to thank my parents who have provided me with the ideal environment to grow up in, and from whom I have inherited my determination and strength. I would like also to thank my brothers Ahmad, Jawad, Yazan and Mohammad and sisters Alaa and Deema for all their belief that I was capable of taking on the PhD. Thank you all for the constant and unconditional love and support.

Finally, I wish to thank my wife, Walaa has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point. She gave me constant support and help. Along with her, I want to acknowledge my three children; Yahya, Yousef, and Bisan. They have never known their dad as anything but a student, it seems. You are my great sources of love and relief from the scholarly endeavor.

I could not have done any of this without you. Thank you all.

Table of Contents

List of Figures

List of Tables

Abbreviations

ACO: Ant Colony Optimisation.

BPA: Business Process Architecture.

BPEL: Business Process Execution Language.

BPE: Business Process Element.

BPM: Business Process Model.

BPMiSearch: A comprehensive framework for service identification from business process models using interactive metaheuristic search. This framework is the main contribution of this research.

BPMN: Business Process Modelling and Notation.

BS: Business Services.

CCR: Cancer Care and Registration.

CRUD: Create Read Update Delete.

DSRM: Design Science Research Methodology.

EA: Evolutionary Algorithm.

EC: Evolutionary Computing.

GA: Genetic Algorithm.

iSBSE: Interactive Search Based Software Engineering.

IS: Information Systems.

IT: Information Technology.

KHCC: King Hussein Cancer Centre.

MOEA: Multi-Objective Evolutionary Algorithm.

OMG: Object Management Group.

SBSE: Search Based Software Engineering.

SIM: Service Identification Method.

SOA: Service Oriented Architecture.

SOEA: Single Objective Evolutionary Algorithm.

SSE: Search Space Element.

srBPA: Semantic-Enriched Riva Business Process Architecture.

UDDI: The Universal Description, Discovery and Integration (UDDI)

WSDL: Web Service Description Language.

XML: Extensible Markup Language.

Chapter 1
Introduction

Service Oriented Architecture (SOA) has been widely utilised in an attempt to provide effective business-IT alignment (Zadeh et al., 2016). Utilising SOA systems helps software engineers to keep abreast of the inevitable and rapid changes in business environments (Yousef et al., 2014). Identifying software services that satisfy the business demand is one of the key activities of developing SOA solutions (Papazoglou, 2007). However, there is a body of evidence to suggest that identifying the right software services is a challenging, non-trivial, and cognitively demanding task to perform because the nature of SOA projects is inherently complex while at the same time SOA as an architectural style promises to preserve the software quality (Shahrbanoo, Ali, and Mehran, 2012). Identifying incorrect services at this stage causes deleterious consequences for downstream development (Jamshidi et al., 2012), as errors made at this stage can be propagated through to the next stages of design, implementation, and verification (Kazemi et al., 2011).

Existing service identification methods (SIMs) can suffer from serious shortcomings. For example, the majority of SIMs are not fully comprehensive as they do not cover all the phases of service identification (i.e., the majority of SIMs focus on the service identification phase alone); therefore, some critical aspects of service identification may be ignored (Zadeh et al., 2016). In addition, there is a lack of computational support such that SIMs rely heavily on the software engineer to fulfil the service identification activities manually (Azevedo et al., 2009). Furthermore, quality assessment procedures utilised by the majority of contemporary SIMs are not sufficient to fully examine the resulting candidate services as well as the service identification approach itself using both quantitative and qualitative evaluation methods (Zadeh et al., 2016; Huergo et al., 2014; Gu and Lago, 2010; Bianchini et al., 2014). Insufficient service assessment can produce low quality service solutions that do not adhere to SOA principles (e.g., low coupling and abstraction of the underlying business), and also that they do not guarantee the feasibility of the resulting solutions (Papazoglou, 2007).

Computational intelligence has been previously applied to address a variety of development problems in software engineering. For example, Search-Based Software Engineering (SBSE) uses a range of bio-inspired metaheuristic search and optimisation techniques such as genetic algorithms or ant colony optimisation to solve software systems engineering problems (Harman and Jones 2001; Simons et al., 2012). Finding the optimum solution using dynamic programming or linear programming may not be feasible for large-scale software engineering problems because of computational scale and complexity. Thus, researchers and practitioners have used metaheuristic search techniques to find near-optimal or good-enough solutions (Harman, 2011).

Incorporating the software engineer 'in-the-loop' of optimisation algorithms, interactive search-based methods have been used to address different software development activities such as software design (Simons and Parmee, 2012), testing (Afzal, Torkar and Feldt, et al., 2009), requirements management (Pitangueira et al., 1999), refactoring (Mariani and Vergilio, 2017), and maintenance (Bavota, Penta and Oliveto, 2014). Where research problems require both objective and subjective evaluation methods to evaluate the quality of the resulting candidate services, an interactive search method appears well suited. Indeed, interactive search methods help to create a common ground and mutual learning between the computationally intelligent algorithm and the software engineer, such that the resulting solutions are reasonable and predictable (Ramirez et al., 2018). Furthermore, Ramirez et al. (2018) point out that interactive search can foster a level of comfort, acceptance and trust in the candidate solutions obtained. However, Ramirez et al. also state that reports of interactive search-based software engineering (iSBSE) to address the service identification problem are not readily available in the research literature.

The application of iSBSE to address problems in software engineering such as the service identification problem presents challenges, however. One key challenge is formulating a suitable solution representation and corresponding fitness measures to completely reflect the reality of the software engineer development activities (Ramirez et al., 2018). In this research, we formulate a solution representation that maps business process models to the corresponding candidate services. With respect to fitness measures, the use of iSBSE strikes a balance between the subjective and objective measures to allow software engineers to drive the trajectory of the search, while at the same time ensuring the feasibility and quality of the resulting SOA solutions. Furthermore, it is important that the implementation of the underlying search algorithm is sufficiently performant to prevent user fatigue as the software engineer interacts with the search offering qualitative preferences and evaluating candidate services.

This research investigates the potential of interactive metaheuristic search to enable the exploration and discovery of feasible and high-quality SOA solutions that satisfy the organisation's business process, such that any business-IT gap is reduced.

1.1. Background

A Business Service is a service that is delivered to business customers by business constituents. For example, banks provide financial services to their customers or retail stores deliver goods to the customers. On the other hand, in the context of service-oriented architecture (SOA), the IT service (i.e., can be called software service) is a term refers to software functionality or multiple functionalities that can be used by different customers for different purposes. The IT service has a mechanism to enable access to capabilities and database through special interface. These software services have constraints and polices that are specified by the standard service description. IT services are used to support the successful delivery of business services. Moreover, some business services can depend entirely on software services to fulfil their functionalities; for example, an online banking service. In this research, the aim is to identify the needed software services that are needed to build an IT system that satisfies the business services. The software services help to automate the business functionalities fully or partially. The process of finding these IT services from the business services is called “Software Service Identification” or in short “Service Identification”. The short term “Service” in this thesis point to the software services, therefore, deriving services or identifying services means software services.

Service Oriented Architecture (SOA) is an approach that is used to build IT system based upon the use of software services. SOA makes building and changing IT systems easier. Traditionally, an IT system meant piecing together a collection of hardware, software and networking, all these components were rigidly integrated, so implementing a change was difficult. With SOA, an IT system is built using easy to assemble, easily reconfigurable components, like building blocks. Each building block represents a business service that the business performs. So to implement a change, these building blocks (i.e., candidate software services) can be assembled in a different way to make something new.

A key component of the web services model is service discovery; which defines a process for finding service providers and retrieving service description documents. The concept of service discovery has been addressed by different solutions, for example, the Universal Description, Discovery and Integration (UDDI) specification. UDDI aims to address a subset of the service discovery requirements by using centralised service discovery model. Another example is the Web Service Inspection Language (WS-Inspection) which provides another related service discovery mechanism.

SOA is based upon the interaction between three key roles: service provider, service registry (i.e., or broker), and service consumer (i.e., or requestor). These roles interact using three main operations: publish, find and bind. The service provider creates software services based on business functionalities (e.g., online banking services), provides access to the web service, and publishes the full description of the software service in a service registry. Service consumer can find the service description in a service registry and bind to a service using the information in the description. Figure 1.1 presents a logical view of the SOA. In this conceptual view of web services architecture, a centralized location for storing service descriptions is provided by the web service architecture. UDDI registry is an example of this type of service registry.

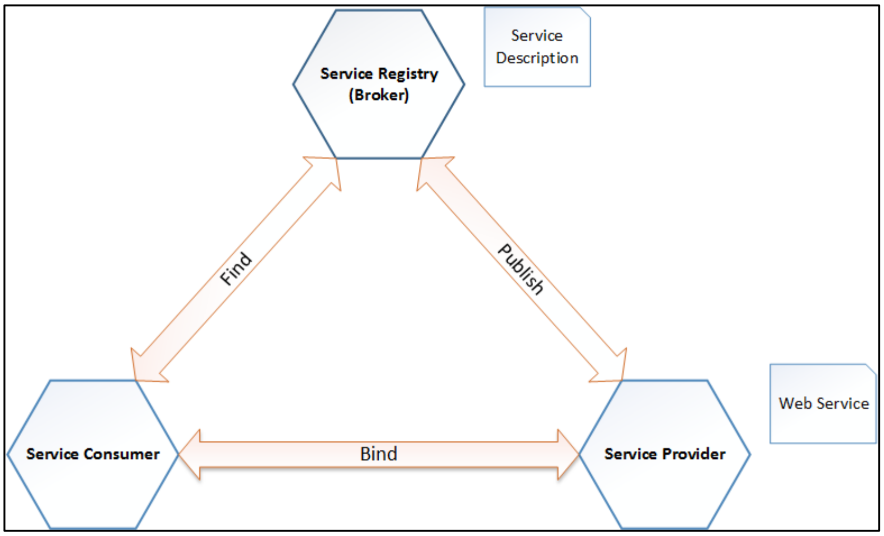

Figure 1.1: Service Oriented Architecture Adapted from (Voormann 2006), Licensed under CC-BY-SA 3.0

An example of using SOA for an organisation is the online banking. In which the bank (i.e., a provider of business banking service) can provide software services for online banking (e.g., make an online check for account balance). The online banking service can be published and used by bank clients (i.e., consumers). The software services for online banking represent a set of corresponding business services provided by the banking business. One web service may contain a set of functionalities, for example, online banking service can comprise different sub-services (e.g., check for balance, make a money transfer, and make a payment). Service identification investigates what business functionalities should be allocated in a web service and how these web services can interact with each other to satisfy the required business functionalities.

1.2. The Research Problem

Modelling the business context as a basis for software systems development has been widely investigated to reduce any gap between business and software systems (Vale et al., 2012). This business-IT alignment helps to mirror, support and automate the business (Odeh and Kamm, 2003). Agility of SOA development underpins business-IT alignment as it represents a product of the SOA flexibility (Zadeh et al., 2016). Owing to SOA's remarkable features such as flexibility, agility and modularity, SOA has received growing attention (Papazoglou et al., 2008; Yousef, 2011; Gu and Lago, 2010; Bianchini et al., 2013). SOA includes two key components: business and technology. Firstly, the business component focuses on business-aligned services to support the business-IT alignment. Secondly, the software aspect manages only the pre-existing technology in relation to services. Ignoring any aspect of the comprehensive guidelines leads to the creation of implementation difficulties that can weaken the realisation of its benefits (Kohlborn et al., 2009).

Software service identification is one of the most significant activities of the design and development of SOA solutions (Jamshidi et al., 2012). Although the identification of key abstractions (i.e., services) is one of the initial tasks in the modelling of an enterprise solution, software engineers can struggle to identify key abstractions to represent appropriate domain elements in a non-arbitrary (i.e., subjective or random) manner (Levi and Arsanjani, 2002). Nevertheless, it is very important to identify correct services, as errors at this stage are expensive and difficult to resolve, which reflects on the need for a robust SIM that derives feasible service solutions (Börner et al., 2012; Boerner and Goeken, 2009; Dijkman et al., 2008). Moreover, this emphasises the importance of not only identifying the right elements (i.e., effectiveness) but also identifying them in a productive manner (i.e., efficiency) (Arsanjani et al., 2008). Despite this, comprehensive systematic approaches capable of producing feasible SOA solutions with services rooted in business entities are not readily available in the research literature (Shashwat and Kumar, 2017; Zadeh et al., 2016; Alkkiomäki and Smolander, 2015).

A further key aspect that distinguishes existing SIMs is the delivery strategy. According to Huergo et al., 2014, Vale et al., 2012, and Gu and Lago, 2010, the delivery strategy can be a business-oriented (top-down), technical-oriented (bottom-up), or goal-service modelling (meet-in-the-middle) strategy. Top-down approaches focus on domain decomposition, whereas bottom-up approaches focus on analysing existing software and technology assets (Zadeh et al., 2016). The delivery strategy is identified according to the available input artefacts. For example, a SIMs that adopts a high-level artefact from the business domain as input (e.g., business processes or business goals) is considered as a top-down approach. Whereas, if the SIM has IT assets (e.g., databases, interfaces and legacy software systems) as input, it can be considered as a bottom-up approach. One of the key challenges to be considered is to identify the right services that not only satisfy current business needs, but also deal with anticipated changes (Frey et al., 2013). Huergo et al. (2014) observe that the majority of SIMs are technically-oriented and they focus on the technical domain. However, to ensure the business-IT alignment, mapping of the business process to the corresponding web services should be established (Cherbakov et al., 2005).

Although the adoption of a business-oriented method is more effective when performing a business-IT alignment, it is more challenging in terms of interactive search and automation. The descriptive and fuzzy nature of business models constrains SIMs, and as a consequence, just a few top-down SIMs aim to propose an interactive semi-automated method, whereas the majority of the existing top-down SIMs lack computational support (Zadeh et al., 2016). In contrast, it has been observed that most of the automated SIMs use a bottom-up delivery strategy and have some technical assets as input (Azevedo et al., 2014; Vale et al., 2012). However, in order to develop interactive and business-oriented computational support for service identification, business entities need to be quantified (i.e., have an 'input data preparation' phase) to prepare the inputs before starting the service identification activities.

A significant consideration to be taken while developing a SIM is the quality assessment of both the resulting services and the service identification approach itself. Many SIMs lack comprehensive evaluation using empirical methods (e.g., case studies) (Huergo et al., 2014; Vale et al., 2012). In this case, according to Bianchini et al. (2009), the resulting SIM can be more theoretical as it does not show sufficient evidence to confirm the feasibility of the resulting services. In addition, there is no guarantee that high-quality services that adhere to SOA principles such as low coupling and high cohesion will be produced (Bianchini et al., 2014; Jamshidi et al., 2008). A comprehensive framework that comprises all the phases of service identification with detailed guidelines that are also validated using a robust empirical method remains to be developed (Shashwat and Kumar, 2017; Zadeh et al., 2016).

Applying an interactive SBSE (iSBSE) technique to address the service identification problem offers a promising solution, because the resulting solutions can engender genuine acceptance by the software engineer, and the software engineer and the computationally intelligent framework can collaborate with common purpose to achieve a set of agreed goals (Klien et al., 2004). The responsive nature of the interactive search allows the participant to evaluate the results during the search process (i.e., not before nor after) such that the search algorithm can use the human preference to learn from the feedback and enhance the outcomes (Aljawawdeh, Simons and Odeh, 2015). Using an interactive search technique is a challenging task because it is difficult to create an optimisation model for the service identification to reflect all aspects required by the real decision makers (Meignan et al., 2015). This requires the discovery of an appropriate representation technique along with the right fitness measures to provide an effective and efficient search experience (Meignan et al., 2015). Furthermore, formulating an appropriate representation and fitness function can be more challenging if the criterion of accepting solutions cannot be expressed explicitly (Guindon, 1990). Hence the use of an interactive search technique is anticipated to bring together computationally intelligent search and software engineer preference in order to raise the level of acceptance and trust of the resulting solutions. In addition, the importance of a mutual understanding between the software engineer and the search approach is emphasised in order to be able to deliver a reliable interactive search method (Klien et al., 2004).

Finally, ensuring satisfactory performance of the search process (i.e., in terms of time required to derive candidate services) is another challenge. This requires ensuring the efficiency and performance of the search, especially with increasing scale and complexity of problem instances (Eiben and Smith, 2015). Unsatisfactory search performance can lead to user fatigue and so ineffective interaction between the software engineer and the search algorithm, which may negatively affect the trajectory of search.

To summarise the key challenges that face the existing SIMs:

- SIMs should cover all the phases and activities of service identification process. For example, a comprehensive SIM covers all the phases of input preparation, service identification, and output refinement rather than managing the service identification phase alone.
- Automation with the metaheuristic search should support the software engineer to reduce the heavy reliance on the software engineer in manual service identification and reduce the cognitive load on the software engineer. Providing automated computationally-intelligent support is challenging because managing top-down input artefacts is complex as these input artefacts have a descriptive nature.
- Interactive search should enrich the quality of the SIM as well as the resulting services. In addition, having the software engineer 'in-the-loop' adds another challenge in the incorporation of human evaluation of candidate solutions, and capturing human qualitative preferences, both explicit and implicit.
- The gap in using comprehensive quality assessment to evaluate the quality of both the resulting candidate services and the service identification approach itself. There is a gap in using validated empirical methods (e.g., case study) to evaluate the quality of SIMs.

This research investigates iSBSE to address these challenges. The main goal of this research is to introduce a novel comprehensive interactive framework for service identification. This framework should cover all the phases of service identification and utilises the human implicit knowledge and explicit experience to enrich the quality of the resulting services. The feasibility of the resulting solutions and conformance to SOA principles are important aspects that also should be satisfied. Significantly, quantification of service identification metrics should enable the evaluation of the proposed framework as well as the resulting candidate SOA solutions. Finally, sufficient evidence of the successful design and implementation of this sophisticated method should be provided and validated using a representative real-life case study.

1.3. Research Motivation

The motivation of this research stems from the importance of producing a comprehensive framework that derives effective candidate services from role-based BPMs. We are firstly motivated to develop this framework to provide a practical example of using the interactive search to address and resolve a practical problem in software engineering that contributes to the field of SOA and Healthcare. Secondly, the requirements of this problem endorse using a triangulation of evaluation methods (i.e., a combination of objective and subjective techniques) to examine the quality of the resulting services as well as the service identification approach itself. Thus, we are motivated to investigate the extent to which interactive metaheuristic search can satisfy this objective. The interactive search means that there is a responsive process between the software engineer and an intelligent search algorithm **during the search process** (i.e., not before or after), which helps the software engineer to drive the trajectory of the search (Aljawawdeh, Simons and Odeh, 2015).

However, there is a strong incentive to initiate a service identification framework that overcomes some limitations of other SIMs such that all the activities of service identification can be handled and automated using an interactive metaheuristic search technique. This results in increasing the quality of the outcomes and reducing the development costs (i.e., time and money). The key characteristics of the required SIM include the following aspects:

a. **Comprehensive**: Needs to include all the phases of service identification not just the mapping between the business elements and the corresponding service components.
b. **Interactive**: It utilises the human interactive preference (Aljawawdeh, Simons and Odeh, 2015) to enrich the quality of the resulting services. Moreover, interaction with computational intelligence creates a mutual ground between the human and computationally intelligent tool.
c. **Appropriate Level of Automation**: utilising automated computational intelligent tools is one of the main characteristics that support bridging the gap between the business and software systems (Odeh and Kamm, 2003). Computational intelligence can be used to support human to satisfy difficult tasks by automating and managing complicated activities, is useful and has many advantages. For example, automating the complicated activities minimises the heavy reliance on software engineers, reduces the cost (i.e., time, money, errors, etc.) of constructing software systems. However, this research aims to utilise the human knowledge and experience to enrich the quality of the outcomes, but not to exclude the human from the loop. In contrast, one of the limitations of many top-down SIMs is to automate the data preparation phase that works directly with the input business artefacts. These activities of the service identification, including the data preparation and the resulting service refinement phases, can be fully automated. The automation of these phases helps to accelerate the development to keep up with the quick changes in business requirements.
d. **Validated method**: The SIM should combine quantitative and qualitative methods to assess the quality of the resulting solutions. The quantitative measures are measured using mathematical formulas, whereas, the subjective feedback of experts is utilised to enrich the quality of the outcomes. A representative case study should be adopted to fully evaluate the novel SIM.

1.4. Research Aim and Objectives

This research aims to contribute to the SOA paradigm by using an interactive metaheuristic search to derive SOA services from BPMN, to automate, simplify, and enrich the service identification process. To address this aim in light of the gap analysis presented in Chapter 2, the following four objectives have been identified:

1) Investigate for an appropriate framework to derive services from business process models that smoothly map business processes to software services and to enable the generalisation of the mapping approach.
2) Investigate the appropriateness of various representations for effective metaheuristic search and the corresponding objective quality fitness measures for SOA. This objective necessitates the development of a suitable representation that is a natural fit for the mapping context from the business process to services. Moreover, this objective needs to define a set of fitness measures to quantify and evaluate the resulting services in order to evaluate their quality.
3) Investigate novel ways of adopting appropriate metaheuristic search algorithms. This requires evaluating and selecting an appropriate search algorithm and then conducting a set of experiments to explore and exploit the search space to find effective and efficient candidate SOA solutions.
4) Explore novel mechanisms in which the software engineer preferences and implicit knowledge and experience can be exploited within an interactive, human 'in-the-loop' search.

The above aim and associated objectives have been established in anticipation of producing significant novel contributions to knowledge, and have formed the basis for the development of a novel framework employing an iSBSE technique to bridge the gap between the business process and the software system-to-be context, paving the way for further research directions in the future.

1.5. The Research Hypothesis and Associated Questions

This research hypothesises that:

"***Using interactive metaheuristic search facilitates the derivation of candidate software services from role-based business process models (BPMs), and in particular BPMN process models***".

Using BPMN in this research to identify the organisation's business workflow has many advantages. For example, it provides an intuitive and easy way for non-expert users in BPM to understand the notation (Object Management Group, 2011). Furthermore, it represents the semantics of complex processes in a graphical form. More importantly, it has good tool support, e.g., BPMN-Modeller (Object Management Group, 2013) and Camunda (Fernandez, 2013). The BPMN reflects the way in which business processes take place in the organisation and shows the chronological sequence of these activities. This means that building an IT-system for the organisation should reflect the business functionalities as services. Therefore, BPMN is used as the starting point to derive SOA software services to achieve the business-IT alignment by mapping BPMN activities to the corresponding service components.

Human preference will be utilised with a combination of quantitative measures to assess the quality of the services such that good-enough (i.e., with low coupling and high cohesion values) solutions will be constructed. This research makes a novel contribution to deriving participants' preference values and combine these values with other quantitative fitness measures in order to evaluate the derived candidate SOA solutions; this will help to enrich the quality of the outcomes.

Based on the above hypothesis and discussion, the following research questions have been formulated with the aim of supporting the development of a methodological approach to assist in proving or refute the above hypothesis:

- **RQ1**: To what extent can role-based business process models such as BPMN 2.0 models be mapped to services following Service Oriented Architecture (SOA) principles?

- **RQ2**: In what way can SOA services be best implemented for metaheuristic search?
- **RQ3**: Can the services solution space be effectively and efficiently explored and exploited in order to derive candidate SOA solutions from BPMN 2.0 models?
- **RQ4**: What is the impact of the human interactive preference on the search outcomes?

1.6. Summary of Research Contributions

The contributions to the knowledge of this research can be summarised as follows:

1. The development of a comprehensive layered interactive framework that is named BPMiSearch, that derives software candidate services from business process models.
2. The formulation of service identification problem to be implemented in an appropriate format to enable the metaheuristic search using a genetic algorithm. This includes preparing the input BPMN models in the right level of granularity to create Search Space Elements (i.e., the basic building blocks of the search space). Then encode these elements in an appropriate way to produce the initial population that is used by the search engine. The novel work in this part is the utilisation of the systematic approach to represent elements derived from BPMN and prepare them to be evolved using an evolutionary computing technique.
3. Arriving at a good-enough combination of genetic parameters (e.g., crossover and mutation probabilities) that enable the exploration and exploitation of the problem search space to arrive at effective and efficient candidate solutions.
4. The development of a novel mechanism to integrate the human interactive preference to drive the trajectory of the search. This integration helped to enrich the quality of the resulting solutions by increasing the fitness values and level of satisfaction of domain experts.

5. The adoption of Design Science Research Methodology (DSRM) to satisfy the objectives of this research. In addition, applying the framework to derive candidate software services for the Cancer Care using a case study.

1.7. Thesis Roadmap and Structure

This thesis consists of seven chapters. Figure 1.2 presents a roadmap for the thesis that shows how the research questions are answered, and how the research hypothesis is addressed. The diagram summarises the main sections of the thesis chapters. **Chapter One** introduces the research problem, identifies the rationale behind this research, and highlights motivations for deriving services from business processes. In addition, this chapter discloses the research aim and objectives, and accordingly, the research hypothesis and the associated questions are presented. This is followed by **Chapter Two** which provides a state-of-the-art literature review in the domain of service identification. This is followed by a thorough analysis of the relevant studies in the service identification to identify the gaps in the literature. In addition, the chapter presents the use of metaheuristic search to derive services from business process, and the potential of using interactive search-based method for service identification. The chapter concludes by summarising the gaps and the need to develop a research design methodology that supports covering these gaps with reference to the interactive search. **Chapter Three** provides a full overview of the research methodology adopted in this research as well as the iterative research design and the case study. **Chapter Four** presents the first DSRM iteration, in which the basis of the layered service identification framework is introduced. In addition, this chapter presents the representation of the SOA services in the search space with practical sample solutions produced by domain experts. **Chapter Five** shows the second DSRM iteration, in which the adoption of the SBSE technique is discussed in detail. Moreover, it demonstrates the search-based method using a representative case study and then evaluates it using different techniques. This chapter reveals the experiments that examine the efficiency and effectiveness of the search-based method and paves the way for using the interactive search method. **Chapter Six** presents the novel interactive search-based framework for service identification (i.e., BPMiSearch). In this chapter, the interactive technique is discussed, followed by an experimental demonstration using domain experts. This chapter illustrates a triangulation of evaluation methods with a comprehensive analysis of the interactive search-based framework in comparison with other techniques. It concludes by presenting the influence of using the human interactive preference on the search outcomes. Finally, a bottom-up approach to answering the research hypothesis, a summary of research outcomes, findings, and suggestions for future directions are presented in **Chapter Seven**.

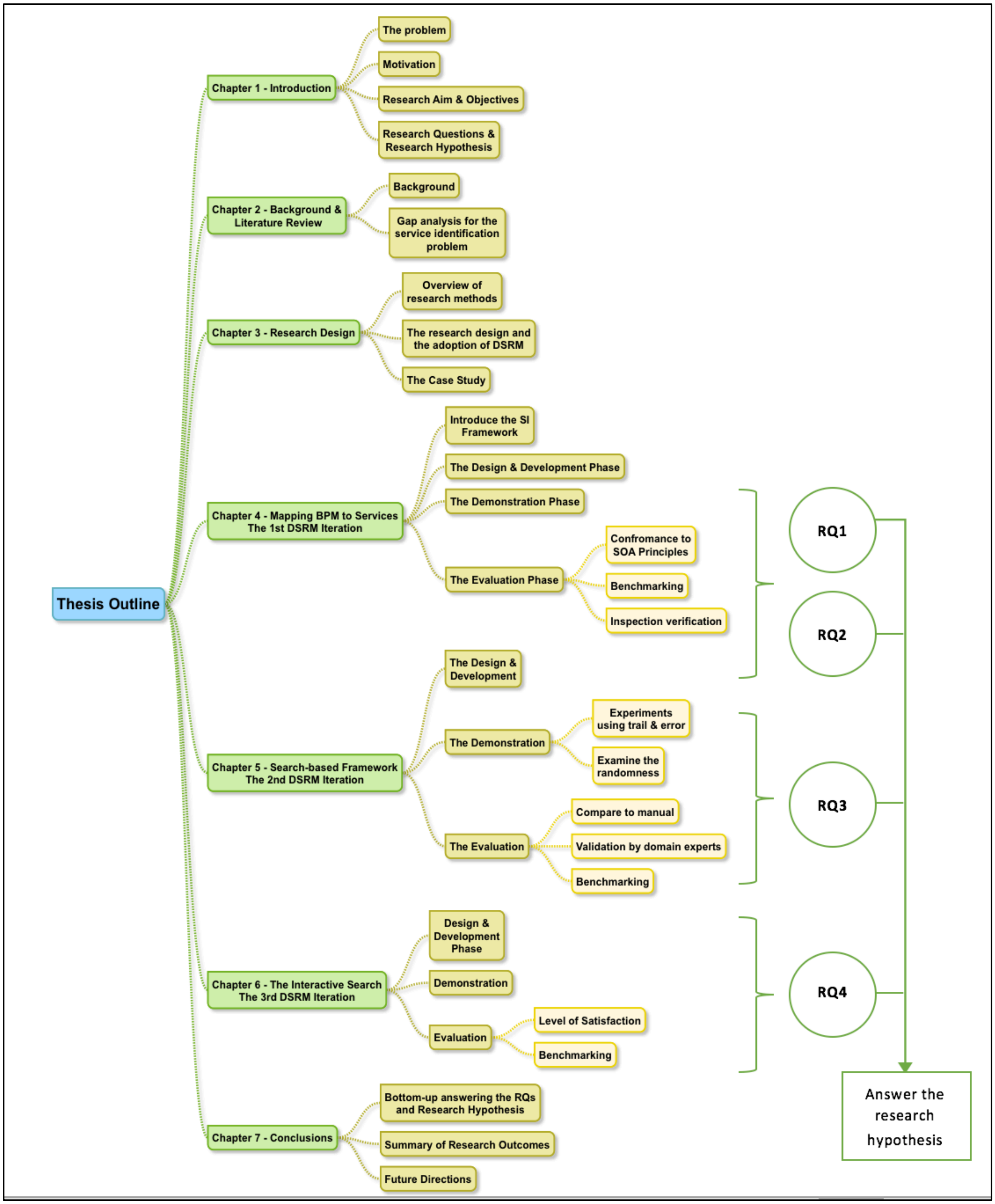

Figure 1.2: Thesis Roadmap

1.8. List of Publications

1. Aljawawdeh, H. J., Simons, C. L., & Odeh, M. (2015, July). Metaheuristic design pattern: Preference. In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation (pp. 1257-1260). ACM.
2. Aljawawdeh, H. J., Odeh, M., Simons, C. L., Lebzo, N. (2018 November). A Metaheuristic Search Framework to Derive Cancer Care Services from Business Process Models. Cancer Care Informatics 2018. [Submitted, in review]
3. Aljawawdeh, H. J., Odeh, Y., Odeh, M., Simons, C. L., Tbakhi, A. (2018 November). A Metaheuristic Framework to Derive Services for Cell Therapy and Applied Genomics from Business Process, A Case Study. Cancer Care Informatics 2018. [In preparation]
4. Aljawawdeh, H.J., Odeh, M., Simons, C. (2018). An Interactive Search-based Framework to Identify SOA Services from Business Process Models. Business Process Architectures and Service-Oriented Environment. Information Software Technology Journal. [In preparation].

Chapter 2
Background and Literature Review

2.1. Background

Service Oriented Architecture (SOA) has gained considerable attention in facing today's business and software challenges resulting from rapid changes in business environments and user expectations. These challenges include business challenges such as agility and flexibility, and software challenges such as reusability and integration (Zadeh et al., 2016). Therefore, transformation to SOA is a goal for most enterprises (Yousef, 2010).

Service identification is one of the key activities and an example of best practice in research that aims to investigate the SOA development lifecycle. The contemporary Service Identification Methods (SIMs) can suffer from critical limitations; for example, are not able to handle all the activities of service identification, so some important aspects are neglect (e.g., input data preparation). Other approaches adopt a very complicated identification criterion, which requires deep analysis of all sets of business process models in order to construct feasible solutions that adhere to SOA principles (Yousef, 2010). In contrast, some SIMs are too simple to fulfil the key principles of SOA development (Papazoglou et al., 2007). The possibility to automate the activities of service identification and reduce human fatigue is another important issue. Failing to automate the activities of service identification is another shortcoming exhibited in the contemporary SIMs.

In addition to the above limitations, it appears that there is a lack of a SIM that fully evaluates the quality of the resulting outcomes to ensure the feasibility of these services. Consequently, poor evaluation of candidate services is another limitation exhibited.

The Business Process (BP) is important to capture several aspects, such as business functions, inventory of existing software systems, identification of business needs, and objectives and quality attributes (Yousef, 2010). BP can be classified into Business Process Architecture (i.e., BPA) and Business Process Modelling (i.e., BPM). Both BPA and BPM provide a good comprehension of the "as-is" environment. In order to achieve business-IT alignment, it is important to take into consideration the "as-is" situations, such that candidate services correspond to the business domain entities (Zadeh et al., 2016).

To explicitly distinguish between the two types of business process, BPMs represent the detailed step-by-step activities used to satisfy business goals and objectives (Havey, 2005). However, BPAs identify all the processes in an organisation and show the relationships between them at a higher level of abstraction (Ould, 1995). In order to derive candidate services that correspond to the organisation's business, it is required to have a detailed representation of the organisation's business processes. Although it is focused on maintaining the process architecture prior to developing a software system to support the organisations' needs (Harmon, 2003), mapping the business activities to the corresponding service components requires maintaining the BPM rather than BPA. Using only BPAs is not sufficient to perform the mapping between the business to SOA candidate services because the detailed activities and business elements are not presented in these models. Based on this discussion, this research adopts BPM to perform the mapping.

The existence of different business process modelling approaches and a set of associated process modelling languages (Khan et al., 2011) makes it challenging to select the appropriate BPM to perform the mapping between the business context and the software services. BPM approaches can be classified into different categories based on their modelling perspectives and characteristics (Giaglis, 2001; Yousef, 2010). These categories are presented as follows:

(i) Role-based modelling approaches: the main constructs involve a set of roles, where each role comprises a set of related roles and activities that interact with each other (Cho, Kim and Hie, 1998). A role can interact with other roles using different techniques such as messages or events (Khan, 2009). The role-based approach presents the organisation's detailed activities connected and ordered in a chronological sequence. Examples of the role-based approaches are Role Activity Diagrams (RAD) (Badica et al., 2003), Business Process Modelling and Notation (BPMN) (Dijkman et al., 2011), and the Unified Modelling Language (UML) (Pooley and King, 1999).

(ii) Functional-based modelling approaches: using a functional business orientation, the whole system performs as a set of 'sub-systems' each performing a specific function (e.g., sales and production). This approach helps each small sub-system to specialise in a specific functionality and makes it easier to manage as it is less complex. However, the main drawback is that producing a common outcome requires collaboration between the small sub-systems in the organisation, which is difficult since the functionalities are not transparently divided amongst the different functional units (Reijers, 2003). An example of a function-based approach is the Integration Definition (IDEF) which represents a family of modelling languages (Kim et al., 2003).

(iii) Activity-based modelling approaches: the main entity in this approach are the activities that construct the approach. Typically, these activities are connected sequentially and conditional gateways control the flow. Examples of the activity-based approaches include Business Process Execution Language (BPEL) and Unified Modelling Language Activity Diagrams (UML AD) (Rodrigues, 2000).

(iv) Goal-based modelling approaches (Khan, 2009): in order to have an effective process execution (i.e., includes aligning individual employees' actions with business objectives, and supporting decision making by documenting the organisation performance), the goal-based business approach captures the business goals and the actors (Kueng and Kawalek, 1997). Using this approach, it is anticipated that organisations will be helped to meet their goals. An example of this approach is the Strategic Goal Role Model (SGRM) and Operational Process Model (OPM) (Jureta et al., 2005). The conflicting goals and confusing constraints with goals are considered to be the key limitations of this approach. Moreover, this approach is not capable of modelling the workflow activities and processes, which is a limitation with regard to achieving business-IT alignment (Greenwood, 2008; Nurcan et al., 2005).

The selection of an appropriate BPM type depends on a set of characteristics provided by the BPM. These include presenting a clear purpose and role. Moreover, the BPM should reveal the relationships between different roles in a chronological sequence which helps to build the binding details between services. In addition, graphical annotations of the selected BPM is important to allow participants from different domains to comprehend the processes and interact with the search. Furthermore, the availability of a real-life case study (i.e., Cancer Care and Registration) with the entire models in BPMN supports using this notation derive candidate services. This case study has been used by a service identification framework (i.e., named BPAOntoSOA) to derive candidate services. Using these models specifically can support the evaluation of this research by holding a comparison with the BPAOntoSOA.

Based on the characteristics of the existing BPM approaches, the role-based approach has been adopted in this research. The key advantage of this approach is that it allows for the underlining and clear definition of the responsibilities of each role (i.e., actor), which supports building a clear understanding of the business processes and helps to improve the entire process (Saidani and Nurcan, 2006). In addition, the role-based methods with the corresponding languages (e.g., BPMN) are flexible enough to handle the changes in the organisational, functional and operational business requirements (Gligor et al., 1998). The basic business process elements can be derived from a BPMN metamodel where relationships between different elements such as Pool, Event and Activity are presented.

BPMN provides a graphical notation that is readily understandable by technical developers as well as business analysts and stakeholders. Flow Objects (i.e., Activities, Events, and Gateways) are the main graphical elements that are used to define the behaviour of a business process. Elements are connected using Sequence Flows, Message Flows, or Associations. Moreover, modelling elements are mainly grouped using Pools and Lanes. Table 2.1 presents the basic BPMN elements derived from the BPMN 2.0 metamodel.

Table 2.1: Basic BPMN Elements (Object Management Group, 2011)

Element type	Remarks
Event	There are three types of events: start, intermediate, and end.
Activity	An Activity is a generic term for work that a company performs in a Process. The types of activities that are a part of a process model are: sub-process and task.
Gateway	Types of gateway include: exclusive, inclusive, parallel, complex, and event-based.
Sequence Flow	A sequence flow is used to show the order that activities will be performed in a process.
Message Flow	A message flow is used to show the flow of messages between two Participants (i.e., sender and receiver).
Data Object	It can be either Data Input or Data Output.
Message	A message represents the contents of a communication between two participants.
Association	An association links information and artefacts with BPMN graphical elements.
Pool	A graphical container for partitioning a set of Activities from other pools.
Lane	A lane is a sub-partition within a process, sometimes within a pool, and used to organise activities.

SBSE techniques have been applied across different aspects of the software systems development lifecycle, such as requirements analysis and scheduling, design, programming, and testing (Harman et al., 2012). Many of these techniques use search to optimise candidate solutions among vast and multi-dimensional search spaces (Simons and Parmee, 2011). However, the emphasis of the SBSE techniques is to enable the exploration and exploitation of the search space with a view to increasing the understanding of potential solutions. The search-based techniques in such cases support making the right decisions when the number of candidate solutions is beyond human comprehension.

Genetic algorithms (GAs) are a subset of Evolutionary Algorithms (EAs) which represents a metaheuristic inspired by the natural selection. Utilising a GA requires two components; a representation (along with the genetic operators) and a fitness function. GA adopts two operators for reproduction; (i) crossover operator that mixes two parent solutions to produce offspring solutions (ii) mutation operator that helps to produce creative solutions and prevent the search from being trapped in local optima. The fitness function is used to measure the quality of population individual solutions (known here as genes) in order to prioritise them (Mohan, 2018). The genes are constantly measured at each iteration of the search to determine the solutions that would live to the next generation based on the fitness values. Fittest solutions are more likely will be selected to breed the new generation of solutions.

Single-objective algorithms are used to tackle problems that have one objective and can be used to obtain one solution at a time. Fitness function can be measured in different ways, for example, weighted-sum fitness function, in which different design metrics (e.g., coupling and cohesion) are combined in one value based on the weight of each design metric. In contrast, multi-objective algorithms are used to manage problems that have multiple constraints or objectives and may involve more than one objective function to be optimised simultaneously (Mohan, 2018). MOEA can is capable to produce multiple solutions, which gives the participant the choice to select between them. However, multi-objective algorithms have some limitations; (i) extra processing is needed to manage the various objectives, this means that more time is needed to produce solutions, (ii) selecting the final solution is subject to the user’s posterior decision on which objective fitness function has the highest priority. One of the most popular MOEA algorithms is the non-dominated genetic algorithm NSGA-II (Deb, Pratap and Agarwal., 2002). NSGA uses a fast non-dominated sorting approach (i.e., performs fewer comparisons between solutions) in order simultaneously optimise each objective without being dominated by any other solution (Yusoff et al., 2011).

An enhancement to multi-objective algorithms are the many-objective algorithms that aim to manage more than three objectives (i.e., most practitioners agree to a maximum of 10-15 objectives but they can reach up to 20 objectives). When using many objectives with MOEA techniques, calculating the fitness measures becomes more computationally expensive due to a large number of objectives. In addition, the impact of the genetic operators (e.g., crossover) may also become inefficient. Therefore, different alternatives can be used to address these issues. An enhancement version of NSGA-II is the NSGA-III by Deb and Jain (2014). This algorithm is designed to manage many objectives by changing the selection mechanisms. NSGA-III uses the non-dominated functionality of it originator algorithm, but with an enhanced selection technique to improve calculating the crowding distance calculations in the previous version of the algorithm (NSGA-II).

Since this research aims to adopt an interactive metaheuristic search algorithm, a single-objective genetic algorithm will be selected for this purpose. The reasons for that can be summarised as follows;

a) Multi- or many- objective algorithms are computationally expensive, and processing the data of a large case study requires a long time. Having the human-in-the-loop becomes difficult as calculating solutions need a long time.
b) Single-objective algorithms can be designed to be more sensitive for human interaction.
c) MOEA requires users to select a solution among a set of potential solutions based on the importance of the design metrics (i.e., after the search), however, this research aims to focus on the interactive

preference (i.e., during the search) rather than posterior preference. Weighted-sum aggregation function can be used to overcome this problem.

For these reasons, a single-objective algorithm with a weighted-summation fitness function will be used to aggregate a combination of design metrics.

The interactive search builds on this emphasis by placing the software engineer directly “in-the-loop” of the search (Ramirez et al., 2018). Typically, the interactive search combines computational quantitative software fitness metrics with human qualitative evaluation to jointly steer the trajectory of the search (Simons et al., 2014). Exploiting human qualitative feedback is significant in providing a mechanism to explore trade-off judgments among competing criteria. The human preference can be classified based on the nature (i.e., implicit or explicit) or timing (i.e., priori, posteriori, or interactive) (Aljawawdeh, Simons and Odeh, 2015). Knowledge and experience play an important role in addressing different problems where interactive preference has been used to enhance the search performance (Ramirez et al., 2018).

The characteristics of such iSBSE techniques seem ideally suited to assisting software engineers with mapping the business context to software system models and architectures. Specifically, the interactive search properly to addresses the service identification problem, taking account of software engineer’s knowledge and experience; which is considered a novel research contribution.

This chapter sheds light on the main aspects of service identification in the literature. It mainly focuses on aspects of the comprehensiveness of the SIMs (i.e., handling all the phases of service identification), automation or (i.e., or semi-automation when having the human in-the-loop), techniques (i.e., this includes using search methods and interactive context), and the quality assessment of services. Studying these is important in order to perform a thorough gap analysis that reveals the shortcomings of other SIMs. The design of this research has been prepared in the building of the outcomes of this chapter.

The second part of this chapter focuses on studying the service identification problem by selecting the most relevant using a specific criterion. The selected studies will be examined using different perspectives to identify the existing gaps.

2.2. Service Identification Problem

Classifying service identification methods according to the delivery strategy results in two significantly different types of methods: business oriented (i.e., top-down) approaches, or technical oriented (i.e., bottom-up) approaches (Huergo et al., 2014; Gu and Lago, 2010). Some methods adopt a mix of both top-down and bottom-up approaches: these are referred to as meet-in-the-middle (Börner, Goeken and Rabhi, 2012). The main difference between the two categories is the origin of the service portfolio (i.e., technical-oriented or business-oriented). Business-oriented methods correspond to different types of entities in the business domain such as business process (i.e., as-is), business goals (i.e., to-be), use cases, and business requirements (Zadeh et al., 2016). However, technical-oriented methods typically rely on technical assets such as databases, existing legacy system source codes, and other technical entities (Vale et al., 2012).

Gap analysis begins by comparing the proposed SIM with contemporary SIMs that adopt the same delivery strategy. Gap analysis is important in being able to identify the missing functionalities in the current SIMs such that the characteristics of the proposed SIM closes these gaps (Kohlborn et al., 2009).

Although the majority of SIMs focus on the technical domain (Vale et al., 2012; Gu and Lago, 2010), business-oriented methods are more effective in achieving the business-IT alignment (Zadeh et al., 2016). The scope of this literature review is to study the relevant publications related to methods that derive SOA services from high abstract level artefacts (e.g., business processes, business goals, and business requirements). Therefore, the selection criterion of the relevant studies is based on the delivery strategy, such that all the studies clearly use a high abstract level input and construct candidate services. The technique should be explicitly presented as well as any tool support.

Based on a critical analysis of the existing top-down SIMs, the analysis of the relevant studies will consider important aspects that shed light on these gaps. To facilitate the critical gap analysis, the selected relevant studies will be examined based on the following aspects: (i) the technique and use of human preference (ii) the automation, and (iii) the quality assessment of candidate services. The selection of the most relevant SIMs and the analysis of the significant issues are presented in the following sections.

2.2.1. Selection of Relevant SIMs

This section applies the criteria of interest (i.e., see Section 2.2) in order to select the relevant studies in the service identification that are relevant to this research project, so as to perform a gap analysis in the next sections. Table 2.2 presents the relevant studies that satisfy the inclusion criteria. The table shows the SIM number, author, year of publication, and the title of the study, sorted by year of publication.

Table 2.2: List of relevant studies in the service identification sorted by publication year

SIM	Author	Year	Title
1	(Khoshnevis and Shams, 2017)	2017	Automating identification of services and their variability for product lines using NSGA-II.
2	(Zadeh, et al., 2016)	2016	Automated service identification framework (ASIF).
3	(Ebrahimifard, et al., 2016)	2016	Mapping BPMN 2.0 Choreography to WS-CDL: A Systematic Method.
4	(Al-Thuhli, et al., 2015)	2015	Migrating social business process to SOA.
5	(Lima & Huacarpuma, 2015)	2015	A Methodology for Identifying Candidate Services and Compositions.
6	(Bianchini, et al., 2014)	2014	Service identification in inter-organizational process design.
7	(Azevedo et al., 2014)	2014	A method for bridging the gap between business process models and services.
8	(Yousef et al., 2014)	2014	Extracting SOA Candidate Software Services from an Organization’s Object-Oriented Models.
9	(Birkmeier, et al., 2013)	2013	Alignment of business and its architectures in the German federal government: a systematic method to identify services from business processes.

10	(Jamshidi, et al., 2012)	2012	An automated service identification method.
11	(Leopold & Mendling, 2012)	2012	Automatic derivation of service candidates from business process model repositories.
12	(Guan, et al., 2012)	2012	A network topology clustering algorithm for service identification.
13	(Kazemi, et al., 2011)	2011	A genetic algorithm-based approach to service identification.
14	(Yousef, 2010)	2010	BPAOntoSOA: A semantically enriched framework for deriving SOA candidate software services from Riva-based business process architecture.
15	(Kohlborn et al., 2009)	2009	Identification and analysis of business and software services—a consolidated approach.
16	(Weller, et al., 2009)	2009	Meet the challenge in service identification: A ratio-based approach.
17	(Bianchini, et al., 2009)	2009	A methodology to enable inter-organizational process design through web services.
18	(Azevedo, et al., 2009)	2009	A Method for Service Identification from Business Process Models in an SOA Approach.
19	(Kim & Doh, 2009)	2009	Formal identification of right-grained services for service-oriented modelling.
20	(Shirazi et al, 2009)	2009	A combinational approach to service identification in SOA.
21	(Ma et al., 2009)	2009	Evaluating service identification with design metrics on business process decomposition.
22	(Dwivedi & Kulkarni, 2008)	2008	A model-driven service identification approach for process centric systems.
23	(Jamshidi, et al., 2008)	2008	To establish an enterprise service model from an enterprise business model.
24	(Mani, et al., 2008)	2008	Using user interface design to enhance service identification.
25	(Papazoglou and Heuvel, 2007)	2007	Business process development life cycle methodology.
26	(Klose, et al., 2007)	2007	Identification of Services - A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning.
27	(Inaganti & Behara, 2007)	2007	Service identification: BPM and SOA handshake.
28	(Wang, et al., 2005)	2005	Normal forms and normalized design method for business service.
29	(Kaabi et al., 2004)	2004	Eliciting service composition in a goal driven manner.

Table 2.2 contains 29 studies that are considered to be the relevant studies based on the review protocol (See Section 2.2). The first contribution within the field was published in 2004 (Kaabi et al., 2004), but recent studies have been published as well (Khoshnevis and Shams, 2017; Zadeh et al., 2016; Ebrahimifard et al., 2016), which indicates that the field is still attractive and gaps in the field still exist. It is concluded

that the development of a new SIM, or improving the existing SIMs to fill the gaps in the service identification is still a realistic objective to be achieved by researchers and practitioners.

The strength of these key studies is the capability of producing business-related services which support the business-IT alignment (Huergo et al., 2014; Gu and Lago, 2010). Some SIMs represent the business logic in the resulting Business Services (BS) (Zadeh et al., 2016; Bianchini et al., 2014; Birkmeier et al., 2013, Azevedo et al., 2014; Guan et al., 2012; Jamshidi et al., 2012; Yousef, 2010; Bianchini et al., 2009; Wang et al., 2005; Inaganti and Behara, 2007; Kim and Doh, 2009; Yousef et al., 2014), or Data Services (DS), in which business-centric entities are represented (Dwivedi and Kulkarni, 2008; Jamshidi et al., 2008; Mani et al., 2008; Azevedo et al., 2009). In contrast, just a few SIMs produce technical-related services, such as IT Services (IS) that represent different levels of technology (Lima and Huacarpuma, 2015; Jamshidi et al., 2012; Dwivedi and Kulkarni, 2008) and Web Services (WS) (Mani et al., 2008; Leopold and Mendling, 2012). Nevertheless, paying specific attention to the integration between the business and IT is important. However, the lack of SIMs that consider the two perspectives reveals a gap in supporting the IT services (Huergo et al., 2014; Gu and Lago, 2010).

2.2.2. Techniques of Service Identification

The choice of a service identification technique depends on different factors such as the input type, the objective of the SIM, and the availability of tool support (Huergo et al., 2014). This section sheds light on the different techniques used for service identification, reflects on the relationship between the technique and the objective of the SIM, and identifies the gaps in service identification techniques.

According to the latest survey in service identification (Huergo et al., 2014; Vale et al., 2012; Gu and Lago, 2010), techniques that have been adopted by the relevant SIMs can be classified into six categories, as follows:

(i) **Clustering algorithm**: SIMs (Zadeh et al., 2016; Yousef et al., 2014; Jamshidi et al., 2012, Yousef, 2010) use this approach to divide the population data or business process elements into a number of groups based on the similarities and relations between them. This technique has been widely used by SIMs whenever the purpose is to derive candidate solutions based on the similarities between the business process elements (Leopold and Mendling, 2012).

(ii) **Selection algorithm**: the selection algorithm is a technique of finding a specific smallest number in a list or array. Using this technique, the derivation of candidate solutions depends on different values such as maximum and minimum relationships between business process elements (Weller et al., 2009). The calculated values support the decision of the static algorithm to locate business process elements inside each service (Dwivedi and Kulkarni, 2008).

(iii) **Guidelines**: or heuristic technique is any approach to problem-solving that employs practical method to speed up the process of finding good-enough solutions. However, this technique does not guarantee to find optimal or logical solutions. This technique presents a set of heuristics and instructions to be accurately followed by the software engineer in order to map business process elements to the corresponding services. Heuristics can be mental shortcuts that ease the cognitive load of making a decision. Heuristics have been adopted widely by different researchers such as (Birkmeier et al., 2013; Klose et al., 2007; Inaganti and Behara, 2007; Azevedo et al., 2009).

(iv) **Value Analysis**: a technique to analyse values and relations of business process entities in order to derive candidate services (Huergo et al., 2014; Gu and Lago, 2010). This approach aims to improve the value of an item or process by understanding its constituent components and their associated costs. It then seeks to find enhancements to the components by either reducing their cost or increasing the value of the functions. In service identification, this technique helps software engineers make the right decisions by providing useful mathematical formulas to collect statistical data about the business process elements (Azevedo et al., 2014). Moreover, value analysis helps to mathematically examine the services using the desired formulas (Vale et al., 2012; Gu and Lago, 2010). The value analysis method has been adopted by a few relevant SIMs (Azevedo et al., 2014; Lima and Huacarpuma, 2015; Bianchini et al., 2009).

(v) **Ontology mapping**: Conceptual representation of the domain knowledge. This technique is based mainly on the definition of a set of rules (e.g., swirl rules) to identify business candidate services from the business process or requirements (Huergo et al., 2014; Yousef et al., 2014; Gu and Lago, 2010). In addition, the ontology annotation is used to semantically describe the resulting services (Teka et al., 2012). The recent surveys in service identification have classified ontology mapping as a stand-alone category.

(vi) **Search-based algorithms:** these techniques (e.g., genetic algorithms, ant colony optimisation and simulated annealing) convert the problem into a computational search problem that can be addressed using a metaheuristic. This requires the definition of the search space that comprises a set of candidate solutions (Khoshnevis and Shams, 2017). Search-based techniques have been used by a number of researchers to derive services from a given business process. Promising results have been revealed, such as the automation and quantitative evaluation of resulting candidate services (Khoshnevis and Shams, 2017; Jamshidi et al., 2012; Kazemi et al., 2011).

Table 2.3 presents the relevant studies classified by their techniques. Some SIMs have adopted multiple techniques; hence, they are classified within multiple categories (Yousef et al., 2014; Jamshidi et al., 2012; Bianchini et al., 2009).

Table 2.3: Techniques for service identification

Technique	**SIM**	**Count**
Clustering Algorithm	(Zadeh et al., 2016; Yousef, 2010; Jamshidi et al., 2012; Bianchini et al., 2014; Guan et al., 2012; Jamshidi et al., 2008; Mani et al., 2008; Azevedo et al., 2009; Kim and Doh, 2009; Leopold and Mendling, 2012; Yousef et al., 2014)	11

Selection Algorithm	(Dwivedi & Kulkarni, 2008; Weller et al., 2009)	2
Guidelines	(Birkmeier et al., 2013; Klose et al., 2007; Inaganti & Behara, 2007; Azevedo et al., 2009; Al-Thuhli et al., 2015; Ebrahimifard et al., 2016; Wang et al., 2005; Kaabi et al., 2004; Papazoglou and Heuvel, 2007; Kohlborn et al., 2009; Ma et al., 2009; Shirazi et al., 2009)	12
Value Analysis	(Azevedo L. G., et al., 2014; Lima & Huacarpuma, 2015; Bianchini et al., 2009)	3
Ontology Mapping	(Yousef et al., 2014; Yousef, 2010; Bianchini et al., 2009)	3
Search-based	(Khoshnevis and Shams, 2017; Kazemi et al., 2011; Jamshidi et al., 2012)	3

It has been observed that 13 out of 29 SIMs have adopted either clustering or selection algorithm. The majority of these SIMs aim to reduce the load on software engineers by automating service identification activities (Bianchini et al., 2014; Zadeh et al., 2016; Kim and Doh, 2009). The choice of a static clustering or selection algorithm supports this goal because these algorithms are simple and are easy to implement. A further advantage is the high performance obtained when using these techniques. However, the main disadvantage of using a static clustering or selection algorithm is the limited number of potential solutions that can be constructed. The static algorithm is capable of producing one solution for the problem based on the clustering criteria (Dwivedi and Kulkarni, 2008). In case the resulting solution is not reasonable or satisfactory, the algorithm becomes useless which increases the load on the software engineer to change the clustering or selection criteria to avoid errors in this stage (Yousef, 2010). Although the static clustering and selection algorithms have several advantages, because of their limitations, it appears likely that the static algorithms (i.e., clustering or selection) maybe considered not well-suited to effectively represent the service identification problem, unless they have been effectively optimised to start with.

With reference to guideline techniques, apparently, this category includes the largest number of studies compared to other techniques (i.e., 12 out of 29). Working with business-related input types with a descriptive and fuzzy nature can be simplified using guidelines and heuristics (Birkmeier et al., 2013; Klose et al., 2007). Generally, this technique explains the mechanism to derive candidate services using clear step-by-step heuristics. An advantage of using guidelines is that deriving candidate services can be achieved without being limited to a specific modelling language or approach (Klose et al., 2007). This feature helps to generalise the mapping process using similar input types and approaches, particularly as SIMs adopt guidelines which focus on the mapping technique and quality attributes rather than technical concerns such as automation (Azevedo et al., 2009; Ebrahimifard et al., 2016). Conversely, the main disadvantage of SIMs that adopt heuristics is the high reliance on human involvement to perform service identification activities (Wang et al, 2005). The automation is not a key objective of this technique, thus applying guidelines requires specialists with a high level of experience and knowledge in the business context as well as in SOA services (Ma et al., 2009; Shirazi et al., 2009). The complexity of guidelines in some cases results in producing theoretical SIMs (Yousef, 2010). For these reasons, it is concluded that this technique may be not an appropriate approach to be used in this research.

With regard to the value analysis technique, the key objective is to provide the user with statistical and analytical values that help the user to evaluate the candidate services and make the right decisions (e.g., number of business process elements, coupling and cohesion). The value analysis can be obtained from the input artefacts using mathematical formulas and calculations. Although these values are useful for evaluating the quality of resulting services since they enable accurate evaluations of the candidate solutions, this technique has the same shortcomings in common with the guidelines technique (e.g., complexity and lack of automation). In addition, it relies heavily on the human participant to acquire the required statistics, and then to complete the value analysis (Huergo et al., 2014). Thus, it appears likely that the value analysis technique is ill-suited for deriving the candidate SOA services while satisfying the objectives of this research.

The ontology mapping technique offers different advantages such as facilitating interoperability and machine reasoning (Gruber, 1995). Moreover, it also enables knowledge transfer by representing and organising the business process and resolving semantic heterogeneities (Yousef, 2010; Munir et al., 2012).

Although the ontology mapping technique has been classified as a special technique in recent surveys (Huergo et al., 2014; Vale et al., 2012; Gu and Lago, 2010), it has been observed that ontology mapping is not a stand-alone technique, as ontology-driven SIMs usually adopt an additional technique (e.g., a clustering algorithm or value analysis) to fulfil different service identification activities. For instance, Yousef et al., (2014) and Yousef, (2010) adopt a clustering algorithm, whereas Bianchini et al., (2009) use value analysis to examine the coupling and cohesion of services. Therefore, it is concluded that using the ontology mapping alone is ill-suited to effectively derive the candidate SOA services from the business process models and is also unsuitable with regard to satisfying the objectives of this research.

With regard to search-based techniques, they have been adopted by a few studies (i.e., 3 out of 29) to derive candidate services from the business process (Khoshnevis and Shams, 2017; Jamshidi et al., 2012; Kazemi et al., 2011). Using a search-based technique is advantageous; as it provides the capability to optimise good-enough solutions even with large-scale problems (Pitangueira et al., 2015; Harman, 2011). Moreover, metaheuristic search supports the automation of service identification activities and helps to generates multiple candidate solutions each time the search runs (i.e., even with the same set of parameters). This advantage offers the capability of selecting the best solution among a set of candidate good-enough solutions (Kazemi et al., 2011). In addition, it offers the possibility of repeating the search if the resulting candidate services are not satisfactory. However, using a metaheuristic search-based technique is challenging. Finding an appropriate representation technique and an accurate fitness measure are difficult tasks to perform (Eiben and Smith, 2003). Furthermore, a key challenge is to formulate the search-based components (i.e., representation and fitness function) to entirely reflect the actual experience of service identification activities (Meignan et al., 2015). This becomes a more challenging task if the fitness measure comprises a set of different and unrelated criteria for solution acceptance (Curtis, Kranser and Iscoe, 1988; Guindon, 1990).

Utilising the subjective human preference can facilitate addressing the limitations of the search-based techniques (Ramirez et al., 2018). There are different approaches to capture the human preference such as a priori (i.e., to set parameters and constraints for the subsequent search), interactively (i.e., to steer the trajectory of the search), or posteriori (i.e., to present the candidate solutions for human inspection after the search reaches termination) (Aljawawdeh, Simons and Odeh, 2015). The incorporation of the "human-in-the-loop" during the search to address problems in software engineering has been reported by an increasing number of researchers as iSBSE has attracted significant attention recently (Ramirez et al., 2018). Utilising iSBSE techniques supports the following: (i) it reflects the reality of the software engineer's development activity on the representation and fitness measure, (ii) it increases the acceptance and trust of the resulting solutions, and (iii) it helps to find a common ground between the software engineer and the computational intelligence such that the resulting solution looks like it is "human-written" (Jones, 2018). These benefits are anticipated to reduce the limitations of the search-based technique.

For these reasons, interactive search-based methods have been commonly applied to achieve better search performance (Ramirez et al., 2018). Interactive techniques have been used to address a variety of software engineering problems that include requirements (Tonella et al., 2013), graphical user interfaces (GUI) (Troiano et al., 2016), software product lines (El Yamany et al., 2014), object-oriented specifications (Simons et al., 2014), software architecture (Vathsavayi et al., 2013), code implementation (Axelsson et al., 2009), and testing and verification (Marculescu et al., 2016; Marculescu et al., 2015).

It has been noticed that there is a lack of research studies that incorporate iSBSE to address the service identification problem, given that some research attempts have previously explored the service identification using SBSE techniques (Khoshnevis et al., 2017; Jamshidi et al., 2012; Kazemi et al., 2011). It is concluded that using an interactive search suits the service identification problem and is anticipated to enrich the quality of the resulting candidate services. It is also anticipated that applying an interactive search for the service identification would result in constructing more effective and efficient solutions, and in addition, it would increase the level of acceptance of the search outcomes obtained (Ramirez et al., 2018). However, incorporating the human preference with the search exposes it to some challenges such as human fatigue (Simons et al., 2014). Therefore, different mechanisms have been applied in order to reduce this limitation (Shackelford and Simons, 2014; Simons et al., 2014). By utilising interactive search with the service identification, there is an expected gain in the advantages of the search-based techniques and it should reduce their limitations.

2.2.3. The Automation

The level of automation reveals the extent to which the activities of service identification can be performed without human engagement (Kazemi et al., 2011; Azevedo et al., 2009). Automating the activities of service identification has many advantages; for example, it reduces the heavy load on human participants that causes human fatigue (Simons and Parmee, 2012). In addition, automation greatly reduces the number of human errors (Kephart and Chess, 2003), and also decreases the system construction, implementation and running costs (i.e., time and money) (Yousef, 2010). Furthermore, automation decreases the required expertise as well as the complexity of service modelling (Baghdadi, 2006).

The technique of service identification has a direct influence on the level of automation as the main objective of many SIMs is to automate the service identification activities (Gu and Lago, 2010; Yousef et al., 2014). However, because of the descriptive nature of the business process or goals, and the lack of tool support, performing the automation is not a trivial task (Zadeh et al., 2016). Therefore, manual mapping techniques (e.g., guidelines and value analysis) have been used by different SIMs (Azevedo et al., 2014; Birkmeier et al., 2013; Klose et al., 2007; Kaabi et al., 2004; Al-Thuhli et al., 2015).

Studying the automation aspect, the relevant SIMs can be classified as:

(i) Automated: this category includes comprehensive SIMs that satisfy and automate all the phases and activities of service identification. Human involvement is not required at any phase (Zadeh et al., 2016; Kazemi et al., 2011).

(ii) Semi-automated: this category includes SIMs that partially automate the service identification activities, such that a software engineer is still required to fulfil some activities manually, such as preparing the input data or evaluating the resulting outcomes.

As presented in Table 2.4, a SIM can be either automated or semi-automated. The rest of the relevant SIMs that do not appear in this table are either manual SIMs, or do not present any automation details.

Table 2.4: Level of Automation in SIMs

Level of Automation	**SIM**	**Count**
Automated	(Khoshnevis and Shams, 2017; Azevedo et al., 2009; Weller et al., 2009)	3
Semi-Automated	(Zadeh et al., 2016; Yousef, 2010; Kazemi et al., 2011; Jamshidi et al., 2012; Dwivedi & Kulkarni, 2008; Bianchini et al., 2014; Guan et al., 2012; Bianchini et al., 2009; Jamshidi et al., 2008; Mani et al., 2008; Azevedo et al., 2009; Leopold & Mendling, 2012; Ma et al., 2009; Yousef et al, 2014)	14

The table reveals that the majority of SIMs belong to the category of semi-automated methods (i.e., 14 out of 29). This is understandable as the first challenge that faces automation is the data input preparation. In this phase, the SIM has to traverse a high abstract level input (e.g., business process), express the required elements and prepare them for the next phase (Zadeh et al, 2016; Yousef, 2010; Jamshidi et al., 2012). The tool support plays a major role in fulfilling this task, thus the lack of tool support means that some activities have to be fulfilled manually. For example, in Yousef (2010) and Yousef et al., (2014), the Riva-based BPA is adopted as an input, and the Essential Business Entities (EBEs) and the relations between them are prepared manually as there is a lack of tool support to traverse Riva models and to complete the integration with the ontology tool (Yousef, 2010). In another example, Zadeh et al.,

(2016) present a framework to identify candidate SOA solutions based on customer requirements. This method automates some of the activities in service identification phase (i.e., mapping process), however, this SIM can be classified as semi-automated because some activities are still not developed such as the scope determination, deriving the customer's requirements, creating the relations matrix, the quantification of the service identification process, and the measurement of service quality factors. In a further research attempt, activities of the services identification are managed and automated in Khoshnevis and Shams (2017). A multi-objective genetic algorithm (i.e., NSGA-II) is adopted to search the domain for proper optimisations. Quantitative design metrics (e.g., coupling and cohesion) are used to evaluate the quality of the resulting services. Human preference is not utilised to enrich the quality of resulting services. Evaluation of services depends on the quantitative fitness function alone.

Automating the SIM becomes more important if the business environment is exposed to rapid changes. In these environments, automation plays a significant role in reducing the time to construct and upgrade the SOA-based IT systems and supports creating efficient candidate services in a short time and at a low cost (Khoshnevis and Shams, 2018). However, different challenges emerge such as the development of a universal method that manages all the different high-level input types (e.g., BPMN, Riva, and RAD). This is a challenging task.

The analysis of automation in SIMs sheds light on a gap in the existing methods. A comprehensive top-down SIM, that automates all the service identification activities, is still missing. Based on this discussion and analysis, it is concluded that developing a SIM that automates all the phases of service identification is useful in addressing the gap in SIMs. In addition, the SIM should examine the validity of the resulting candidate services automatically, such that the outcomes are feasible solutions.

2.2.4. Service Quality Assessment

Quality evaluation is a critical factor that should be considered when developing a SIM. The majority of SIMs state quality evaluation as having a critical role; however, only a minority of these SIMs have quantified their calculations (Zadeh et al., 2016). Quality assessment of candidate services using design metrics is a critical aspect in improving the quality of the candidate services and the service identification approach itself (Huergo et al., 2014). Quantification of quality factors supports the automation of quality assessment (Jamshidi et al., 2012). The quality attributes can be derived from the stakeholder's requirements, the business objectives, and also from the SOA principles (Papazoglou, 2007). The importance of observing the basic software quality design principles has been emphasised (e.g., low coupling and high cohesion) during the whole SOA lifecycle (Erl, 2007). Although the majority of existing SIMs calculate these factors based on technical-oriented input types such as CRUD matrix (i.e., includes atomic operations such as Create, Read, Update, and Delete), these operations do not represent the relationship aggregation in high-level input types (Zadeh et al., 2016). Therefore, it is important to calculate the needed quality factors based on business-aligned input types such as business process models in order to ensure the business-IT alignment (Qian et al., 2009; Jamshidi et al., 2012).

Quality assessment can be quantitative or qualitative or a mixed-design that combines the two approaches (Gu and Lago, 2010; Huergo et al., 2014). Usually, quantitative and qualitative measurement techniques are considered to differ fundamentally. Nevertheless, their objectives and applications overlap in numerous ways. Qualitative assessment is useful in providing a rich, detailed and valid assessment that contributes to the in-depth understanding of the business process and the resulting candidate solutions (Al-Thuhli et al., 2015). Furthermore, qualitative techniques are suitable for gaining a thorough assessment of the outcomes and to make the resulting solutions more acceptable by stakeholders. However, qualitative data analysis is non-statistical and its methodological approach is primarily guided by the actual material at hand (Atlas, 2018). Likewise, the main objective of the quantitative quality assessment is the quantification of the data, which enables the generalisation of the results from a sample to the entire population of interest. Moreover, quantitative assessment provides useful actions and recommendations based on the understanding and analysis of the sample data. Although the findings are usually descriptive in nature, they are only conclusive within the sample of collected data and within the numerical framework (Qian et al., 2009). Quantitative quality attributes usually adopt design metrics to assess capabilities (Huergo et al., 2014). For example, these metrics include coupling (Khlif et al., 2010), cohesion (Khlif et al., 2009), granularity (Weller et al., 2009), and reusability (Jamshidi et al., 2012). Using quantitative metrics supports the automation of the SIM. As discussed earlier, the descriptive characteristics of the input artefacts (e.g., business process) complicate the quantification of these models. The quantitative evaluation values can be obtained using mathematical calculations (Zadeh et al., 2016; Kazemi et al., 2011).

As presented in Table 2.5, SIMs adopt either qualitative or quantitative assessment methods. Only a few SIMs use a combination method (i.e., a mixed approach that combines quantitative and qualitative evaluation methods).

Table 2.5: Quality Assessment

Type	**SIM**	**Count**
Qualitative Assessment	(Klose, Knackstedt and Beverungen, 2007; Inaganti and Behara, 2007; Al-Thuhli et al., 2015)	3
Quantitative Assessment	(Zadeh et al., 2016; Leopold and Mendling, 2012; Dwivedi and Kulkarni, 2008; Bianchini et al., 2014; Bianchini et al., 2009; Guan et al., 2012; Jamshidi et al., 2008; Wang et al., 2005; Mani et al., 2008; Azevedo et al., 2009; Kim and Doh, 2009; Kazemi et al., 2011; Yousef, 2010; Yousef et al., 2014; Kohlborn et al., 2009; Ma et al., 2009; Ebrahimifard et al., 2016; Azevedo et al., 2014; Khoshnevis and Shams, 2017)	19
Combination	(Birkmeier et al., 2013; Jamshidi et al., 2012)	2

It is clear that the majority of the relevant studies (i.e., 19 out of 29) adopt quantitative evaluation techniques. For example, some SIMs use coupling and cohesion design metrics to evaluate the relationships between the candidate services (Khoshnevis and Shams, 2018; Kazemi et al., 2011), whereas a lower number of SIMs (i.e., 3 out of 29) use qualitative assessment to evaluate the resulting services. Using the qualitative assessment, an expert should analyse the resulting solution and provide an evaluation that is subject to the level of knowledge and experience of the expert (Klose et al., 2007; Inaganti and Behara, 2007). Only two relevant studies (Birkmeier et al., 2013; Jamshidi et al., 2012) adopt a mixed-approach for quality assessment. In these studies, a quantitative assessment is used to evaluate the candidate solutions based on the relationships between the business entities, and a survey is employed to evaluate the approach itself using subjective evaluation. The remainder of the relevant studies has not revealed any quality assessment activities (Kaabi et al., 2004; Weller et al., 2009).

Typically, quality assessment focuses on one aspect: either the resulting services or the mapping approach itself. On the one hand, evaluating the resulting services is significant in confirming the validity and feasibility of these services, as well as providing an indicator to the effectiveness and efficiency of the candidate solutions (Zadeh et al., 2016). On the other hand, evaluating the approach is important for examining its usefulness (i.e., in terms of flexibility, efficiency), ease of use (i.e., ease of learning the approach, ease of application, and manageability), and participants' attitude towards using the method or being satisfied with the outcomes (Birkmeier et al., 2013; Jamshidi et al., 2012). However, observing the quality assessment provided by the contemporary SIMs identifies a gap in these methods. A comprehensive evaluation framework to assess all the aspects of service identification in order to enrich the quality of the outcomes, as well as the SIM mechanism is still missing.

Reflecting on the choice of quality assessment type, there is a direct influence of technique and automation on the quality assessment. SIMs that aim to automate the process of service identification usually use quantitative measures to assess the quality of the outcomes. The adoption of quantitative measures supports the SIM automation; this refers to the possibility of utilising the quantitative attributes to obtain a wide range of useful statistical analysis information that can be calculated mathematically. In addition, quantitative data enable accurate benchmarking of the results in comparison with other competitive studies (Azevedo et al., 2009). These reasons justify the use of quantitative measures by the majority of relevant SIMs that aim to automate the service identification activities.

Conversely, qualitative assessment requires quantification of human preference, which is considered more challenging and non-statistical, but the experiments are replicable as the qualitative assessment does not claim that results are universal (Seaman, 1999). Nevertheless, some SIMs consider human involvement as a shortcoming of a SIM, as it may cause human fatigue (Kazemi et al., 2011). In order to utilise the points of strength of the two methods, a combination of quantitative and qualitative techniques should be considered. It is important to know that the two evaluation approaches are not the opposite of each other.

With reference to the focus of assessment, SIMs evaluate either the resulting candidate services or the service identification approach itself. The majority of existing studies evaluate only one artefact (i.e., either the candidate services or the approach). For example, SIMs that assess the resulting candidate services include Jamshidi et al., (2012); Kazemi et al., (2011); Bianchini et al., (2009); and Khoshnevis and Shams, (2017), whereas, other SIMs evaluate the service identification approach such as Klose et al., (2007); Yousef, (2014); and Zadeh et al., (2016). A minority of studies aim to perform a comprehensive evaluation to assess both the candidate services and the identification approach (Birkmeier et al., 2013; Yousef; 2010). Evaluating the outcomes from one perspective is not comprehensive enough to generalise the results (Easterbrook et al., 2008). Therefore, there is a lack of comprehensive SIMs that evaluate the service identification approach as well as the outcomes using a triangulation of evaluation methods that include both quantitative and qualitative measures (Huergo et al., 2014; Vale et al., 2012).

2.3. Conclusions

Service identification is one of the most important activities in the SOA development lifecycle. In performing a gap analysis on a set of relevant studies, three critical gaps in the state-of-the-art of service identification were identified. These gaps are (i) lack of comprehensive SIMs that cover all the phases and activities of service identification, (ii) lack of automation especially of input data preparation and service identification activities, and (iii) lack of sufficient quality assessment. Based on the inclusion criterion, 29 studies were considered as the relevant studies that use high-level inputs such as business process or business requirements. The characteristics of the relevant SIMs were analysed from different perspectives that include the technique and application of human preference, level of automation, and quality assessment.

This study of the literature has established a set of important observations. The most critical insight that emerged from the evaluation is that none of the examined SIMs is comprehensive enough to address all the phases of service identification (i.e., that includes the data preparation, service identification, and output refinement) and cover key SOA concepts (i.e., business and software services) to a sufficient extent. Handling these phases necessitates the satisfaction of the full development life cycle of the SOA applications. This conclusion is supported by the literature where the gap of a unified, comprehensive approach has been identified in a number of publications (Khoshnevis and Shams, 2017; Zadeh et al., 2016; Huergo et al., 2014; Jamshidi et al., 2012).

Automation of service identification activities is an important issue to promote the agility of SOA development. However, the fuzzy nature of the business-related input types limits the automation to just a few activities and necessitates human engagement to fulfil the other activities. In addition, the availability of the tool support of the input business artefact (e.g., BPMN) plays a significant role in automation.

With reference to quality assessment, the majority of existing SIMs use either quantitative or qualitative assessment measures to evaluate the quality of the services. However, none of the relevant studies performs a full evaluation test using a combination of evaluation methods (i.e., a mixed approach of quantitative and qualitative measures) to evaluate the resulting services. Moreover, it is observed that the current relevant SIMs focus on evaluating either the resulting services or the service identification approach, but not both. In addition, there is no combination of quantitative and qualitative assessment technique to interactively derive services. The lack of sufficient quality assessment results in producing low-quality solutions and affects the feasibility of the resulting candidate services as well as producing o complex methods that are not easy to use may be produced.

With regard to the technique used by SIMs, there is no doubt that the selection of an appropriate technique for service identification is one of the significant decisions in order to derive the right services from the beginning. The current techniques still suffer from serious shortcomings, such as being too complicated, too naïve to satisfy the principles of SOA, placing a heavy load on the human participant when performing service identification activities, or not reflecting the reality of the software engineer activities (Huergo et al., 2014; Gu and Lago, 2010; Yousef, 2010). Nevertheless, the SBSE techniques (e.g., genetic algorithms) show promising results. However, they still have some limitations. Although using the interactive SBSE is anticipated to overcome this limitation, it is observed that none of the examined SIMs have used iSBSE techniques for service identification.

From the above observations, the following can be concluded:

1. A comprehensive SIM can be applied to achieve business-IT alignment by deriving candidate SOA services from business processes. The entire activities of this method can be automated and validated in order to produce high-quality feasible services.

2. Role-based BPM, and specifically the BPMN, are considered appropriate modelling approaches to represent the business requirements and can be utilised to derive candidate SOA services.
3. Current service identification techniques have not exploited the interactive SBSE techniques to derive SOA candidate services from the business process. However, the interactive search can provide a promising solution to service identification by supporting the automation and quality assessment using a combination of quantitative and qualitative measures. Moreover, iSBSE techniques are likely to be suitable for satisfying the business-IT alignment as well as being more likely to satisfy the expectations of the stakeholders. Utilising iSBSE for service identification requires in the first place the discovery of a suitable representation method along with the fitness measures.
4. An automatic (or semi-automatic) service identification from the business process models is anticipated to assist in promoting the agile development of SOA solutions. Automation supports to keep up-to-date with the rapid changes in business environment.

This chapter has provided a state-of-the-art review of the service identification to promote business-IT alignment as a vital aid towards finding the significant gaps in current SIMs. This literature review has not only supported the provision of a knowledge-base to identify the problem, but has also paved the way for the design of a research artefact that can propose a solution to these problems. Consequently, this chapter is linked to both steps “1” (i.e., Problem identification and motivation) and “2” (i.e., Objectives of a solution) in the DSRM (Peffers et al., 2007). These two steps set the basis for the service identification problem and reveal the significance of addressing this problem.

The next chapter presents the research methodology utilised in this thesis within the design science research context. The interactive search-based framework, the main research artefact in this research, is presented to derive business-aligned candidate SOA services from business process models.

Chapter 3
Research Design

3.1. Introduction

Following the detailed review of the state-of-the-art literature in Chapter 2 regarding the service identification methods (SIMs), it was concluded that the contemporary SIMs that derive SOA candidate services from the business process can suffer from one or more problems, namely: (1) not covering all the phases of service identification, (2) not automating the activities of service identification, and (3) not performing a comprehensive quality assessment. The first problem results in producing methods that are not practically applicable due to some important enterprise concerns being ignored (Jamshidi et al., 2012). The second problem results in a heavy reliance upon the human factor in fulfilling the service identification activities. In addition, due to the lack of automation, the candidate services are more exposed to errors as the related architectural decisions rely heavily on the human participant's knowledge and experience. The third problem, due to the lack of adequate quality assessment, results in constructing low-quality or not-feasible candidate services and also produces complex service identification approaches. Addressing these issues can result in many improvements on the derivation of SOA services from the business process as well as better business-IT alignment that practically bridges the gap between business processes and SOA services.

In a step towards resolving these problems, the research methodology proposed for this research will be accomplished by means of Design Science Research Methodology (DSRM) (Prat et al., 2014; Hevner et al., 2004; Peffers et al., 2006; Hevner, 2007). The interactive search-based framework presented in this research derives SOA services from a given enterprise's role-based business process models (e.g., BPMN).

This chapter has the following objectives:

1) Define the scope of the research;
2) Present the research methodology and set the requirements of the research artefact according to its context;
3) Identify the required characteristics that the interactive search-based framework needs to possess; and
4) Present the interactive search-based framework for service identification that aims to achieve the research objectives.

The main focus of this research is on the proposition that an interactive search-based technique can be the driver for the development of an effective and comprehensive service identification framework that derives SOA candidate services from role-based business process models (i.e., BPMN). Therefore, service identification has been formulated as a search-based optimisation problem (Harman and Jones, 2001). A metaheuristic optimisation algorithm is proposed as a central technique to address this problem. An incremental process has been conducted to address the gaps identified in Chapter 2 as follows. Firstly, a comprehensive framework has been developed to satisfy all the phases and activities of service identification. Secondly, a search-based technique has been adopted to satisfy the automation and the quantitative quality assessment of the resulting candidate services. Thirdly, to enrich the quality of the resulting outcomes, human preference has been adopted in the context of interactive search to guide the construction of SOA solutions. The research framework is utilised to handle the incremental development of the service identification framework through a set of iterations (Peffers et al., 2007).

3.2. Overview of Research Methods

Experimental software engineering is a sub-domain of software engineering where the focus is on performing experiments on software systems (Seaman, 1999). Empirical research in software engineering comprises different methodologies that are adopted to accumulate knowledge to understand the research problem and evaluate the proposed products (Easterbrook et al., 2008). Empirical research methods include different methodologies such as positivist, interpretive, and design science (Collis and Hussey, 2013). Positivism uses quantitative methods to conduct the research and is usually used to study existing relations within phenomena. Interpretive research is used to analyse the social context of the phenomena to build a comprehensive understanding. The interpretive method considers the subjective meanings of participants when they interact with their environment; therefore, qualitative techniques are used to derive and analyse the acquired data (Rowlands, 2005). Design science is a research perspective that helps to conduct the research using iterative stages. The design science paradigm helps the researcher to gauge the progress and changes in the research. This paradigm allows the adoption of both quantitative and qualitative methods to perform the evaluation.

The nature of this research project plays a major role in selecting the right methodology. Conducting this research project requires two main issues to be resolved: (i) incremental development is required to develop an interactive search-based framework that satisfies all the objectives of this research (ii) different empirical methods (e.g., case study) are required at each development stage to perform the evaluation. On the one hand, given that positivist research is objective, it neglects ideologies, values and passions; this is considered a key drawback (Ryan, 2006). On the other hand, the interpretive method focuses mainly on the qualitative side and employs qualitative methods and an inductive process (Orlikowski and Baroudi, 1991). Thus, it is concluded that these methods are not comprehensive enough to satisfy the requirements of this research, and thus they are ill-suited for this project.

The DSRM has an iterative nature such that resulting artefacts are produced incrementally in different iterations. In addition, DSRM allows the adoption of both quantitative and qualitative approaches, which is a key requirement when conducting all the phases of this research. Different empirical methods such as experimental research (i.e., controlled experiment), case-study (Walsham, 1995), survey (Sjoberg et al., 2005), and statistical analysis (Kitchenham et al., 2008) can be used through the DSRM iterations; hence, DSRM is recommended within the domain of the information systems (Hevner et al., 2004). Based on the advantages of DSRM, and considering the nature and requirements of this research study, it was concluded that DSRM is the appropriate method to carry out the phases of this research, and thus it has been adopted to develop the framework of this research.

The general concept of design science was devised by Simon, (1996) based on the creation of innovative design artefacts with particular settings. The design science method was recognised as being useful in Information Systems (IS) by Hevner et al. (2004) who derived a set of guidelines to support this. These guidelines include the design as an artefact, problem relevance, design evaluation, research contributions, research rigour, and design as a search (Hevner, 2007). DSRM integrates practices, procedures, and principles that are required to fulfil the objectives of IS research studies. Moreover, DSRM frames all the activities of the research such as problem definition, design, evaluation and communication. Peffers et al., (2007) combine different procedures, principles, and practices that present a DSRM process model. This model comprises six phases as depicted in Figure 3.1.

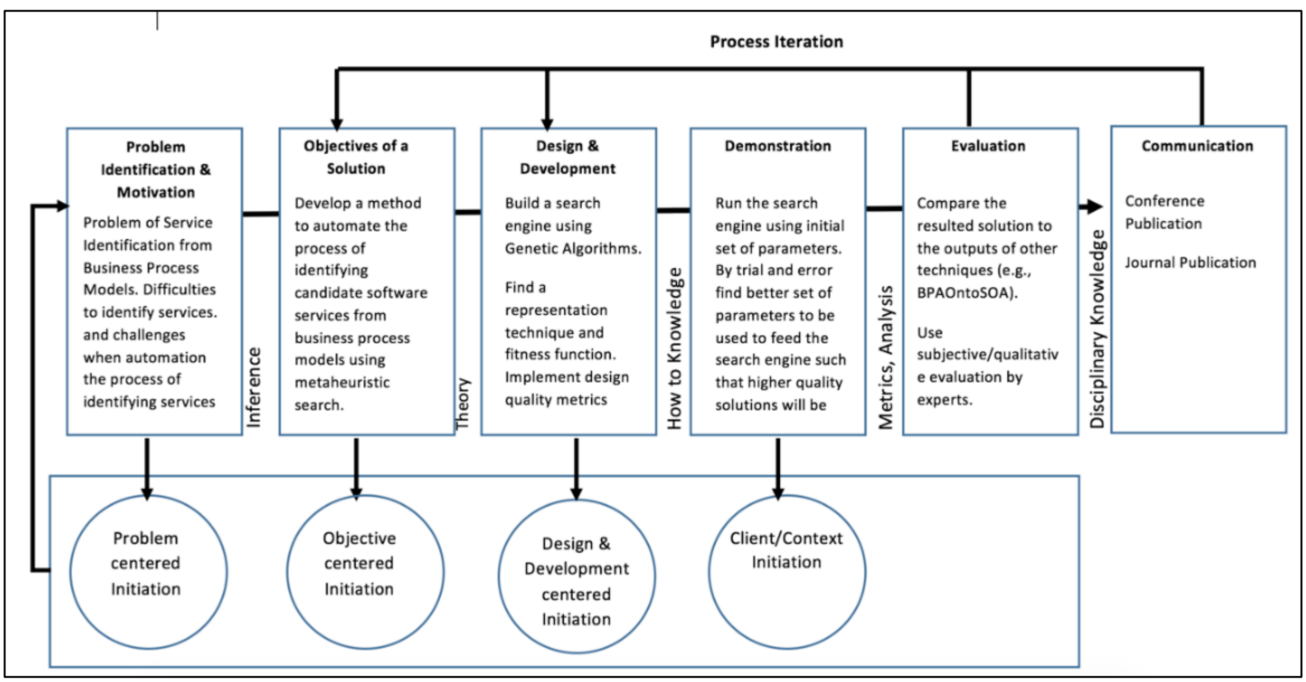

Figure 3.1: Design Science Research Methodology (DSRM) Process Model (Peffers et al., 2007), Licensed under CC-BY-SA 3.0.

Each phase of the DSRM process has its own characteristics and has a specific task to perform as follows:

1) Problem identification and motivation: define the research problem and the motivation drawn from the literature review, as well as possible methods that guide researchers to find a candidate solution.
2) Objectives of a solution: use the outcomes of the problem identification phase to derive the solution objectives. Select the best solution amongst the set of potential solutions.
3) Design: develop the design and artefacts (e.g., constructs, models, methods and instantiations) of the solutions. This phase can be divided into two phases, namely the Perception design phase (i.e., initial preparation step) and the Detailed design and Prototyping phase (i.e., in which the artefacts are created).
4) Demonstration: verify that the design meets its specifications and objectives. This can be done by means of a case study, simulation, or other types of experiments.
5) Evaluation: assess the efficiency and effectiveness of the solution artefacts, and observe the results from the demonstration phase in comparison with the expected results. This phase comprises a set of techniques that may include triangulation of methods by using quantitative and qualitative metrics. Based on the evaluation results, the decision should determine whether to iterate back to the previous phase (i.e., Design) or to move on to the next phase.
6) Communication: publish the findings and share them with other researchers through professional publications and media.

This research adopts DSRM as it facilitates constructing, demonstrating, and evaluating IT artefacts since DSRM process divides the research life cycle into a number of phases and focuses on the development, demonstration, and evaluation of IT artifacts within the discipline of Information Technology (IT). Moreover, it supports the adoption of other empirical methods (e.g., case study) to conduct the demonstration and evaluation experiments. DSRM also supports the incremental development of IT artefacts by conducting multiple iterations. Each iteration comprises a set of phases such as design and development, demonstration, and evaluation. Each iterative loop feeds into the next loop as well as building on the outcomes of the previous iteration.

The next section presents the adoption of the research methodology for this study in line with the DSRM as described above. Furthermore, the research design reflects on the fulfilment of the gaps identified in Chapter 2.

3.3. The Research Design and the Adoption of DSRM

This section presents the design of this research where the DSRM process model is applied. The DSRM process iterates on the phases of design and implementation, demonstration, and evaluation to incrementally develop the service identification framework. Conclusions and outcomes of an iteration feed into the next iteration until the research objectives are satisfied. Learning from each iteration before proceeding to the next one is an essential characteristic of DSRM.

Figure 3.2 depicts the design of this research using DSRM which covers all the required activities that are needed to satisfy the research objectives. Firstly, the DSRM manages the problem identification and research motivation (i.e., first DSRM phase). Next, a thorough literature review is conducted to study the state-of-the-art with regard to service identification. In light of the results of the gap analysis, the research objectives are identified (i.e., second DSRM phase). The iterative nature of DSRM provides support to incrementally develop the research artefacts until the research objectives are satisfied. Through these iterations, the service identification layered framework is developed, demonstrated, and evaluated. Each iteration includes the phases of design and development (i.e., third DSRM phase), demonstration (i.e., fourth DSRM phase), and evaluation (i.e., fifth DSRM phase). The outcomes of each phase feed into the next phase, and based on the evaluation results looping on these phases continues (i.e., the objectives are not met completely) or stops (i.e., the outcomes are satisfactory and the research objectives are met).

The research conclusions, outcomes, and future directions are managed in the last DSRM phase (i.e., communication). In this phase, results are communicated to disseminate the research outcomes and highlight research contributions.

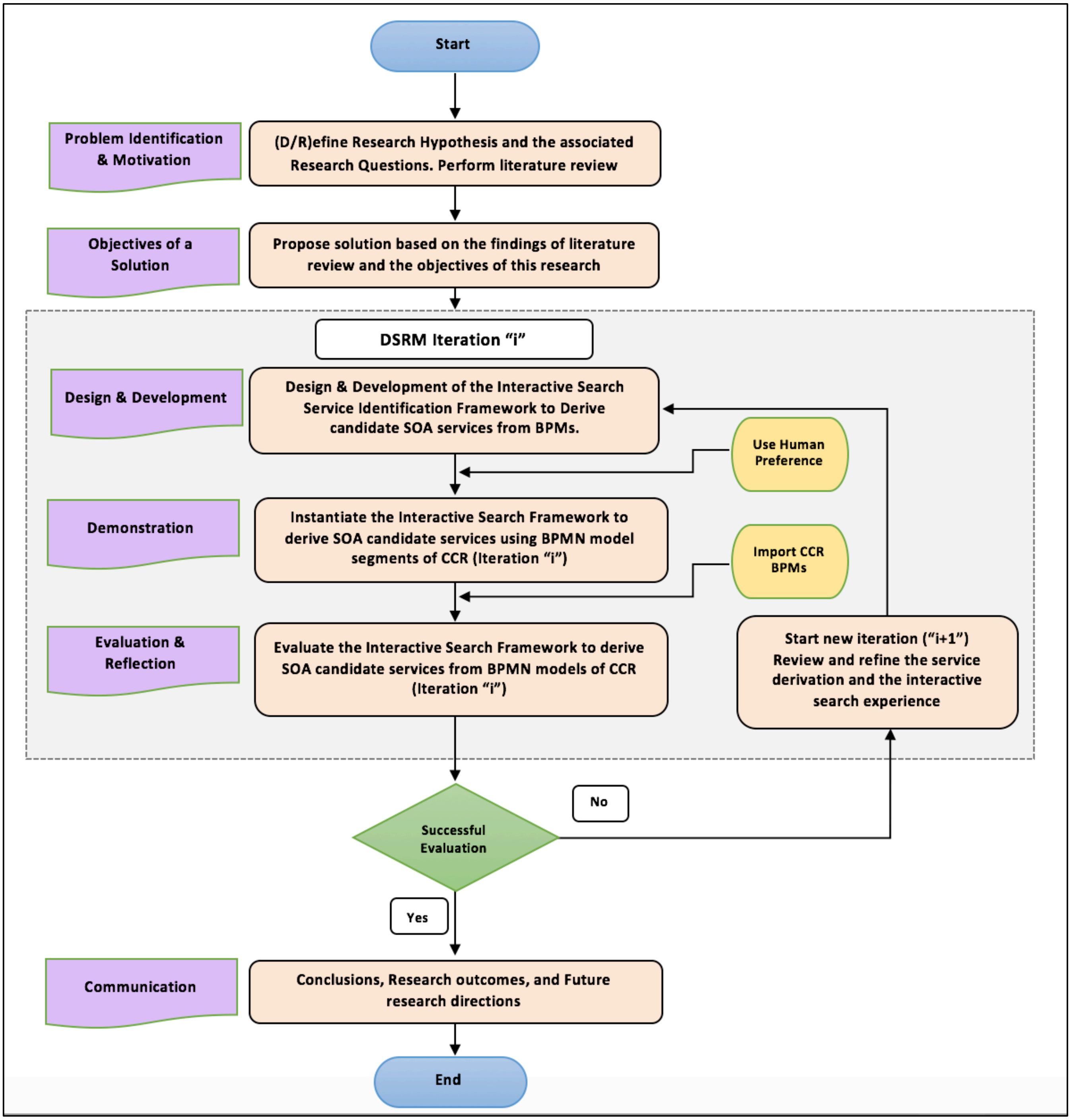

Figure 3.2: Research Methodology in DSRM Phases

The development of the proposed framework is distributed on three iterations and each iteration has an input and an anticipated output. The design of this research is prepared so as to address the identified gaps in the literature review of service identification and incrementally develop a framework to derive services from BPMs such that each iteration extends the service identification framework.

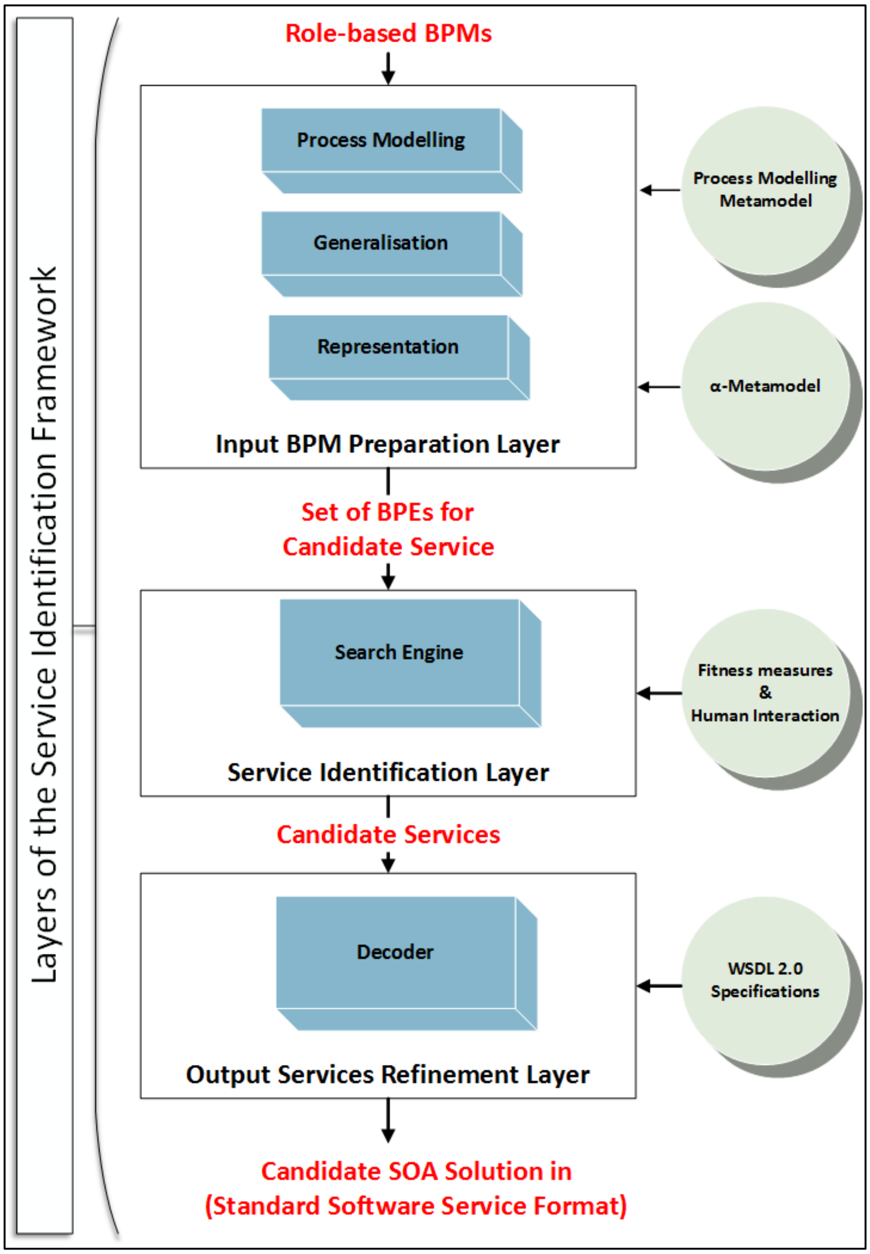

Figure 3.3: The Architectural Design of BPMiSearch Framework

In the first iteration, the basis of the service identification framework is defined and the search space is initialised. That includes presenting the details of each layer of this framework. However, the activities of the service identification layer (i.e., the middle layer) will be managed manually by domain experts. Domain experts are the people with high-experience and knowledge of the workflow (i.e., fully or partially) of an organisation regardless of their original background. Testing the framework using domain experts helps to reflect on the effectiveness of the proposed framework, and feeds into the next iteration before using a search algorithm. The characteristics of the service identification framework are presented in full detail in Chapter 4. The second iteration focuses on using the search algorithm components (i.e., the representation and fitness function) to identify the services. This iteration presents the utilisation of a genetic algorithm to perform the search and the set of experiments that aim to explore and exploit the search space. The outcome of the second phase is a fully-automated service identification framework that uses search with quantitative measures. This iteration paves the way for the use of interactive search in the next iteration. Chapter 5 depicts the 2^{nd} DSRM iteration. The search experience is extended in the third iteration by adding a human qualitative fitness measurement to enrich the quantitative fitness measures presented in the 2^{nd} DSRM iteration. In this iteration, the human preference would be utilised to steer the trajectory of the search to enrich the quality of the resulting candidate services. Experiments of this iteration should clearly highlight the impact of human preference on the search outcomes. The three DSRM iterations are presented in detail in the next sections.

3.3.1. DSRM First Iteration: The Search Space

The objective of this iteration is to address the first gap identified in the state-of-the-art literature review by presenting a comprehensive method that manages all the phases of the service identification. This section briefly introduces the main DSRM activities of this phase.

This DSRM iteration has two main objectives that are aligned with the first and the second research questions: RQ1 and RQ2. The first objective relates to the investigation of possible contributions of various business context modelling activities. As discussed in Chapter 2, role-based business process models have been selected as an input. The first research question (RQ1) has been formulated to be aligned with this objective:

> ***"To what extent can role-based business process models such as BPMN 2.0 models be mapped to services following Service Oriented Architecture (SOA) principles?"***

Addressing this research question necessitates applying a top-down approach that uses a role-based BPM (e.g., BPMN 2.0) as input in order to derive SOA services. Initially, this can be completed by defining the set of business process elements that are needed for service identification. Following this, it is important to define the main targeted SOA principles (e.g., loose coupling and high cohesion) to be satisfied that feasible solutions can be produced (Papazoglou et al., 2007; Erl, 2008). This triggers a set of sub-questions such as (i) what business process elements should be considered? (ii) What metamodel should be adapted to traverse the BPMN diagrams to derive the required business process elements? (iii) What is the right granularity level of the business process elements to be considered? And (iv) what is the criterion for preparing the basic business process elements at that level of granularity?

To address this objective, a layered framework is proposed to manage the generalisation and abstraction of the input role-based BPMs. Moreover, a metamodel is presented to enable the generic service identification framework to address different role-based BPMs, even when adopting different modelling languages (e.g., BPMN, RAD, and UML AD). Figure 3.4 presents the three sequential activities of this layer: (i) business process modelling (ii) generalisation, and (iii) representation.

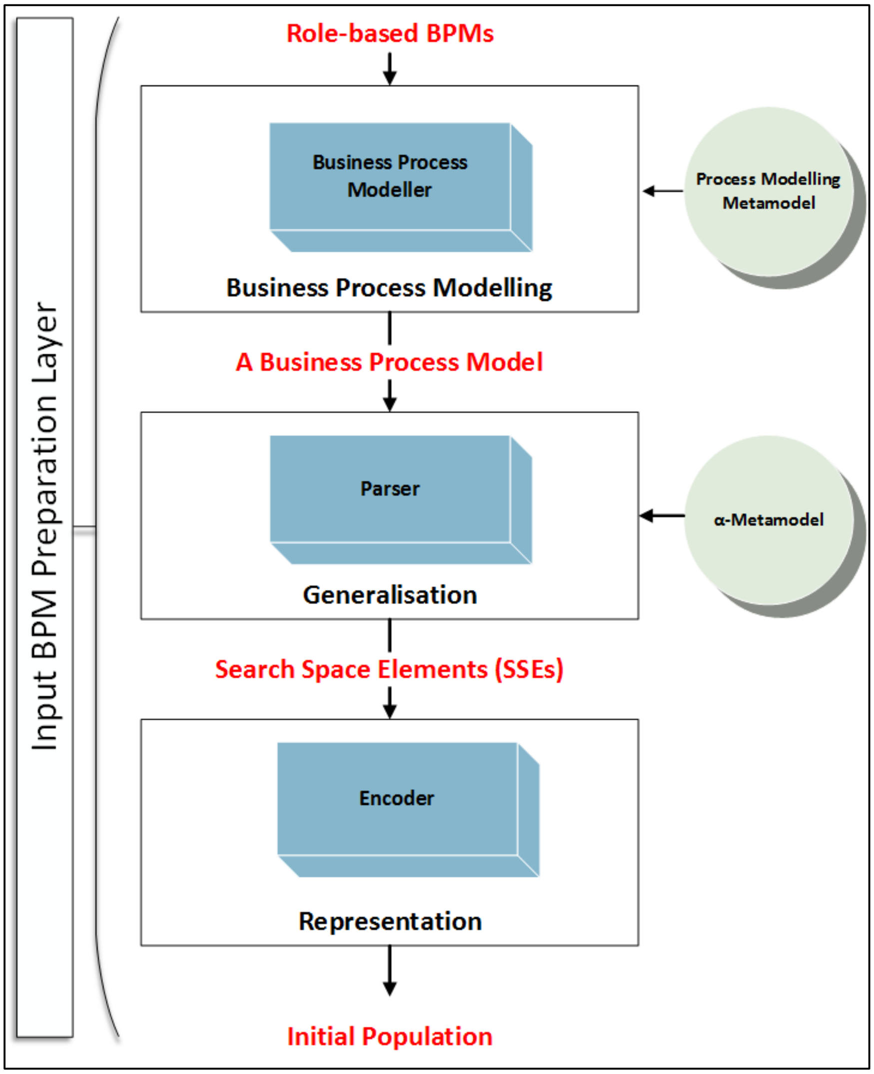

Figure 3.4: Input BPM Preparation Layer

These activities are necessary to ensure the flexibility of the generic design of the service identification framework and also to promote the usability of the framework's sub-layers.

The second objective of this phase is to investigate the appropriateness of various system modelling techniques as a basis for the representation of effective metaheuristic search. The anticipated output is a representation technique that suits the construction of SOA services within the proposed search-based technique, as well as the formulation of a suitable fitness function to examine the resulting solutions. Therefore, the second research question (RQ2) can be formulated as follows:

"In what way can SOA services be best implemented for metaheuristic search?"

Answering this research question necessitates investigating the different representation methods such as string, binary, integer, and tree representation techniques. Based on its appropriateness for the service identification domain, a suitable representation should be utilised to construct the search space. Subsequently, the initial population would be generated using input data from the previous step.

3.3.1.1. Design and Development

The design and development phase includes the following:

(i) The design and development of a comprehensive, layered service identification framework: this framework is anticipated to cover all the necessary activities to derive candidate SOA services from the BPMs. These activities are divided into three major layers.

  1) Input preparation phase: this layer aims to prepare the input role-based models in a generic format that allows the framework to be generalised to manage different modelling languages. A special purpose metamodel that is named α-Metamodel is proposed to encapsulate the input business process elements and tag them with generic tags that allow the generalisation activity. Since the BPMN is adopted by this research as the role-based modelling language, a systematic approach to traverse the BPMN models and derive the needed raw business process elements has been designed. These raw elements are prepared at a higher abstract level of granularity to create the Search Space Elements (SSEs), which represent the basic blocks required to construct the search space.
  2) Service identification: allocate the SSEs in candidate services based on the dependency and connections between these elements.

3) Output data refinement: resulting candidate services are extracted again to find the raw business process elements. These elements and activities are mapped to the corresponding service components based on a specific mapping criterion.

(ii) Representation method: to represent the service identification problem solution space such that it is sufficiently abstract for human comprehension but also effective for a computational evolutionary search. This includes the following activities:

1) Thoroughly analyse the current representation techniques (e.g., tree-based, string, and binary representation) to find an appropriate and adequate representation method that suits the service identification problem.

2) Based on the selection of the representation technique, a clear criterion of encoding and decoding should be formulated to convert the SSEs into a genotype (i.e., build the search space), then decode the resulting candidate solutions to SOA services.

3.3.1.2. Demonstration

(i) A manual mapping experiment to identify services from BPMs would be utilised to test the service identification method. This includes a set of activities (i) to traverse the input BPMN models to derive the raw business process elements that construct the SSEs (ii) to allocate the SSEs in candidate clusters (i.e., services) based on the relationships between them, and finally (iii) to match each element in the resulting candidate services to the corresponding service component based on the standards of web services (Inaganti and Behara, 2007).

(ii) A sufficient and representative case study would be used to examine the mapping approach. Cancer care and registration (CCR) of King Hussein Cancer Centre (KHCC), Jordan (Yousef, 2010) is considered as a sufficient and representative case study (i.e., see Section 3.4), thus, CCR BPMN would be used as input from which Cancer Care services would be derived.

(iii) A statistical analysis test will need to be performed to examine the effectiveness of the proposed mapping method in comparison with randomly produced services. Statistical analysis checks for significant improvement of the proposed method using fitness measures that are calculated mathematically for the outcomes of the two methods.

Results of this phase will be used in the DSRM evaluation phase to assess the method and compare it to other relevant SIMs.

3.3.1.3. Evaluation

To evaluate the outcomes of this DSRM iteration as well as the provided framework, validation and verification techniques would be adopted to compare and contrast various models selected as candidates for this research. Validation techniques reveal whether the right product is being developed, whereas, verification techniques reveal whether the product is being developed in the right way.

(i) Baseline evaluation would be used to check and inform if the outcomes of the proposed method outperforms the outcomes of a random method.

(ii) The conformance to SOA principles is to be examined to inform the feasibility of the resulting candidate services.

(iii) An industrial standard benchmark will be used to compare the developed method to other approaches. Comparing the characteristics of the developed method to the characteristics of other methods is important to validate the resulting service solutions (Dolan and Moré, 2002). Conducting a comparison with a similar framework for service identification, i.e., BPAOntoSOA (Yousef, 2010) is a beneficial way to show the improvements that have been achieved by developing the BPMiSearch framework. Nevertheless, the BPAOntoSOA is an ontology-driven framework that derives SOA candidate services from Riva-based BPA. However, a set of significant factors support conducting an accurate comparison such as: (i) BPAOntoSOA is a comprehensive framework that covers all the activities of service identification, which helps to match and compare the activities along all phases and activities (ii) both BPAOntoSOA and the developed method are top-down approaches that derive candidate SOA services from business process (iii) the two frameworks adopt the CCR case study to evaluation, which helps to examine the resulting outcomes of the two SIMs using the same fitness measures and supports conducting an accurate comparison. The comparison at this iteration focuses on the following aspects: the abstraction level of the inputs, and the conformance to SOA principles.

(iv) A Walkthrough method would be utilised to perform a verification test to inform the correctness of the developed method by inspecting the components that comprise the resulting candidate services. Based on Zowghi and Gervasi, (2003), correctness is defined as a combination of completeness (i.e., the availability of all the desired information to hold a certain context) and consistency (i.e., the situation where specifications have no internal contradictions). Inspecting the number of resulting elements in comparison with the elements in the original BPMs would reveal the consistency between the input BPMs and output services.

(v) To inform the validity of the representation technique, a representation is required to:

1) Satisfy the notion of services and conform to the principles of SOA.
2) Define a clear form of encoding the problem elements to genotype elements, and then decode from genotype to real-world elements.
3) Enable efficient genetic operators (i.e., selection, crossover, and mutation).
4) Enable an obvious criterion for evaluating the resulted solution.

3.3.2. DSRM Second iteration: The Search Process

This iteration aims to address the second gap of service identification identified in the literature review, (i.e., the lack of automation). Building on the outcomes of the previous iteration, the activities of the initial framework that were developed at the previous stage are to be fully automated at this stage using metaheuristic search. Therefore, the objective of this DSRM iteration is to investigate novel ways of adopting and formulating an appropriate search algorithm that fulfils the objectives of this research, leading to the development of a search 'engine' to identify SOA services from BPMs. Accordingly, the third research question (RQ3) is formulated in order to be aligned with this objective:

> ***"How can the services solution space be effectively and efficiently explored and exploited in order to derive candidate SOA solutions from BPMN 2.0 models?"***

Satisfying this research question requires conducting a set of experiments to explore and exploit the search space to find SOA services of good quality. The experiments help to test the genetic operators as well as the appropriateness of the fitness function to evaluate the resulting services. Moreover, to fulfil the objective of this iteration, a comparison with other techniques should be performed to show the effectiveness of using a search-based method to derive candidate services from BPMs.

Chapter 5 illustrates in detail the development, demonstration, and evaluation phases of the proposed search-based method of service identification. The following sections briefly introduce the activities within each DSRM phase that belongs to this iteration.

3.3.2.1. Design and Development

The methodology to provide a search-based framework for service identification includes the following activities:

(i) Introduce the development and implementation details of the genetic algorithm. This includes the representation along with the genetic operators (e.g., selection, crossover, and mutation) and fitness function

(ii) Present the implementation details of the fitness function that quantitatively evaluates the resulting candidate services. Formulas of the fitness measures have been introduced in the previous iteration.

3.3.2.2. Demonstration

(i) Initial parameter tuning experiments are conducted to investigate the proposed representation method and associated genetic operators. The genetic parameters (e.g., population size, selection, crossover, and mutation probabilities) are to be explored and tuned using empirical trial and error experiments. It is important that the search is largely replicable and to ensure that it is not driven by an element of randomness, thus each experiment comprises multiple runs (e.g., 50 runs) enabling population average population fitness and standard deviation to be calculated and recorded. Resulting solutions are to be compared to the outcomes of a random algorithm. In the course of developing efficient and effective search applications, it is often necessary to tune parameters by empirical investigation. However, since each tuning parameter has a large scope of possible values, a large number of potential collaborations of parameters are anticipated. This large number of parameter collaborations causes an explosion of parameter combinations at the end (De Jong 2006).

(ii) The effectiveness and efficiency of the search-based method are to be investigated. The performance of candidate SOA solutions will be examined by measuring the coupling and cohesion values. The effectiveness of the algorithm (i.e., represented by the average population fitness values at convergence) will also be compared with values obtained from other approaches (e.g., baseline and manual methods). The higher the fitness value, the better the quality. In contrast, the efficiency reveals the number of generations needed to achieve the best fitness values or to arrive at a fitness plateau. The efficiency would be measured by either the number of generations or the time needed to arrive at a fitness plateau.

(iii) Consistent with the overall investigation strategy, algorithm speed is important and will need to be measured. The observation of the algorithm speed is significant to pave the way for the interactive search in the next iteration.

(iv) Clear colourful visualisations of candidate SOA solutions will be produced to support the later evaluation of resulting candidate solutions by domain experts.

3.3.2.3. Evaluation

(i) The effectiveness and efficiency of the search-based method will be assessed by analysing and comparing the fitness values of sample services produced using the search-based method in comparison with sample services produced using the baseline (i.e., random) method. A statistical analysis test will be conducted to compare the outcomes of the two methods (i.e., search and baseline) in terms of the quality metrics (i.e., population average coupling and cohesion). A set of statistical analysis tests will be conducted to examine the significant improvement of the search method over the baseline method using a hypothesis test (e.g., t-test). The selection of the proper hypothesis test depends on different factors such as the normal distribution of the data as well as the sample size. The null and alternative hypotheses to be examined are formulated as follows:

$$\boldsymbol{H_0}: No\ significant\ difference\ between\ the\ Search\ method\ outcomes\ in\ comparison\ to\ the\ baseline\ mehtod\ outcomes.$$

$$\boldsymbol{H_\alpha}: There\ is\ a\ significant\ difference\ between\ the\ Search\ method\ outcomes\ in\ comparison\ to\ the\ baseline\ method\ outcomes$$

The results of the statistical analysis test are anticipated to show whether the search method outperforms the baseline method (i.e., the null hypothesis is rejected), which means that the search method outcomes are better than the outcomes of the baseline method (i.e., in terms of effectiveness and efficiency). However, the acceptance of the null hypothesis means that the search method with the quantitative fitness function is not a good alternative to enhance the quality of the SOA solutions.

(ii) The investigation of the local search will at this point be extended to another comparison using services produced by domain experts (i.e., the outcomes of the 1st DSRM iteration presented in

Chapter 4). Therefore, conducting this comparison is anticipated to highlight the importance of using the search method to automate the derivation of services as well as the capability of the search method to enrich the quality of resulting services (i.e., in terms of effectiveness). The null and alternative hypotheses of this test can thus be formulated as follows:

$\boldsymbol{H_0}: No\ significant\ difference\ between\ the\ search\ method\ outcomes\ in\ comparison\ to\ the\ manual\ method\ outcomes.$

$\boldsymbol{H_\alpha}: There\ is\ a\ significant\ difference\ between\ the\ search\ method\ outcomes\ in\ comparison\ to\ the\ manual\ method\ outcomes$

Results of the statistical analysis test will show whether the search outcomes outperform the manual method outcomes (i.e., the null hypothesis is rejected). However, the acceptance of the null hypothesis means that the search method is not capable of improving the quality of the resulting solutions to outperform the manual produced services.

(iii) The resulting candidate solutions will be examined to check for validity and feasibility using the knowledge and experience of domain experts. With the support of the colourful visualisation of the resulting services, domain experts will be able to analyse and evaluate the search outcomes.

(iv) Evaluation by domain experts: the assessment of experts is important to inform the feasibility of the resulting candidate services, as well as to inform the level of satisfaction. Using domain experts at the King Hussein Cancer Centre (KHCC) in Jordan, a set of derived candidate service solutions will be evaluated. With the support of the colourful visualisation of the resulting services, the domain experts will be able to analyse and evaluate the search outcomes. Participants are anticipated to provide their feedback using a Likert-scale (Likert, 1932). This numerical scale will be utilised to elicit the feedback from participants as it helps to quantify the participant's feedback values. The Likert-scale is designed to measure the participant's opinions or attitudes by measuring the levels of agreement or disagreement (Bowling, 1997; Burns and Grove, 1997) using a fixed-choice response format. This ordinal scale is considered one of the most reliable scales among other scale sizes as it performs best based on various research studies (Van et al., 2004). Moreover, it provides the required detail that helps to analyse what a user is trying to say. Table 3.1 presents the satisfaction scale that allows participants to share their experience and evaluate each service. This scale ranges from "Very Satisfied" to "Very Dissatisfied". Each choice is converted to the corresponding value as this table suggests.

Table 3.1: Likert-type Satisfaction Scale

Scale Number	Description	Human Feedback Value (%)
10	Very Satisfied	100
9		90
8	Satisfied	80
7		70
6	Neutral	60
5		50
4	Dissatisfied	40
3		30
2	Very Dissatisfied	20
1		10
Void Response	N/A	0

(v) Benchmarking: a further investigation will be used to examine the features of using the search SIM in comparison with the BPAOntoSOA (Yousef, 2010). The comparison with BPAOntoSOA in this iteration focuses on aspects such as automation, multiple solutions and correction actions, feasibility checks, and the format of the resulting candidate services.

3.3.3. DSRM Third Iteration: The Human Preference

The fourth objective is associated with investigating ways in which user preference, implicit knowledge and experience can be exploited within an interactive, human-in-the-loop search, enabling appropriate machine learning techniques to facilitate mutual learning between machine and human. Addressing this objective is anticipated to bridge the quality assessment gap (i.e., quantitative and qualitative) identified in the service identification. The Interactive Search provides a comprehensive mechanism that combines subjective human feedback with objective fitness measures. Moreover, it is anticipated that the derived candidate services are more likely to satisfy stakeholders because of the human participation in producing these services. Therefore, to address this objective, the fourth research question (RQ4) states:

> ***"What is the impact of human interactive preference on the search outcomes?"***

This requires introducing an interactive search-based framework (i.e., BPMiSearch) that provides the capability of eliciting participant's feedback in order to enrich the quality of the resulting solutions. RQ4 triggers a set of sub-questions, such as the proper time of eliciting human feedback, the appropriate technique to ask for feedback values, suitable mechanisms to use the feedback values, and the possibility of reducing human fatigue. The evaluation of Interactive Search outcomes should be designed properly to assess the impact of human preference on the quality of the derived candidate services in comparison with services produced using a search method with quantitative measurements only (i.e., developed in the previous iteration).

3.3.3.1. Design and Development

(i) To jointly steer the trajectory of the search, a balance is required between the human participant and the interactive support framework. However, several factors should be considered when striking an appropriate balance between the quantitative and qualitative fitness measures. These factors include the selection of representative presentation individuals from the population, as well as finding a sufficient number of presentation solutions that can effectively steer the direction of the search. The scale and complexity of the problem play an important role in selecting the presentation solutions.

(ii) A novel mechanism would be introduced to derive the participant's subjective evaluation during the search process. In the interactive context, providing qualitative human evaluation helps to steer the trajectory of the search. Participants' feedback suggests that positive interaction with the search framework might improve the quality and appearance of the produced candidate software services (Simons and Parmee, 2012). However, because the metaheuristic search is typically population-based, presenting each individual for presentation is not possible as it would cause evaluation fatigue (Shackelford and Simons, 2014). Instead, a number of individuals should be selected for presentation such that an effective search could be conducted. Thus, the design of an effective selection mechanism to select presentation individuals is a challenging but significant mechanism in the metaheuristic interactive search. The selection of presentation individuals should be performed using a specific criterion and at the proper time. Different solutions are proposed to select presentation individuals such as reducing the population size or selecting an individual for presentation after a fixed number of generations (Shackelford and Simons, 2014). The colourful visualisation of presentation solutions is anticipated to improve the participant's learning curve and help users to accurately comprehend the presentation solutions.

(iii) It is important to effectively integrate machine-based quantitative fitness metrics with qualitative user evaluations to perform a metaheuristic search. The quantitative search process is performed using the coupling factor and cohesion of services are accumulated in a weighted fitness function. At some selected interactive generational intervals, SOA candidate solutions are to be presented with a colourful visual illustration of the service solutions. The participant will be asked to subjectively evaluate the SOA solutions by providing numerical numbers (i.e., using the Likert scale) that ranges from 1 (i.e., poor solution, the user is very dissatisfied) to 5 (i.e., good solution, the user is very satisfied). A special tool is provided to allow for the evaluation of each candidate service as well as the entire solution. The weight of the subjective assessment is investigated thoroughly and selected based mainly on the experience of participants.

(iv) Reduce interaction fatigue by creating a balance between machine-calculated fitness measures and human interaction. To investigate this balance, a number of issues should be considered. This includes the size and complexity of the data segments to be presented to the participant and the level of the participant's experience and knowledge. In addition, the length of the interactive session, as well as the possibility to take breaks as needed are important factors that support reducing human fatigue.

3.3.3.2. Demonstration

(i) Domain experts at KHCC will collaboratively interact with the search method to steer the trajectory of the search. Genetic parameters that have been obtained in the previous iteration will be used to provide an accurate comparison with the Search method and to show the impact of human interaction on the resulting solutions.

(ii) A sufficient and representative case study (i.e., CCR) will be utilised to demonstrate the interaction with the framework that has been developed at the current iteration. The impact of the human interaction will be observed and analysed statistically by conducting a comparison with the Search method depicted in the previous iteration.

3.3.3.3. Evaluation

(i) Significant improvements using the Search method: examine the effectiveness and efficiency of the Interactive Search in comparison with the Search method. Statistical analysis is required to clearly identify any significant improvements.

(ii) Level of satisfaction: measure the level of the domain expert's satisfaction by deriving their subjective feedback. This experiment helps to understand the impact of human interaction on the level of satisfaction regarding the candidate services constructed with the collaboration of the domain experts. In addition, the level of satisfaction reflects the feasibility of the resulting solutions and informs their validity as well.

(iii) Benchmarking: the benchmarking at this stage sheds light mainly on the quality assessment approach provided by the BPMiSearch in comparison with BPAOntoSOA. Benchmarking is useful to ensure that the BPMiSearch framework provides a comprehensive quality assessment method that ensures the feasibility and quality of the resulting solutions. In this iteration, the benchmarking will focus on two aspects: the quality assessment (i.e., the use of quantitative and qualitative quality assessments to evaluate the resulting services) and subjective evaluation by domain experts to measure their level of satisfaction.

3.4. The Research Case Study

The case study is an important empirical method for evaluating the resulting artefacts of the DSRM iterations. The characteristics of the case study should be representative and sufficient enough to carry out all the phases of DSRM. A case study can be representative if it aims to reflect a broad population of cases in order to represent the full variation of a population (Seawright and Gerring, 2008). It can be also identified to be sufficient if it covers the full aspects that are required to examine a research artefact (Gerring, J., & Cojocaru, 2016). Moreover, as this research aims to provide an interactive method (i.e., with human in-the-loop), part of being a representative example, it is required to have access to domain experts who comprehend the workflow of the organisation and have the capability to interact with the framework and evaluate the results. The selection of domain experts depends in the first place on the domain of experiments. For example, a university example requires the selection of domain experts that understand the workflow of that university regardless to their original background. In a university, a domain expert can be a lecturer, a module leader, or software developer, the condition is to be familiar with the activities that take place in the university. Hospital is another example, in which a domain expert can be a doctor, a lab manager, a system analyst, or nurse. Understanding the workflow of the hospital is the key explicit requirement that should be available in any participant. Selecting the right case study is a significant task as this supports the possible generalisation of the developed service identification framework to be used in other domains. Consequently, the next section presents the basis of selecting the research case study, which includes all the requirements for the BPMiSearch framework development and evaluation that should be provided by a case study in order for it to be considered as sufficient and representative. The CCR case study is examined against the list of requirements to determine if the case study is representative and sufficient.

3.4.1. The Basis for Selecting the Research Case Study

In the context of DSRM, evaluation is a necessary phase to validate the resulting artefacts (Ahmad, 2016). As mentioned in the previous section, a list of requirements should be identified to examine the candidate case study such that satisfying these needs identifies the representativeness and sufficiency of the chosen case study. These requirements are identified based on the needs of this research project so that the BPMiSearch can be evaluated. For example, to evaluate the performance of the search process, the focus would be on the scale and complexity of the case study. The main requirements to be satisfied and also the driver of each requirement are presented as follows:

1) **Comprehensiveness**: The case study should comprise all the basic features and business process elements that are necessary for the initiation of the interactive search-based framework, such that SOA candidate services can be derived from its associated role-based BPMs. The main driver of this requirement is the need to cover all the possible elements that can be found in different research or industrial domains. Satisfying the evaluation using a comprehensive case study (i.e., has all the basic types of elements) implies that the developed method is able to utilise all different types of elements. In addition, this test provides evidence that the developed method is capable to fulfil the service identification regardless of the domain (i.e., can be used in other domains). Satisfying the first

sufficiency requirement needs the case study to include the set of elements presented in Table 2.1.

2) **The Scale and Complexity**: these factors are important to assess if the case study is sufficient and representative. The driver to check the scale and complexity is the need to assess the performance of the developed service identification approach with large and complex examples. The scale presents the size of the case study (i.e., small, medium, or large), whereas the complexity indicates the interconnection between the business process elements (i.e., using messages and flow elements), which helps to demonstrate the robustness of the case study. A large case study with high complexity is more likely to be adequate and representative as it illustrates and covers more potential scenarios compared to small and simple case studies. Adopting a real-life case study for an enterprise supports the construction of a comprehensive example that reflects an actual scenario. From the search point of view, a complex and large-scale case study helps to monitor and analyse the performance of the search algorithm.

   The scale and complexity of a case study can be examined using different measures (Gruhn and Laue, 2006) that include:

   (i) Size of models: these can be measured by calculating the number of activities.

   (ii) Control Flow Complexity (i.e., CFC): the number of linearly-independent paths through a program (Gruhn and Laue, 2006). This can be defined as the number of abstract states that have to be contemplated when the process is developed (Cardoso, 2005).

   (iii) The number of decisions: making a decision indicates performing a set of activities and transactions to obtain information or change the state.

   (iv) The number of roles involved in the process.

3) **Flexibility**: this aspect determines the capability of allocating BPM business process elements within different candidate services without affecting the actual workflow. The main reason to assess the

flexibility is to discover whether the elements in the case study can be prepared at a different granularity level when constructing the Search Space Elements (SSEs). Using SSEs to build the initial search space is expected to improve the performance of the search while protecting the logical sequence of activities. In addition, the flexibility of the business process elements helps to build reusable services. To assess the flexibility, different characteristics should be observed such as: (i) the involvement of different roles (i.e., participants or departments) such that the activities are distributed in different pools and lanes (ii) models and activities should be connected in a chronological sequence using sequence flows, message flows, and associations (iii) the divergence and convergence of sequence flows should be controlled using different gateway types as well as events that happened during the course of a process. From the search algorithm point of view, the flexibility of the case study models supports identifying more feasible solutions with a different fitness. This helps to find candidate solutions with better performance.

4) **Availability of Domain Experts**: the interaction with the BPMiSearch framework is a key activity to derive candidate services. This activity requires the access to domain experts who comprehend the workflow and have the capability to interact with the search framework to drive the trajectory of the search. In addition, domain experts would reflect on the quality of the resulting software services in comparison with results from other techniques (e.g., non-interactive search) in order to show the impact of the human interaction on the results. Moreover, human domain experts would be asked to show their level of satisfaction with the resulting solutions. All these activities (i.e., interacting with the search and evaluating the results) makes the access to domain experts a key requirement that makes the case study sufficient and representative. Without access to human resources, the case study is not considered as representative.

3.4.2. The CCR Case study: Overview and Adoption

The CCR case study (Aburub et al., 2008; Yousef et al., 2014) has been adopted to demonstrate the service identification processes. The CCR case study supports the traceability of the behaviour of activities that are conducted within different DSRM iterations. It provides a comprehensive representation of these activities within the roles as well as the conversation between these roles.

Choosing CCR BPMs for demonstration and evaluation of the resulting artefacts contributes to the domain of "SOA and Healthcare" (Yousef, 2010). This means that SOA can be utilised to help organisations in the healthcare domain to improve their efficiency and operational capabilities as well as effectively manage their costs. Therefore, selecting the CCR case study to be used by the Interactive Search Framework supports using SOA in the healthcare domain in Jordan (Yousef, 2010).

The CCR represents a real-world example for the King Hussein Cancer Centre (i.e., KHCC) in Jordan (Aburub, 2006). This case study has been validated and improved by previous research attempts (Aburub, 2008; Yousef, 2010; Odeh, 2015; Ahmad, 2016). Role Activity Diagrams (RADs) were used by (Aburub, 2006) to construct the BPMs. These diagrams were migrated to BPMN 1.0 by (Yousef et al., 2009) and then upgraded to BPMN 2.0 by Ahmad and Odeh (2012) such that more tools can be used to work with these models.

Based on the definitions of representativeness and sufficiency, the CCR case study is considered as a representative and sufficient case study. To assess this assertion, the CCR case study should be able to satisfy the requirements presented in the previous section (Section 3.4.1). Therefore, the characteristics of the CCR case study are examined as follows:

1) **Comprehensiveness**: CCR case study comprises all types of BPMN business process elements as expressed from the BPMN metamodel. Figure 3.5 depicts a partial process of the CCR case study in which the patient interacts with the Admission Clerk. Deriving BPMN elements starts from the

Intermediate Catch Event (L12) in the Patient lane, and all the subsequent elements are recorded while following the chronological sequence.

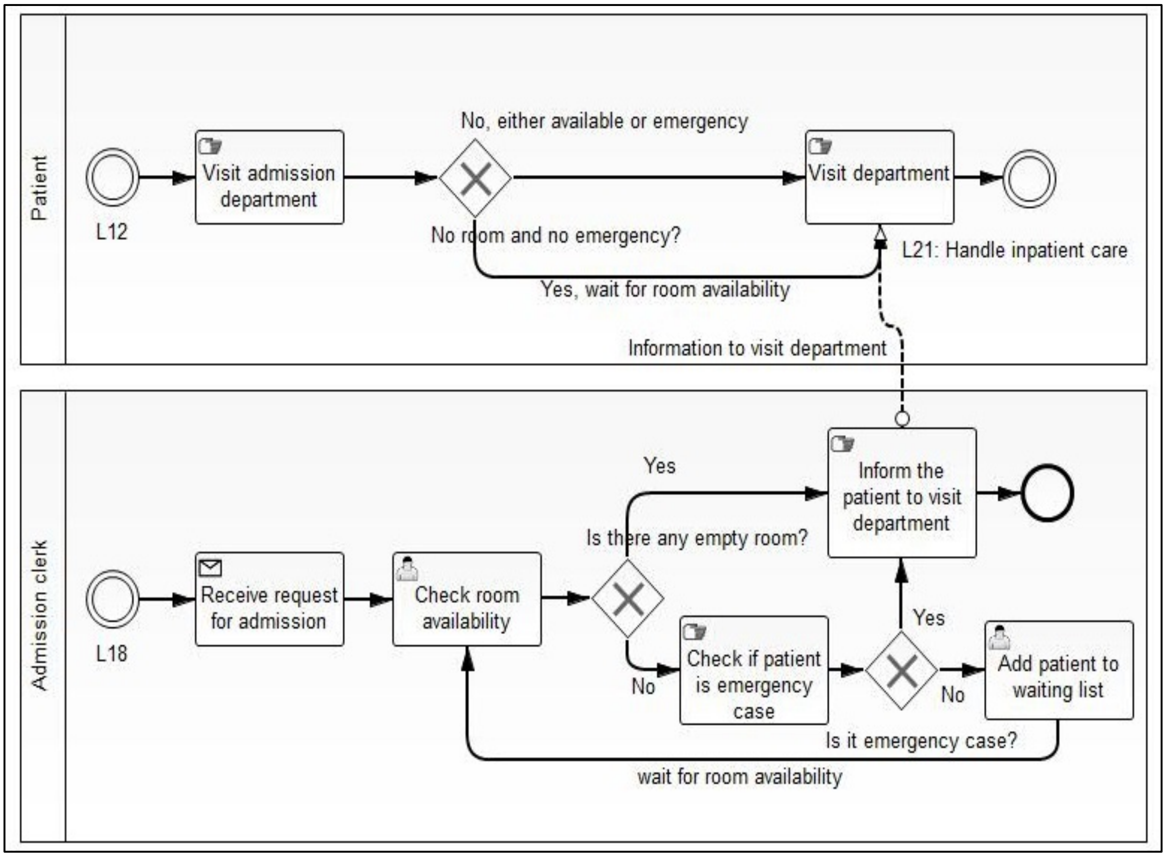

Figure 3.5: Manage Patient Admission Model of CCR Case Study (Ahmad, 2015) used with author's permission.

Table 3.2 shows the resulting elements derived from the Patient-Admission process. The process name, business process element, and element type are presented in the table. The resulting elements include the essential types presented in the BPMN metamodel (i.e., see Table 2.1) that include activities (i.e., user task, manual task, and send task), Events, Gateways, and Flow Objects.

Table 3.2: Business Process Elements of Handle Patient Admission Model

Process	Element	Type
Handle Patient Admission	Pool: CP11	Pool contains two lanes
	Lane: Patient	Swimlane
	Event L12	Intermediate Catch Event

	Activity: Visit admission department	User task
	Gateway	Exclusive Gateway
	Activity: Visit department	Manual Task
	Event L21: handle inpatient care	User task
	Lane: Admission clerk	
	Event L18	Intermediate Catch Event
	Activity: Receive request for admission	Receive Task
	Activity: Check room availability	User Task
	Gateway	Exclusive Gateway
	Activity: Inform the patient to visit department	Manual Task
	Activity: Check if patient is emergency case	Manual Task
	Gateway	Exclusive Gateway
	Activity: Add patient to waiting list	User Task
	Message: Inform the patient to visit department	Message flow to inform the appointment details
	Event	End event

Collecting statistical data from the entire modules, Table 3.3 shows some statistics from the case study (i.e., the BPMN models of CCR are presented in Appendix A, and a list of search space elements (SSEs) with detailed business process elements of CCR is presented in Appendix B). The first observation is that the CCR contains all the basic elements provided by the BPMN 2.0 metamodel (e.g., see elements from only one model presented by Table 3.2). Secondly, the number of elements of each type is large (e.g., 227 activities, 65 gateways), which clearly shows that the CCR is a large-scale case study.

In studying these statistics and observing the derived elements, it is concluded that the CCR case study is a comprehensive case study that includes all the basic types of BPMN elements that can be found in any other case study. Therefore, the first requirement is satisfied.

Table 3.3: Statistics of CCR Case Study

Element	Type	Count
Activity	Send Task	36
	Receive Task	32
	User Task	67
	Manual Task	92
Event	Start	16
	End	25
	Intermediate Throw	43
	Intermediate Catch	22
Gateway	Complex	5
	Exclusive	59
	Parallel	1
Data and Flow Objects	Message Flow	36
	Message	36
	Association	1
	Data Object	45
	Sequence Flow	410
Pool and Participation	Pool	18
	Lane	33

2) **Scale and Complexity:** To assess the scale and complexity of the CCR, the test is based on the criterion defined in Section 3.4.1. Table 3.4 presents statistics derived from the CCR case study which reveals that the CCR contains 227 activities, 162 independent paths, 26 decisions, and 18

different roles (i.e., participants or departments). Based on these statistics, and referring to the inspection criterion, the CCR is considered a complex and large-scale case study. This conclusion indicates that the scale and complexity requirements are also satisfied.

Table 3.4: Scale and Complexity of CCR Case Study

Element	Count
Size of models (number of activities)	227
CFC (number of linearly-independent paths)	162
Number of decisions (number of process completions)	26
Number of roles involved	18

3) **Flexibility:** based on the criterion described in the previous section, it is observed that BPMs of CCR contain multiple pools and lanes that belong to various roles in which the business process elements are distributed. The flow elements (e.g., activities and gateways) are connected using sequence flows, messages, and associations, whereas the gateways and events enclose the activity elements and identify the required paths for the workflow. All the business process elements are connected in a chronological sequence, such that following this sequence leads to traversing the entire elements in the case study models. It is observed that the CCR case study fulfils the flexibility criterion and satisfies this requirement.

4) **Domain Experts:** the CCR case study has been developed at the beginning by researchers and domain experts at KHCC. These domain experts have presented a continues support for research projects that benefit the Cancer Care Informatics at KHCC, which resulted in conducting different research projects at KHCC such as (Aburub, 2006; Yousef, 2010, Odeh, 2015; Ahmad, 2016). KHCC management and domain experts have shown their willingness to support this research as well as any further research attempts in the future, which satisfies this requirement.

Fulfilling the requirements of comprehensiveness, scale and complexity, and flexibility, it is concluded that the CCR case study is considered a sufficient and representative case study. It includes all the essential features and elements derived from the BPMN metamodel; the scale is sufficient and the complexity is high, and in addition, the case study allows for high flexibility to work with the business process elements. In addition, the CCR BPMs cover all the activities at the KHCC, and it has also been extensively utilised, evaluated, and improved to evaluate previous research attempts (Aburub, 2006; Yousef, 2010, Odeh, 2015; Ahmad, 2016), and this results in a rigorous case study.

Moreover, it is important to acknowledge that the characteristics of this case study can help to use data segments of the CCR case study through the desired iterations of DSRM. For example, some data segments will be used by domain experts during the first iteration to develop the base of BPMiSearch framework. In the second iteration, the entire CCR processes and elements will be used as an input to the search algorithm. Having the human 'in-the-loop' in the third iteration, domain experts can evaluate some of the data segments of the CCR processes. The flexibility and comprehensiveness of the CCR support handling all different ways of using the CCR models and elements.

3.5. Summary

The proposed Interactive Search-based Framework (BPMiSearch) is a design artefact having the capability of deriving SOA services of an enterprise from its role-based BPM. The input to this framework is role-based BPM, and in particular a BPM format. The output is an SOA model that comprises a set of candidate services. In this chapter, the adoption of selecting the Design Science Research Methodology (DSRM) as a research framework is discussed. Then the basic requirements of BPMiSearch have been specified according to the research objectives in light of the conclusions drawn in Chapter 2. Correspondingly, the characteristics of BPMiSearch have been derived based on the research requirements, aims and objectives, as well as the research methodology that is presented in Section 3.3 using the DSRM paradigm.

In other words, this chapter has outlined the clear objectives of a solution in design science research which is the second step in the DSRM model by Peffers et al., (2006). This has paved the way for describing the foundations of BPMiSearch as a generic layered interactive framework to derive SOA candidate services of an organisation from its associated BPMN models. In addition, this framework adheres to the key SOA principles such as low coupling and high cohesion.

This chapter briefly introduces the main phases and iterations that are required to develop the BPMiSearch framework. The development of this framework is distributed over three key iterations with a specific objective for each of these iterations. The outcomes of each iteration feed into the next iteration. The incremental nature of the DSRM framework supports the development, demonstration, and evaluation of each added feature for the BPMiSearch framework, which results in producing a high-quality service identification framework. It is anticipated that using this framework to develop the BPMiSearch would fulfil the research requirements and leads to satisfying the research hypothesis.

Finally, this chapter discusses the main aspects of the CCR case study and justifies the reasons for selection. The sufficiency and representativeness of the CCR case study are discussed in detail to ensure that the BPMiSearch framework can be generalised and used in other potential domains.

Chapter 4
The Service Identification Framework

4.1. Introduction

After outlining the design of the Business Process Models Interactive Search (BPMiSearch) Framework and describing its layers and characteristics in Chapter 3, this chapter aims to present the basis of the generic Service Identification Framework, which is one of the unique contributions. The development of BPMiSearch Framework has been divided into three DSRM iterations, the 'Initial Design', the 'Search Process', and the 'Interactive Search'. This chapter begins the initial design phase of the adapted DSRM (Peffers et al., 2006) as presented in Section 3.3. This iteration presents the design and development of a comprehensive layered framework to derive candidate SOA services from role-based BPMs. This includes a set of activities such as providing a systematic approach to traverse the BPMs and record the required business process elements, prepare them in a generic and abstract format with a proper level of granularity, allocate the business process elements into candidate services, and finally, refine the output data to map each business element to the corresponding service component. However, the activities of the service identification layer will be performed manually in this iteration as a proof of concept. The second iteration focuses on improving the service identification process by using a search algorithm, which automates the activities of this phase as well as the check for feasibility and quality of resulting services. Finally, the third iteration upgrades the search algorithm (i.e., presented in the second iteration) by using an interactive search, which is anticipated to enrich the quality of resulting services as well as the satisfaction level of domain experts. Therefore, the first research question (RQ1) is aligned with this objective. RQ1 states:

"To what extent can role-based business process models such as BPMN 2.0 models be mapped to services following Service Oriented Architecture (SOA) principles?"

Satisfying RQ1 requires the application of a top-down approach to derive SOA candidate services from a role-based BPM (e.g., BPMN). Resulting candidate services should be feasible and adhere to SOA principles (e.g., loose coupling and high cohesion) (Lima and Huacarpuma, 2015). Addressing this research question triggers a set of sub-questions such as: (i) what are the business process elements that should be derived (i.e., based on the BPMN metamodel)? (ii) how to prepare these elements at an appropriate level of granularity? (iii) how to allocate these elements in candidate services based on the relationships between them? and (iv) how to measure the fitness of the resulting services?

The second research question (RQ2) states:

"***In what way can SOA services be best implemented for metaheuristic search?***"

RQ2 utilises the outcomes of answering RQ1 by means of an evolutionary computing technique (i.e., a genetic algorithm) as the basis for automating the service identification problem. Addressing RQ2 necessitates the examination of the different representation techniques to find an appropriate method to represent the candidate services and construct the search space (Jamshidi et al., 2012). The outcomes of addressing RQ2 are the key components of the genetic algorithm that include the representation along with the associated genetic operators (i.e., selection, recombination, and mutation), and the fitness measures (Harman and McMinn, 2010).

This DSRM iteration involves three phases: design and development (Section 4.2), demonstration (Section 4.3), and evaluation (Section 4.4). In the design and development phase, the initial design of the service identification framework is presented as well as the adoption of a representation technique. In the demonstration phase, a case study is utilised to exhibit the service identification framework. Following this, in the evaluation phase, the validity of BPMiSearch is examined by applying validation and verification checks in Section 4.4.

4.2. DSRM Design and Development Phase

In this chapter, a new approach has been introduced to describe the current research attempt to bridge the gap between the business process models and SOA services. Based on the proposed approach, a new architectural framework, namely the BPMiSearch, which uses BPMs to derive SOA services is introduced. One of the main characteristics of this framework is the need for it to be comprehensive (i.e., it manages all the activities of service identification) and generalisable (i.e., can be extended to manage different role-based BPM languages such as BPMN, RAD, and UML AD). Figure 4.1 presents a generic architectural design of the BPMiSearch Framework for service identification.

BPMiSearch consists of three logical layers: (i) Input BPM Preparation and Generalisation (ii) Service Identification, and (iii) Output Services Refinement. Each layer has a specific input and processes it using its internal computational components to generate a specific output which feeds into the lower layer. In addition, the BPMiSearch comprises a metamodel that defines the syntactic and semantic guidelines in order to translate the input role-based BPM at the corresponding layers. The design of this framework promotes the reusability of outcomes as well as the flexibility of the input modelling language. The following subsections describe in more detail each of the BPMiSearch individual layers and their essential components.

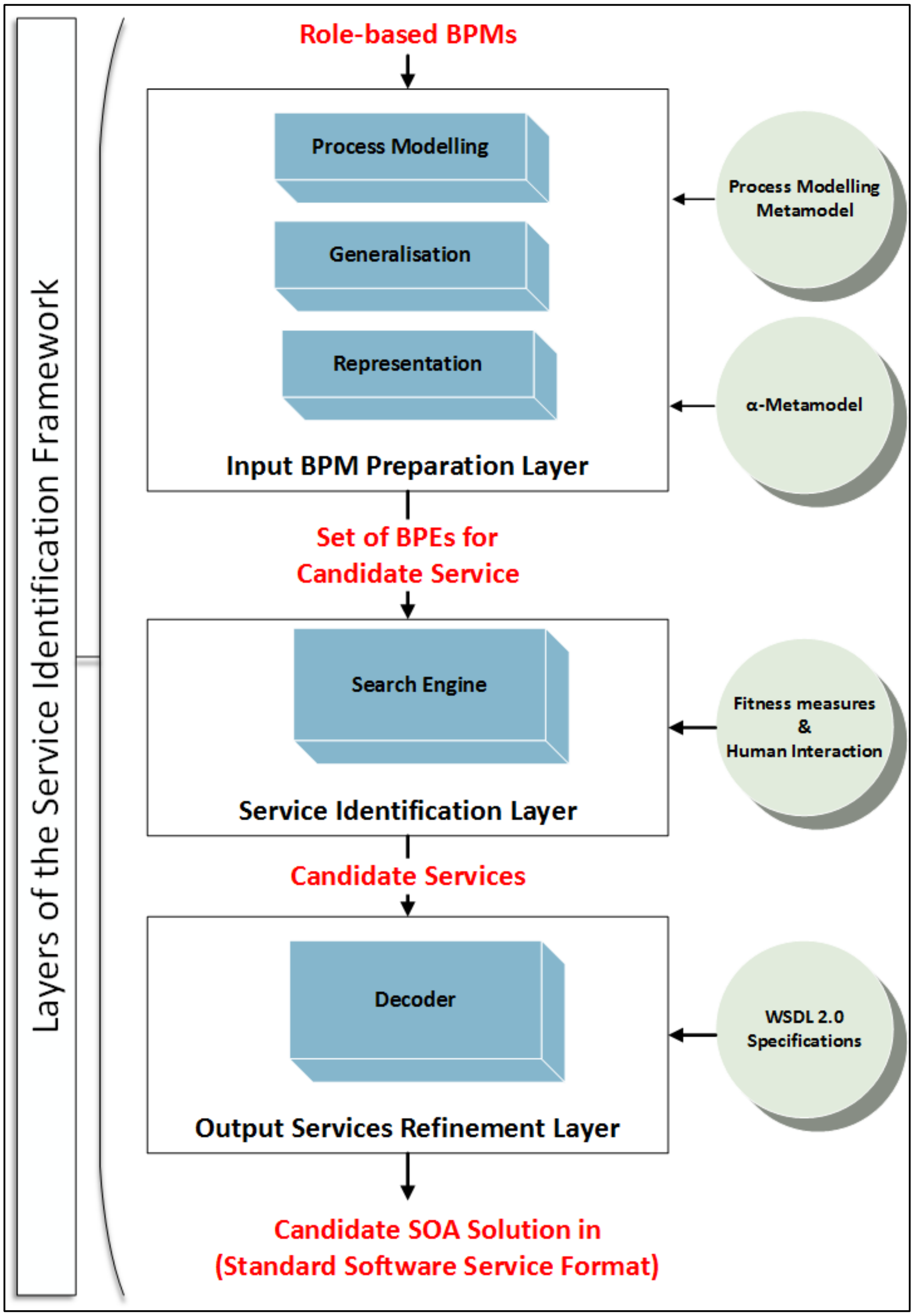

Figure 4.1: The Architectural Design of the BPMiSearch Framework

4.2.1. Input BPM Preparation Layer

The purpose of this layer is to prepare the input BPMs in the required format that constructs the search space. The main advantage of this phase is to provide the capability to generalise the framework such that different modelling languages (e.g., BPMN and RAD) can be used to derive SOA services. Subsequently, the outcome of this layer is an initial population that is required to apply an interactive search to derive feasible candidate SOA services. The following sub-sections describe each activity in more detail.

4.2.1.1. Business Process Modelling

This layer deals with the modelling characteristics of a business process. The purpose of this layer is to use the input BPM (e.g., BPMN) to generate a generic form of process model that is used to construct the search space elements (SSEs) in the next step. In addition, data about each business process element are recorded to help the service identification activities such as calculating the fitness values for candidate services. The recorded details include the role, type, ID number, and interaction details (i.e., source and destination of each connection). Since role-based business process modelling languages are syntactically different but semantically similar, these common concepts can be generalised and given consistent names for a particular modelling approach (Khan, 2009). These common names can be consistently used at the lower layers of the BPMiSearch framework as the SSEs interact with each other to perform business functionalities.

For the purposes of this research and also the service identification needs, the key set of business process elements will be encapsulated with new attributes. Each type of attribute represents a label that includes a set of elements that are semantically similar. As a result, unified types will be produced using a shared syntax. For example, Task in BPMN and Action in the UML diagram can be categorised as one category (i.e., flow element) (Geambasu, 2012). The same applies to Pool in BPMN or Swimlane in UML AD as both belong to one category (i.e., Pool). The main attribute tags that will encapsulate these business elements are Flow Element (FE), Gateway (GW), Event (Event), Data Object (DO), Pool (Pool), and Interaction Element (IE).

The generic metamodel (i.e., α-Metamodel) has been derived to encapsulate the syntactic heterogeneity of different role-based modelling languages. However, in this research, the α-Metamodel is more specific to the BPMN. The specifications of the input BPMs can be derived from the input BPM Process Modelling Metamodel (e.g., BPMN metamodel). Table 4.1 presents the proposed key elements of the α-Metamodel along with the corresponding types from BPMN (i.e., or other role-based modelling languages) and the common tag that will be used for that type of element.

The α-Metamodel is important to identify common modelling constructs that comprise different role-based BPM languages. Hence, there could be multiple implementations of the α-Metamodel where each implementation is appropriate for a particular modelling approach. The α-Metamodel can also be extended to handle specialised constructs in different modelling languages, for example 'Activity' and 'Action Node' in UML AD. The outcome of this layer is a generic business process model represented as sets of business process models in which a mask encapsulates each element to help utilise this element in the next lower layer. In addition, detailed information about each element is recorded to help allocate these elements to the right services.

Table 4.1: α-Metamodel key elements with the included types and tags

Element	Included types	Tag	Example Business Process Elements
Flow Element	Task and Activity elements	<FE>	User Task, Send Task and Receive Task.
Gateway	Gateway types	<GW>	Complex, Exclusive and Parallel
Event	Border events, Intermediate events, and triggers	<Event>	Start, End, Intermediate
Data Object	Data elements, files, Databases, …	<DO>	Files and Documents
Pool	Pools, lanes, swimlanes, …	<Pool>	Process, Pool and Lane
Interaction Element	Messages, association, Message flows, …	<IE>	Send/Receive Messages and Data Objects

Figure 4.2 shows the extraction of the generic data (i.e., the right side) from the original BPMs (i.e., the left side). Each element in the derived data is granted a type and ID number. In addition, data about each element are recorded while traversing the models. Collected data includes the type, ID, source and destination of the connection, role name and ID. Since the business process elements will be prepared in

a new level of granularity, the role that is assigned to each element is recorded, but the swim lanes and pools are removed to allow changing the granularity level in the next step.

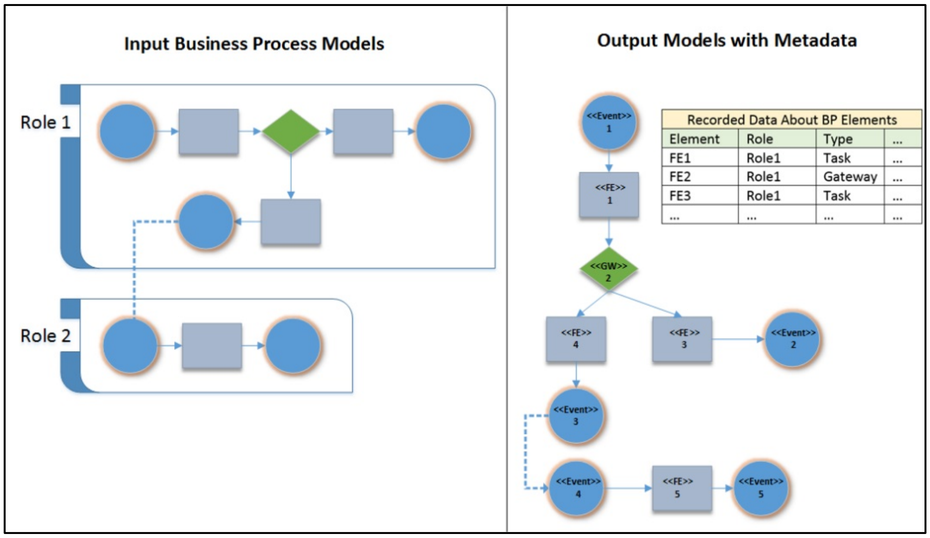

Figure 4.2: Modelling the Business Process Elements

4.2.1.2. Abstraction and Generalisation

The purpose of the abstraction layer is to prepare the generic form of BPMs at an intermediate level of granularity (i.e., not as fine-grained as single tasks, nor as coarse-grained as roles of the input BPMs). The outcomes of this layer are a set of Search Space Elements (SSEs) that encapsulate the business functions. This layer is very important because it supports generalising the service identification framework. The level of granularity indicates the scope of functionality that is implemented by the resulting candidate services (Papazoglou, 2003). On the one hand, coarse-grained services support encapsulating the business activities and hide the conversation between services (Poorazizi et al., 2015). In addition, the opportunities for services to be used in different contexts would be higher because they perform different business functionalities. On the other hand, finer-grained services support the composability and flexibility principles of SOA (Papazoglou, 2003).

Preparing the raw business process elements at an intermediate level of granularity is not only required to generalise the framework, but it also has many advantages. Firstly, it protects the main structure of the business functions in the same chronological sequence which maintains a clear purpose for each building block (Alghamdi, Potter and Drew, 2016). Secondly, it increases the search efficiency (i.e., speed) by using a smaller number of elements to construct the search space. Utilising the SSEs as the building blocks of the search space rather than the raw business process elements for the search is more efficient. Thirdly, it reduces the complexity of the search and enriches the quality of the resulting solutions. If the search space contains a large number of elements, the potential number of resulting services is huge as the possibilities of placing elements in candidate services would be very large. However, grouping these raw elements as abstract level elements (i.e., SSEs) would effectively support reducing the complexity of the search. This also helps to examine the feasibility of the resulting solutions.

Grouping the raw BPM elements to construct SSEs is governed using a set of constraints that identify the characteristics of the SSEs. The main characteristics of an SSE are that it includes a set of flow elements (i.e., one or more) that are connected chronologically (i.e., using sequence flow elements) to perform one business service. An SSE should take the roles and tasks into account, such that each SSE performs one task (i.e., task-oriented) and belongs to one role. Based on these characteristics, the key constraints to create an SSE are (i) the flow elements that comprise an SSE should be connected chronologically, (ii) flow elements should belong to the same role, and (iii) each SSE should perform one business functionality; this can be identified using the surrounding elements (e.g., lanes and events) that help to recognise the borders of each SSE.

This layer takes the α-Metamodel as input which describes the semantics and syntactic characteristics of the input models. In this layer, a systematic approach is utilised to traverse the input BPMs and record the raw business process elements. A systematic approach has been developed to traverse input BPMs to construct the SSEs. This approach is described as follows:

(i) Initiate an SSE (SSE_n) to allocate BPMN elements.

(ii) Start reading the input BPMs one by one and for each model begin with a start event (i.e., it could also be an Intermediate Catch Event), and follow the chronological sequence of elements.

(iii) Record each flow element (e.g., activities, tasks) identifying that activity type, entire role, process, source, and destination. Then, observe the activities that make the transactions and trade messages between different roles.

(iv) Record the sequence flow elements identifying the source and destination.

(v) Allocate the activities, gateways, and sequence flow in the current cluster.

(vi) Observe the border elements and events and record the start and end points with all enclosed elements. Initiate a new grouping element (i.e., SSE_{n+1}) and use it to allocate the next element.

Traversing all the input process models, it is anticipated to have a set of SSEs that group the entire flow elements, as well as a set of sequence flow elements that connect the resulting SSEs.

Figure 4.3 and Figure 4.4 present the proposed algorithm that aims to traverse the input BPMN models, record the needed data and group the business process elements into SSEs. The final outcome of this algorithm is a set of SSEs that are connected using message flows and sequence flows. The algorithm is presented in two figures to make it more readable. The first part of the algorithm (Figure 4.3) loops on all the input models and passes them one by one to the second algorithm. The return value from the second part of the algorithm (i.e., the resulting SSEs and connections) are stored in an array. The second part of the algorithm (Figure 4.4) loops on elements inside the current BPMN model. It follows the chronological sequence and searches for the border elements such as 'Start Event', 'End Event', 'Intermediate Event', and 'Pool', and groups the elements between two border elements in the current SSE and initialises a new SSE. Arriving at the final element, the resulting set of SSEs is returned to the first algorithm.

```
Algorithm I: Deriving BPMN Elements
Input:
    ➢ Set of BPMN models, BPMN_Set = {M0, M1, …, Mk}
Output:
    ➢ Set of Search Space Elements (SSEs)
Begin
  For each business process model Mx in BPMN_Set
        Call Create_SSEs( Mx)
        Add SSEs Elements to the set of SSEs
  End For
End
```

Figure 4.3: Algorithm I Deriving BPMN Elements

```
Algorithm II: Create_SSEs
Input:
    ➢ Business Process Diagram Mx
Output:
    ➢ Set of resulting SSEs
Begin
  For each element in the business process
        Current_element = first element
        Initialise SSE_Set
        Find all connection objects conn_set = {conn0, conn1, … connk}
        For each connj in conn_set
          Initialise SSEn
          if Current_element is not added to elements_container then
                add current_element to SSEn
                save additional info {participant, data element, connection element …}
                current_element = find_next
          End if
        End for

        if Current_element = Border_element then
                Save SSEn
                Initialise SSEn+1
        End if
  End for
End
```

Figure 4.4: Algorithm II Create Search Space Elements

Figure 4.5 shows an example of how the input BPM raw elements (i.e., produced in the previous layer) would be grouped to produce SSEs. On the left-hand side, the input models are presented with all the available data about the models and elements. On the right-hand side, the resulting SSEs are presented in such a way that they group different elements. Note that interactions between the SSEs are originally granted from the interactions between the elements in the input models. This means that these interactions will also be presented when deriving candidate services.

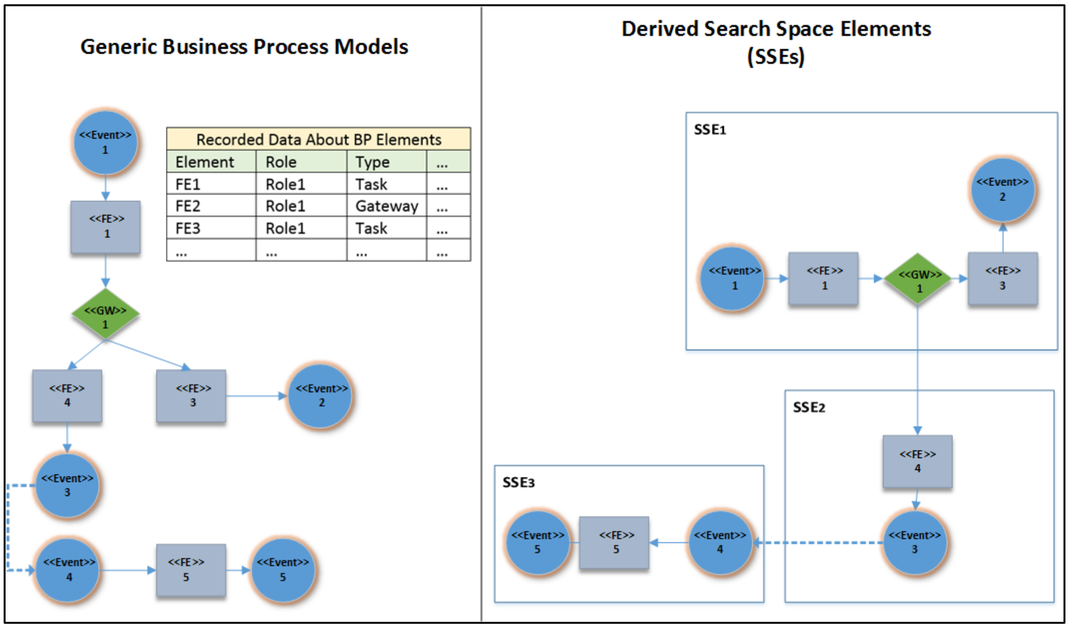

Figure 4.5: Deriving Search Space Elements (SSEs) from the Generic BPMs. On the right-hand side the generic BPMs, on the left-hand side the resulting SSEs.

4.2.1.3. Representation

The aim of computational search and exploration in service identification is to enable the discovery of useful and stimulating candidate software services expressed using BPMN process models which are demonstrably traceable to the stated service identification problem. To underpin such evolutionary search and exploration of the software service identification solution space, it is necessary to represent the design problem and the design solution such that they are not only sufficiently abstract for human comprehension but also a natural fit for a computational evolutionary search. A novel implementation of the service identification problem has been formulated. This formulation is based on Bin Packing problem such that a natural abstraction of the essential characteristics of the software service identification solution is provided. This implementation directly facilitates effective visualisation for the interacting human designer (Bozkurt, Harman and Hassoun, 2013). An integer-based implementation is used to implement the problem and represent the BPEs. The representation and the associated operators should consider the input as the business process and the output as service elements (Jamshidi et al., 2009). In addition, the anticipated representation is required to:

1) Satisfy the notion of services and conform to the principles of SOA.
2) Define a clear form of encoding the problem elements to genotype (i.e., basic data type or structure that represents the underlying genetic coding) elements, and then decode from genotype to real-world elements or phenotype (i.e., the solution within the original problem context).
3) Enable efficient genetic operators (i.e., selection, recombination, and mutation).
4) Enable an obvious criterion for evaluating the resulting solution.

The service identification problem can be implemented and solved as a Bin Packing problem (El Hayek, Moukrim and Negre, 2008). Bin packing aims to group a set of objects into the minimum number of bins (i.e., to use the space or time efficiently). One of the key constraints is that the overlapping between different bins is not allowed (Berkey and Wang, 1987). The classical version of the Bin Packing problem aims to minimise the number of bins. However, real-world problems are more complicated and difficult. Bin Packing is one of the mathematical problems known as NP-complete (Berger and Leighton, 1998; Chekuri and Khanna, 1999; Korf, 2003) because developing a method to solve it optimally or find the best optimal solution is still not possible, even for a medium-sized problem. The most relevant generalisation of the Bin Packing problem is the two-dimensional (2BP) version, which aims to load a set of rectangular items into larger identical bins with the aim of minimising the number of bins without overlapping loaded items (Li et al., 2014). This problem is widely used in industry, for example, container loading in transportation operations, the paging of articles in newspapers, and cutting material (e.g., fabrics, paper, glass and wood) in order to minimise the amount of material and consequently reduce the cost (i.e., time and money). Nevertheless, each practical problem has its own specific features (El Hayek, Moukrim and Negre, 2008).

Bin Packing can be utilised to implement the service identification problem because service identification aims to pack the business process elements (i.e., BPEs) into candidate services (i.e., bins). However, in service identification, the goal is to maximise the effectiveness (i.e., produce solutions with a high fitness value) and minimise the cost (i.e., find good-enough candidate solutions in a short time). Therefore, the service identification adapts the basic Bin Packing problem but applies its own specifications. The adaptation of Bin Packing should enable the representation and allocation of the BPEs within bins (i.e., candidate services). Therefore, the key objective of the data preparation layer of the proposed service identification framework is to formulate and prepare the input data (i.e., BPMN models) in an appropriate format, such that they can be allocated in candidate services. The implementation of Bin Packing constraints (i.e., no overlapping of elements between two candidate services and the removal of the empty services) can ensure the feasibility of the resulting candidate services.

In this research, the encoding and decoding mechanisms should be utilised to allocate the input business process elements derived from BPMN models into candidate services. This encoding and decoding (mapping the business functions into IT services) will help to allocate elements that perform together in candidate services such that the resulting IT service-based system will be more effective and efficient. In this chapter, the concept of representation is discussed, whereas, Chapter 5 describes the technical and implementation details of the representation and genetic operators. Choosing the right representation is one of the most critical factors informing the success of an evolutionary algorithm (Simons, 2011; Kazemi et al., 2011). In many cases, there are different candidate representation options, but having the experience and knowledge of the problem domain helps to choose the right one (Eiben and Smith, 2003). Examples of different representations include binary strings, integers, real-valued, permutation, and tree representations (Eiben and Smith, 2003; Simons et al., 2010). The capability of handling the constraints that ensure the feasibility of resulting solutions is an important factor when selecting an appropriate representation method (Simons et al., 2011).

This research adopts an integer representation approach that is inspired by Bowman, Briand and Labiche (2010) to represent the service identification problem. This representation combines a vector of integer values (Ahmed and Deb, 2013) with a string of values (Whitley, 1993) to formulate a more powerful method that fulfils the service identification problem. Chromosomes basically comprise only integer values, where the position of the gene within the chromosome represents the SSE, and the gene's integer value denotes their service assignment. Each position in the vector is assigned to one SSE. In this way, it is impossible to have an empty service represented in the chromosome. For example, Table 4.2 denotes that the SSE_1 (i.e., the first gene) belongs to the 2nd service, whereas, SSE_2 belongs to the 6th service. The resulting solution presented in this example is represented by the vector (2, 6, 6, 2, 1, 5, 3, 4, 1, 3). Since the chromosome representation assigns an SSE to each gene within the chromosome, the length of the chromosome consequently equals the number of SSEs in the search space. The positions of the elements in the first row are static and will not change. Instead, the genetic operators (i.e., crossover and mutation) would be applied to the gene position (i.e., the second row).

Table 4.2: Representation example

SSE_1	SSE_2	SSE_3	SSE_4	SSE_5	SSE_6	SSE_7	SSE_8	SSE_9	SSE_{10}
2	6	6	2	1	5	3	4	1	3

Adopting this representation method efficiently utilises integers to represent service IDs, which is one of the most natural problem representation methods that naturally takes the form of grouping elements together (Back, 1997; Eiben and Smith 2003). A key advantage of this representation technique is the implementation of a check for constraints that guarantees the feasibility of the resulting SOA solutions. For example, resulting services do not lack SSEs, and in addition, each SSE can appear in one service only. This significant advantage reduces the execution time and leads to an improvement in the performance of the search process. According to this representation, the genetic operators (i.e., selection, crossover, and mutation) will be formulated. The implementation of these genetic operators and the fitness function will be presented in the next iteration (Chapter 5).

4.2.2. Service Identification Layer

This layer contains a set of activities to map the SSEs produced in the previous phase to the SOA candidate services. These activities are: (i) allocate these elements in candidate services based on the interaction and consistency between them (ii) measure the fitness values of the resulting solution and refine the services accordingly. The two activities can be repeated multiple times such that the allocation of SSEs would be tuned in light of the obtained fitness values. Using a search-based or an interactive search-based method is effective in performing these activities. However, in this chapter, the activities of the service identification phase (i.e., second layer) will be performed manually by domain experts.

4.2.2.1. Allocating Search Space Elements in Candidate Services

This layer aims to identify candidate services by allocating a set of activities that perform at least one business functionality together (Papazoglou et al., 2009). The input for this layer is a set of SSEs that comprise the search space, whereas the anticipated output is a set of clusters (i.e., candidate services) that group SSEs that perform together to fulfil the business requirements. The interaction between the candidate services aims to satisfy the functionalities of the resulting services.

Finding the best composition of SSEs that comprise the list of candidate services is the key challenge of this layer. Grouping the SSEs in candidate services relies on the relations between these building blocks, which can be assessed using objective fitness measures. Performing one functionality reveals a clear purpose of the service, which supports several SOA principles such as usability, low coupling, and high cohesion (Jafarov and Lewis, 2015). Grouping SSEs into low-coupled and high-cohesive services when proposing sets of candidate services is a critical issue. For example, allocating all SSEs in one service produces the ideal coupling value (i.e., zero), but the cohesion will be very low. In contrast, allocating each SSE in a separate service produces services with high cohesion. But because of the connections between these services, the coupling value will be high. Thus, it is important to design a fitness function that creates a balance between coupling and cohesion values.

The search algorithm would then be used to find the candidate services that guarantee the best fitness values in the resulting candidate services. The goal of the search algorithm is to find an optimisation that satisfies the high-quality measures calculated with the fitness function, which requires the discovery of a trade-off between different fitness measures. The best situation of a resultant candidate solution is when SSEs inside each service have the maximum intra-relations (i.e., internal relations inside a service) and minimum inter-relations (i.e., relations between services). This supports the reusability of the resulting services by producing services with high cohesion and low coupling (Zadeh et al., 2016). In addition, it is anticipated that in an optimised solution the business process elements that perform together will be grouped in the same SSEs and consequently in same services.

4.2.2.2. The Quality of Candidate Solutions

The quality of the resulting candidate solutions can be examined using a fitness function along with the fitness design metrics. The fitness function is the second key component of a search-based method. The effective factors in deriving candidate services that are assessed using fitness function are coupling and cohesion. The choice of these two metrics is due to their strong influence on the granularity level as well as the reusability of the services (Jamshidi et al., 2012; Zadeh et al., 2016). The fitness values highlight the quality of the desired solution by applying mathematical calculations to quantitatively assess the inner relations inside each service (i.e., the cohesion), and the degree of the relationship between one service and other services (i.e., the coupling value), see Jamshidi et al., (2012).

The resulting SOA candidate solutions are evaluated using a combination of fitness measures (Hansen, 2016). The first fitness measure evaluates the structural reliability of a SOA solution. Good SOA designs typically strive for low coupling between services to ensure the design is robust yet flexible enough to change (Simons and White, 2014). The first design metric is inspired by the coupling factor (CpF) (Zadeh et al., 2016). Ideally, the resulting services are stand-alone i.e., they do not need to be connected to other services to fulfil any task. These services score a coupling of zero.

The second design metric adopted is cohesion to reflect the extent that a SOA service has a clear purpose (i.e., service abstracts the underlying business), which can be achieved using the cohesion metric. The cohesion of Service (ChS) metric has been adopted by (Larman, 2012; Chalimeda et al., 2001) to measure the intra-relations of a service. Thus, if the service performs only one function, the cohesion scores a ChS of 1.0. Good solutions are obtained through the minimization of coupling, and the maximization of cohesion.

Indeed, the Coupling and Cohesion metrics have been widely investigated in many surveys, and consequently different potential formulas for them have been suggested as a result (Zadeh et al., 2016; Khlif et al., 2009). A single-objective fitness function that aggregates the values of all design metrics in a single value is implemented (Praditwong et al., 2011; Vanderfeesten et al., 2007). The goal is to maximize the weighted sum of all the quality metrics (Kessentini et al., 2014).

The candidate services exposed after applying the search method include a list of tasks for each service. Each service is specified using a service number, names of the related business process, and detailed tasks and activities inside the candidate service.

1) Coupling: The coupling metric measures the relative degree of interdependence among services (Papazoglou, 2006), by quantifying the extent to which the services within a solution are interconnected (Khlif et al., 2010). The conversation between the services is implemented using

send/receive messages. The number of messages and the size of them are important factors in representing the degree of coupling between these services (Zadeh et al., 2016). Although this is a simple count, it is very useful in identifying real-world problems by demonstrating the effectiveness of the general strategy (Jamshidi et al., 2012). Based on this, the coupling factor of a service X, denoted by CpF(x) is formulated in equation 4.1.

$$CpF(x) = \frac{I_x}{SSE_x + I_x} \tag{4.1}$$

I_χ represents the number of interactions with other services, and SSE_χ is the number of the SSEs inside the service. For example, if a candidate service comprises six SSEs, and calls three other services, then the coupling factor $CpF(x) = \frac{3}{6+3} = 0.33$.

In another case, if a candidate service comprises a number of SSEs but does not initiate any interactions (i.e., stand-alone service), the coupling factor will be zero. It is advantageous to minimise coupling between activities within different services to conform to the SOA principle of loose coupling (Papazoglou, 2003). This principle helps to minimise the dependency between services as the increase of coupling affects the reuse and composability of services as well (Khlif et al., 2010). Moreover, the performance of the systems deteriorates when more messages are exchanged providing a guide to a larger communication overhead (Pimentel and Nickerson, 2012). In contrast, tight coupling makes it difficult to isolate the services when a change is implemented in a small part of the service solution. Therefore, the elimination of unnecessary relationships and dependencies between services is very useful. Although minimising the coupling to the lowest possible level is encouraged, some level of coupling is unavoidable and cannot be considered in isolation (Khlif el al., 2010). Indeed, allocating all activities within a single candidate service would reduce the coupling to zero, but in practice, this cannot be considered as the best solution (Jamshidi et al., 2009).

2) Cohesion: This metric indicates the degree of strength of relationships between the operations of a service (Reijers et al., 2004; Jamshidi et al., 2012). Within a service, each SSE is strongly connected to other SSEs if their chronological relationships have a strong dependency (Jamshidi et al., 2009). The impact of grouping different tasks in one candidate service is minimising the cohesion of that service as it no longer focuses on a single functionality. Nevertheless, the strongest cohesion can be achieved when a service focuses on one conceptual task (Khlif et al., 2009). According to this description, the internal dependency (i.e., cohesion) metric helps to optimise the correlation of SSEs such that each candidate service encapsulates the more relevant SSEs. The formula for Cohesion of a Service x, denoted by ChS(x), is presented in equation 4.2.

$$ChS(x) = \frac{1}{|SSE_x| * |SSE_x - 1|} \sum_{i=1,j=1}^{n} \delta_{ij}$$

$$\delta_{ij} = \begin{cases} 1, SSE_i \ calls \ with \ SSE_j \\ 0, Otherwise \end{cases} \quad (4.2)$$

SSE_x represents the number of elements inside the service, n represents the number of potential connections inside the service, and δ_{ij} represents the existence of a relationship between SSE_i and SSE_j. An example may help to clarify this formula. A service contains three elements S = {SSE_1, SSE_2, SSE_3}, SSE_1 requests data from SSE_2, and SSE_2 requests data from SSE_3. The total number of elements is three, whereas the number of connections between elements is two. Thus, the cohesion of service is calculated as follows:

$ChS(x)$ = (1 / (|3| * |3-1|)) * 2 = (1/6) * 2 = 1/3.

Weighted Sum Fitness Function: The fitness function adopted by the search-based service identification method combines the coupling and cohesion design metrics in a single fitness value. The goal is to maximise the weighted sum of the fitness value. This can be achieved by finding the best allocation of

SSEs in the candidate services that achieves high-cohesion and low-coupling. Examples of using similar approaches are found in Simons et al., (2010) and Seng et al., (2006). The complement to one of the coupling values (i.e., $1 - CpF(x)$) is accumulated with the fitness value as the objective is to minimise the coupling value. A weight is assigned to each design metric before being accumulated to the total fitness value (e.g., 50% each). The weights can be modified based on the experimental design and empirical trial-and-error experiments. Equation 4.3 formulates the accumulated fitness function.

$$Fitness(x) = (Weight_{coupling} * \left(1 - CpF(x)\right) + (Weight_{cohesion} * ChS(x)) \qquad (4.3)$$

An example of calculating the fitness value for a service would clarify the formula. If the $CpF(x)$ = 0.2 and $ChS(x)$= 0.70. Considering that coupling and cohesion have equal weights (i.e., 0.5 each), the fitness value is calculated as follows:

$Fitness(x)$ = 0.5 * (1 – 0.2) + 0.5 * 0.70 = 0.4 + 0.35 = 0.75.

The resulting value (i.e., 0.75) reflects the overall percentage of successful allocation of SSEs in services. This number means that 75% the candidate services comprise relevant SSEs and functionalities that perform as one block. The ultimate goal is to increase this percentage to reach 100%.

4.2.3. Service Refinement Layer

Web services have been defined as web application components that can be published, found and used on the Web (w3schools, 2016). A standard XML-based interface definition language that is used to describe the functionalities of a web service is called the Web Services Description Language (WSDL) (Zheng et al., 2014). The web service is self-contained, modular, distributed, and located locally or web-based (W3Schools, 2015, Papazoglou et al., 2009). The WSDL 2.0 file is a machine-readable description that describes a web service; it comprises all the information on how to call the service, what parameters to pass and what data structures would be returned. Services are also described in WSDL as network endpoints or ports. The web service contains a set of operations (i.e., functions) and interfaces that describe the input and output messages to be exchanged.

In the service refinement layer, the business process elements that comprise the candidate services created at the previous layer are mapped to the corresponding components based on the standards of the WSDL files (Kamienski et al., 2010). The resulting WSDL files are important to implement the services by providing accessible functions and operations (Offermann et al., 2009). The binding details that help to access each resulting service, in addition to the operation names, inputs and outputs, are described by the WSDL files.

The SSEs that construct the candidate service from the previous layer are extracted from the basic business process element and then each element is mapped to the corresponding component of the WSDL file. In this research, a mapping criterion is proposed to match the business elements to the service components. Since this research adopts BPMN, Table 4.3 presents the mapping that associates each type of the BPMN elements with the corresponding entity type of WSDL component. This mapping method is proposed to decode the resulting SOA solutions.

Table 4.3: Mapping the BPMN Elements to WSDL File Components

BPMN 2.0 Element	**WSDL component**
Activity elements (e.g., User Task)	Operation
Data Elements	Service inputs and outputs
Messages	
Events and connection elements	Binding, Interconnections, and communications between services

This mapping locates the basic business entities in the corresponding sections in the WSDL file. Creating the binding between the resulting WSDL files is a key activity to prepare the SOA solution in the final format. Although there is no direct "one-to-one" mapping for every single element in BPMN to WSDL, only the basic BPMN elements are mapped to the corresponding WSDL components. On the one hand, some of these elements are mapped directly, such as tasks that are mapped to operations, and data elements that are mapped to input and output artefacts. On the other hand, other BPMN elements are mapped indirectly, such as gateways, pools, and associations that help to design the flow and interactions between services.

Figure 4.6 presents an example of deriving the service components from BPMN and mapping them to the corresponding service components. This sample data segment from the Cancer Care and Registration (CCR) case study shows the main elements in a service that manages the booking of an appointment. In the candidate service, a patient requests an appointment by sending a request to the receptionist, and the date/time of the appointment is being sent to the patient when the booking is prepared. The first step is to find the corresponding match for each business process element, and then to create the WSDL file in the standard structure.

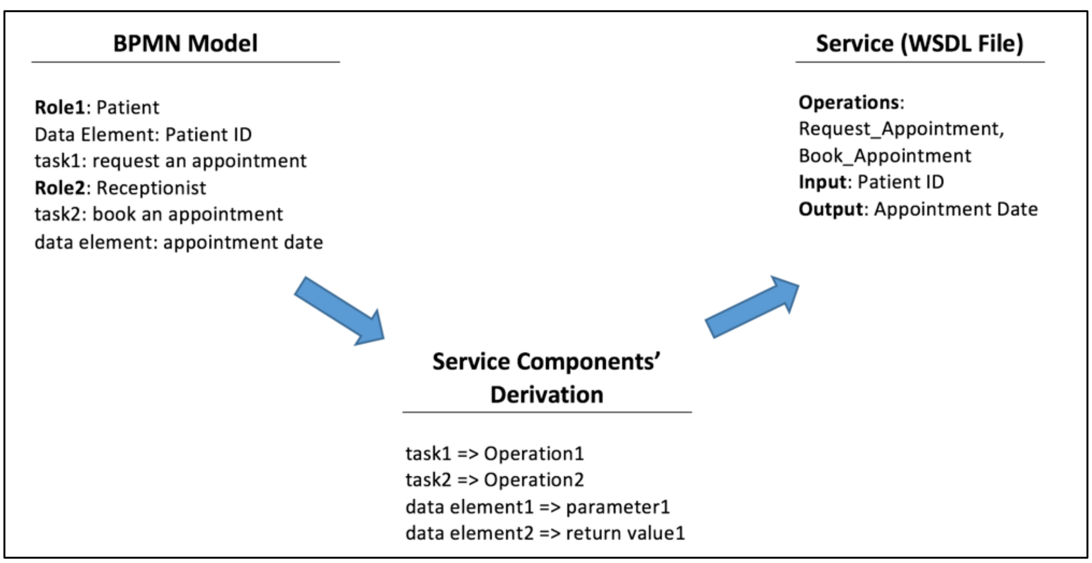

Figure 4.6: Mapping to Service Components Example

The WSDL files produced at this layer comprise the header of the entitled web services, the operations, the binding details (i.e., the connected services), the available parameters (i.e., for inputs and outputs), and the data types (Weissman and Bobrowski, 2009). The next chapter presents a sample WSDL file for a resulting solution after automating the service identification using the search-based method.

4.3. DSRM Demonstration Phase

This section presents the proposed search-based framework by example. The service identification framework is presented using the CCR case study (see Section 3.4). The objective of this DSRM phase is to test the service identification framework using a real-life case study. Therefore, the next section presents an experiment in which a set of CCR BPMN models are used as input in order to derive candidate Cancer Care services.

4.3.1. Experiment: Service Identification Using Domain Experts

This experiment aims to validate the service identification framework by showing how to derive candidate services from the BPMN. However, as a proof of concept, domain experts at the King Hussein Cancer Centre (KHCC) perform the service identification activities manually. The allocation of SSEs in candidate services is based on the relationships and interactions between these elements and relies upon the knowledge and experience of the participant domain experts. This experiment helps to show the capability of the proposed framework to derive feasible service solutions.

To evaluate the service identification framework, the fitness values for the resulting solutions are compared to the fitness values for sample solutions that are produced randomly. The dependent and independent variables for this experiment are presented in Table 4.4.

The input models to be used in this experiment are the BPMN models of the CCR case study (i.e., see Section 3.4. The input models are also presented in Appendix A. Traversing these models and preparing them is a key step in producing the SSEs (i.e., all SSEs are presented in Appendix B) that encapsulate the entire business functionalities.

Table 4.4: Parameters of the Service Identification Experiment

Experiment	Parameter Type	Values
Manual Service Identification Experiment	Independent	▪ Number of services, SSE IDs, Allocation of SSEs
	Dependent	▪ Effectiveness (i.e., in terms of Coupling, cohesion, and fitness value)

With regard to the participants, seven domain experts from KHCC with different levels of experience were recruited to participate in deriving the candidate SOA solutions from the BPMN models. Understanding the interactions between different SSEs relies upon the implicit knowledge and experience of the domain experts. A number of sessions and exercises have been conducted to discuss the project aim and objectives with the participants, and also to help participants to understand the business process models of KHCC. Appendix D presents the materials and exercises that were used to fulfil this experiment.

4.3.2. Results

The BPMN models of the CCR case study have been passed as input to be traversed and prepared as SSEs. Preparing these elements encapsulates the business functionalities in blocks which helps the domain experts to identify the services. As a result of applying the systematic approach (i.e., see Section 4.2.1), a set of 33 SSEs has been produced. Table 4.5 presents a sample SSEs that are produced applying the systematic approach. The full list of resulting SSEs is depicted in Table in Appendix B.

Table B.1 presents the resulting 33 SSEs with the activities that belong to each SSE in detail. Note that some roles have been distributed on multiple SSEs. For example, the “Patient” role has been distributed to nine SSEs. Each SSE performs a different activity, for example, “Patient1” interacts with the imaging department, whereas “Patient2” interacts with the admissions department. The SSEs are presented to the domain experts in the next step who would allocate them in groups according to their functionalities.

Table 4.5: Sample SSEs produced in input data preparation phase

SSE ID	SSE Name/Role	Tasks and Activities	Remarks
SSE 1	$Receptionist_5$ [Handle a Patient General Reception]	• Request appointment	Interactions between patient and reception to request an appointment.
SSE 2	$Receptionist_6$ [Handle a Patient General Reception]	• Book an appointment by phone • Visit clinic	Book an appointment by phone.
SSE 3	$Patient_9$	• Pay or come to an agreement • Visit doctor • Book appointment by phone • Visit Clinic	Interaction between patient, doctor and click reception.

Figure 4.7 depicts the relationships between the resulting SSEs. The visualisation of the interconnections between the SSEs helps to facilitate understanding the relationships between the resulting SSEs. The row and column headers represent the SSEs' IDs. The rows represent the source SSEs that initiate the connection, while the columns represent the destination SSEs. The intersection between the rows and columns represents the degree of connection between the source and destination elements; the larger the number the stronger the dependency. However, the domain experts at KHCC have a thorough understanding of these relationships based on their experience and knowledge, which is very helpful to identify the right services from the beginning.

Destination Elements (columns) / **Source Elements** (rows)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1		2																																
2	3																																	
3				3	1																												
4			2																														
5			1									1																						
6							2					1		1																				
7																																	1	
8									1																								
9								1																									
10											2																					
11										2																						
12																																	
13													4			1			1		1		1												
14																					1											
15														3							1											
16																																	
17																		1															
18																																	
19																				1													
20																			2															
21																							1											
22																					1											
23																								1									
24																									1								
25																								1									
26																																	
27																																	
28	2																														1	
29																														2			
30																													1				
31																																	
32																																	
33																																	

Figure 4.7: Relationships between SSEs of CCR Case Study

As a result of this experiment, the KHCC domain experts have successfully created a set of service solutions by grouping the relevant SSEs into candidate services. Each domain expert has produced one solution. Table 4.6 presents a candidate solution that is created by one of the domain experts. The columns in this table represent the service number, the set of SSEs that comprise the entitled service, coupling, cohesion, and fitness measures. The implementation of the fitness function metrics will be discussed in detail in Chapter 5, however, in this chapter, we point to these equations. The coupling value is calculated using the formula presented in Equation 4.1, whilst cohesion is calculated using the formula in Equation 4.2, and fitness is then calculated using the formula presented in Equation 4.3. The last row in the table shows the average values for this solution. The coupling = 0.15, cohesion = 0.189, and fitness = 0.5195.

Table 4.6: A Candidate SOA Solution created by KHCC domain expert

Service #	SSE ID	Coupling	Cohesion	Fitness
Service 1	1 2 6 7 16 32	0.25	0.083	0.4165
Service 2	12 13 14 15 17 18 19 20 33	.25	0	.625
Service 3	21 22 23 24 25	.25	.111	.4305
Service 4	3 4 5 26 27 28 29 30 31	.25	.167	.4585
Service 5	8 9 10 11	0	.333	.6665
Average		**0.15**	**0.189**	**0.5195**

The other solutions produced by the participants are presented in Appendix C. Each participant created one candidate solution using the available set of SSEs and using his/her knowledge and experience. Table 4.7 reveals a summary of the fitness values obtained in the resulting solutions. The fitness values obtained from the resulting solutions range from 0.4355 to 0.587.

Table 4.7: Summary of fitness values for manual solutions by KHCC domain experts

Solution	Coupling	Cohesion	Fitness
Participant 1	0.15	0.189	0.5195
Participant 2	0.195	0.116	0.4605
Participant 3	0.181	0.355	0.587
Participant 4	0.258	0.308	0.525
Participant 5	0.208	0.147	0.4695
Participant 6	0.263	0.134	0.4355
Participant 7	0.224	0.207	0.4915
Average	0.211	0.208	0.498

To measure whether the resulting solutions are created by chance, a set of solutions are created using a random allocation method. This technique allocates SSEs into candidate services at random. The number of services is random as well. The fitness values for the resulting solutions are also calculated using the same equations. Table 4.8 shows a summary of fitness values that belong to the resulting random services.

Table 4.8: Fitness measures for baseline solution samples produced randomly

Solution	Coupling	Cohesion	Fitness
Participant 1	0.4363	0.001	0.28235
Participant 2	0.4326	0	0.2837
Participant 3	0.4366	0	0.2817
Participant 4	0.4903	0	0.25485
Participant 5	0.5134	0.0023	0.24445

Participant 6	0.4796	0	0.2602
Participant 7	0.4344	0.0125	0.28905
Average	**0.46**	**0.0023**	**0.271**

The random solutions have obtained high coupling and low cohesion values, which explains the low average fitness value for the entire (Fitness = 0.271). Figure 4.8 shows the curves for the domain experts' solutions (i.e., dashed green curve) and for the random solutions (solid orange line).

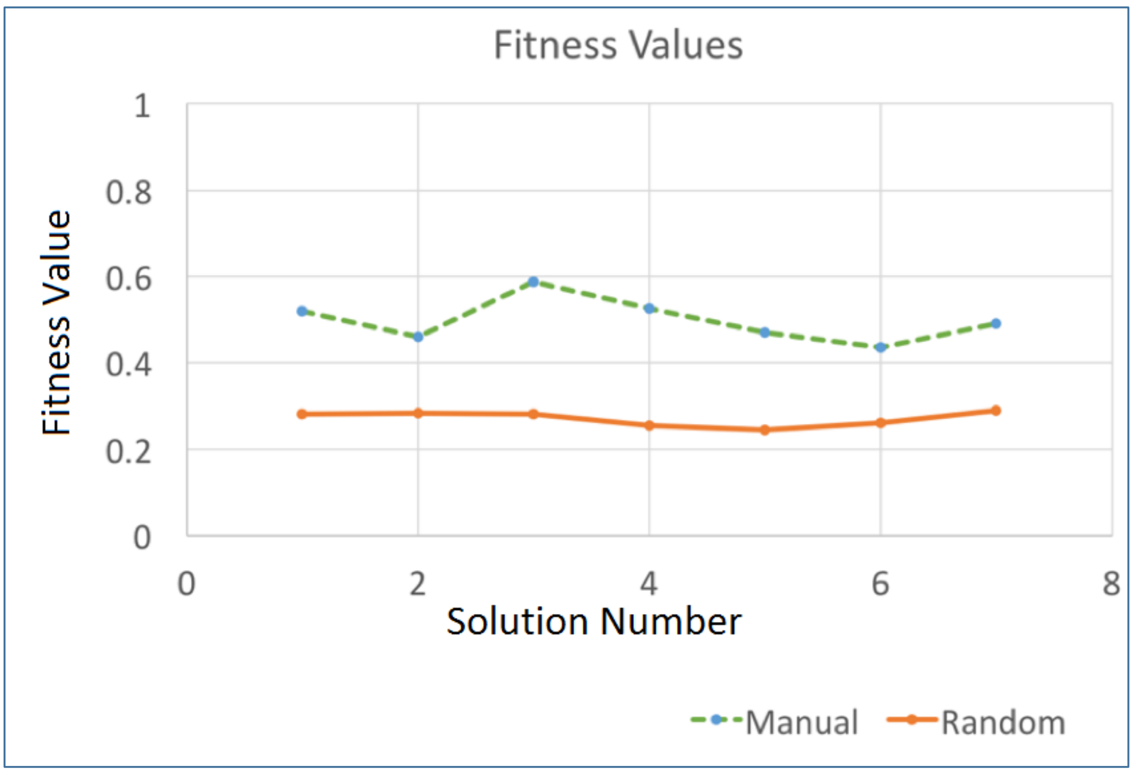

Figure 4.8: Fitness values of manual solutions and random solutions

It will be observed that fitness values for all participants' solutions have higher fitness values in comparison to the random service solutions, which represent an indicator to show that the proposed BPMiSearch framework outperforms the baseline services. However, the statistical analysis test should confirm this observation.

4.3.3. Analysis

With regard to the input preparation layer, the importance of this layer appears when the SSEs are produced. These SSEs protect the basic chronological sequence and encapsulate the key activities required to perform as stand-alone blocks. More importantly, the participants work with only 33 SSEs rather than working with approximately a thousand raw elements to derive candidate services. This advantage reflects the high importance of the input-preparation layer. It is also anticipated that the search will result in high performance.

To compare the domain experts' solutions (i.e., denoted by the BPMiSearch solutions) with the randomly produced sample solutions (i.e., denoted by Baseline solutions), a statistical analysis test would be employed. The statistical analysis test establishes whether the difference between the means is significantly different.

The first step is to compare the means of the two populations produced by two independent methods (i.e., the BPMiSearch and Baseline), and a data distribution test should be performed to confirm that the sample data have been drawn from a normally distributed population. Due to the small sample size, a Shapiro-Wilk test for normality is adopted at an alpha level of 0.05. The null and alternative hypotheses for this test are formulated as follows:

H_0: the population of data samples is normally-distributed.

H_1: the population of data samples is not normally-distributed.

If the p-value is less than or equal to the chosen alpha level (i.e., 0.05), then the null hypothesis is rejected. Failing the normality test indicates that in a 95% confidence level the data are not normally distributed.

Results of applying the Shapiro-Wilk normality test are presented in Table 4.9. For both the BPMiSearch and Baseline data samples, the p-values (denoted by Sig. column), are greater than the alpha level. Moreover, the skewness and kurtosis are within the acceptable limits of ±2 (Trochim and Donnelly, 2006; Field, 2000; Field, 2009; Gravetter and Wallnau, 2014). Thus, we accept the null hypothesis, which implies the data are normally-distributed, and that indicates the possibility of performing a parametric test such as the t-test.

Table 4.9: Shapiro-Wilk Normal Distribution Test

Method	Descriptives				Shapiro-Wilk		
	Mean	**SD**	**Skewness**	**Kurtosis**	**Statistic**	**df**	**Sig.**
$\text{BPMiSearch}_{\text{Fitness}}$	0.498	0.050	0.723	0.416	0.156	7	0.200
$\text{Baseline}_{\text{Fitness}}$	0.271	0.174	-0.579	-1.642	0.868	7	0.177

Since the means of two populations from different methods (i.e., independent methods) are to be compared, a t-test can be used to perform this comparison. Table 4.10 reveals the t-test results. The $\text{Baseline}_{\text{Fitness}}$ samples group (N = 7) was associated with a fitness value $\text{BPMiSearch}_{\text{Fitness}}$ (M = 0.498, SD = 0.051). The $\text{Baseline}_{\text{Fitness}}$ samples group (N = 7) is associated with a numerically smaller fitness value $\text{Baseline}_{\text{Fitness}}$ (M= 0.271, SD = 0.017).

Table 4.10: Independent samples t-test results

Method	Group Statistics			Independent Samples Test				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	Effect-size r
$\text{BPMiSearch}_{\text{Fitness}}$	0.498	0.050	0.019	11.276	12	0.000 (< 0.05)	5.972	0.948
$\text{Baseline}_{\text{Fitness}}$	0.271	0.017	0.007					

To test the hypothesis that the $\text{BPMiSearch}_{\text{Fitness}}$ samples and $\text{Baseline}_{\text{Fitness}}$ samples are associated with statistically significantly different mean fitness values, an independent samples t-test is performed. The independent samples t-test is associated with a statistically significant effect, $t(12) = 11.276$, $p < 0.05$. Thus. The $\text{BPMiSearch}_{\text{Fitness}}$ samples are associated with a statistically significantly larger mean fitness value than the Baseline$_{Fitness}$ samples. Cohen's d is estimated at 6.0234, with an effect-size at 0.948, which is a large effect based on Cohen's (1992) guidelines.

Results reveal that the BPMiSearch framework has successfully derived software services from the BPMN models of the CCR case study, and the resulting service solutions are feasible and are of a significantly higher quality (i.e., in terms of effectiveness) compared to services produced randomly. The feasibility of the solutions stems from the fulfilment of the SOA service constraints such that each service does not lack for SSEs, and each SSE is allocated in one service only. These constraints establish the construction of services that conform to the SOA principles (e.g., loose coupling and abstracting the underlying business).

This indicates the capability of the developed framework for deriving candidate services from the BPMN. The implicit experience and knowledge of the domain experts at KHCC have played a significant role in deriving candidate services for Cancer Care processes they have studied. In addition, the domain experts have validated the proposed service identification framework by conducting a successful service derivation. Thus, it is anticipated that using search-based or interactive search-based methods to automate the service identification process will help to find solutions with higher fitness values.

4.4. DSRM Evaluation Phase

The evaluation process comprises two activities: the validation (i.e., the right product is to be built) and the verification (i.e., ensure the product is built to the right specifications). DSRM supports creating artifacts that address and resolve real-life problems, and the evaluation is an important part of that (Hevner et al., 2004, Simon 1996). The evaluation phase of the service identification framework is guided by two approaches that are integrated with the DSRM Holistic Evaluation Method (Prat et al, 2014). This section reports on how to use the perspectives of DSRM to evaluate the proposed SIM. DSRM helps to evaluate the artifacts along different dimensions such as the structure (i.e., verify that the structure is well-designed and free of errors) and activity (i.e., validate the behaviour of the framework components).

Among the associated evaluation criteria some perspectives were selected as they fit the assessment of the proposed framework. From a structural dimension, the correctness (i.e., includes consistency and completeness) criteria were selected (Part, et al., 2014). From the activity dimension, the performance and efficiency criteria were selected (Harman et al., 2007; Salomon, 1996).

Table 4.11 summarises the evaluation phase of this iteration; the columns represent the evaluation activity, and the rows represent the method to be adopted. Note that the verification activities focus on inspecting the correctness of the framework structure, whereas, the validation activities focus on the behaviour of the search-based service identification method (SIM).

Table 4.11: Search-based SIM Evaluation Activities

Evaluation Activity	
Validation	**Verification**
▪ Examine the conformance to SOA principles (i.e., coupling and cohesion) ▪ Benchmarking: compare to BPAOntoSOA Framework	▪ Walkthrough (inspection) method to evaluate the correctness of the outcomes and compare them to the inputs

4.4.1. Validation

In order to validate the BPMiSearch Framework, two methods have been adopted; (i) show that the resulting candidate services adhere to the selected SOA principles, and (ii) benchmarking to contrast and compare BPMiSearch to BPAOntoSOA from different aspects.

4.4.1.1. Conformance to SOA Principles

Showing that resulting services conform to SOA principles is important in presenting a high-quality candidate services' product and also to show that the implementation of the resulting solutions is feasible. The main SOA principles being addressed in the research are loose coupling and high cohesion (Allweyer, 2016; Papazoglou, 2003). The set of constraints we have implemented in our approach ensure confidence about the existence of these principles. The anticipated SOA principle we are targeting are presented as follows:

1) Loose Coupling: This principle confirms that the resulting candidate services can be independent. The higher the dependency between software services, the tighter the coupling. Minimising the dependency between candidate services is important as this allows for the production of stand-alone services. Ostensibly, in terms of loose coupling, two service types could be produced: stand-alone services and services with low dependability (Yousef et al., 2011). Using a design metric to measure the coupling factor is the first step towards satisfying the principle of loose coupling. Producing stand-alone candidate services is the ideal situation, but it is not possible all the time. Nevertheless, some factors encourage this principle: (i) the SSEs are generated based on the task rather than the role which gives more flexibility to locate that element in such a way that it results in low coupling (ii) grouping SSEs based on the functions they perform and the interconnections between them helps to avoid calling other external services as all the required functionalities are grouped in the same service (iii) the coupling design metric that assigns the best fitness value to stand-alone services (i.e., or services with low dependability) suggests to the user (i.e., or the search algorithm) that the desired elements should be allocated together as this will produce higher quality services. As presented in the previous section (4.3) some of the services produced by domain experts obtained the best coupling value (i.e., zero). The proposed service identification framework obviously supports the production of these stand-alone services.
2) High Cohesion (Abstracting the Underlying Business): The term, Cohesion of services, means the encapsulation of the relevant elements such that they collaborate to perform one or more related functionalities (Shim et al., 2008). This minimises the need to contact other services to perform extra functionality. This looks similar to the coupling principle, and therefore, the same factors encourage

the search-based service identification framework to support the cohesion principle. This service identification framework supports the derivation of the candidate service with high cohesion by (i) preparing the search elements at the right level of granularity, which provides the flexibility to produce services that focus on performing specific tasks (ii) and formulating the cohesion design metric to assign high cohesion values to services that have a clear purpose e.g., those that perform only one functionality.

No doubt the consistency between the coupling and cohesion adds another advantage as they move towards the same goals. Thus, distributing the SSEs on services based on the interconnection between them supports the principle of cohesion, as well as endorsing the principle of coupling (Papazoglou, 2006). For example, using the matrix presented in Figure 4.7, if elements 1 and 2 are allocated in one service, this will result in a service with a coupling of zero, and cohesion of 1. The fitness value for this service is 1, which is an ideal situation according to the fitness function.

4.4.1.2. Benchmarking

Benchmarking is an important technique to validate the proposed SIM (Dolan and Moré, 2002). Conducting a comparison with other successful methods is an approach to display the improvements that have been achieved by developing our framework. In this research, the proposed service identification framework is compared to the BPAOntoSOA (Yousef et al., 2010). BPAOntoSOA is an ontology-driven framework for service identification that derives candidate SOA services from business process architectures (BPA). This framework has developed a static clustering algorithm (i.e., RPA-Clustering algorithm) to parse the entities in the Riva-based BPA and group the relevant elements into candidate services based on the relationships between them. The outcome of this framework is a list of candidate services in which the main functionalities of each service are presented. BPAOntoSOA automates some activities of the service identification (i.e., the clustering algorithm). However, the data preparation should be accomplished by a software engineer. In addition, the layer of output data refinement is missing, thus the resulting services are prepared as a list of functionalities rather than services that follow the standard specifications of web services, e.g., Web Service Description Language (WSDL) file. The reasons for selecting this technique to perform benchmarking have been discussed in Section 3.3.1.3, to recap, the key reasons are: (i) both BPAOntoSOA and BPMiSearch are top-down approaches that adopt business process as input (ii) both frameworks use the same case study (i.e., CCR case study) for evaluation, which helps to compare the outcomes of the two frameworks, and (iii) all the implementation specifications of BPAOntoSOA are available in full detail, which supports accurate comparisons in different aspects.

It is important to emphasise that benchmarking evaluation with BPAOntoSOA will be driven by the DSRM iterations, such that in each iteration, the newly developed parts of our framework (i.e., BPMiSearch) enable comparing different aspects. For example, in the first iteration, the abstraction level of inputs for the two methods will be compared, whereas, in the second iteration the search algorithm enables to compare the automation aspect, while in the third iteration, developing the interactive framework enables to compare the two frameworks in terms of quality assessment. Distributing the benchmarking on DSRM iterations is useful to highlight and appraise the additional components that are added to the BPMiSearch framework.

Table 4.12 shows a comparison between the two SIMs in terms of the abstraction level of inputs and the conformance to SOA principles. However, in the following chapters, other aspects of service identification will be discussed and compared.

Table 4.12: Comparison between BPAOntoSOA and the proposed service identification framework

Comparison	BPAOntoSOA	Search-Based
Abstraction level of inputs	Adopts the Riva-based business process architecture (BPA) diagrams as input.	Adopts BPMN 2.0 as input.

Conformance to SOA principles	Considers SOA principles of loose coupling, reusability, composability, and stateless.	Considers the high cohesion and low coupling principles.

With regard to the abstraction level of the inputs, ostensibly BPAOntoSOA adopts a Riva-based BPA which is at a higher abstraction level compared to the BPMN that is adopted by the search-based framework. However, at this level of abstraction, the Riva-based BPA diagram can display the entire view of the enterprise activated in a wider view compared to the BPMN. Reflecting on the abstraction level of inputs, the key drawback appears when deriving the operations that construct the candidate services; in BPAOntoSOA this operation requires an extra phase (i.e., analysing each control process and unit of work to find the workflow activities). Conversely, using the role-based BPM (i.e., BPMN) provides all the required activities and elements to identify the functionalities of each service in detail. Therefore, the search-based framework outperforms BPAOntoSOA from this perspective.

Conformance to SOA principles is a crucial aspect to be provided by both SIMs. BPAOntoSOA is based on a static clustering algorithm that identifies RPA clusters (i.e., candidate services). The RPA algorithm uses Riva-based business process architecture (BPA) as input, after preparing the second cut of Riva diagrams, all relationships between units of work (UoW) are identified. The first cluster (i.e., candidate service) is instantiated and the first UoW is allocated to that cluster. The algorithm traverses the relationships between the UoWs and allocates them together in a candidate service if they are directly connected. The UoWs have a high abstract level in the diagram such that each UoW comprises a large set of functionalities. In contrast, the search framework creates the search space elements (SSEs) at the beginning with a finer grained level of granularity, and then allocates these SSEs in candidate services based on the interconnections between them. In BPAOntoSOA, there is a possibility of producing a solution that does not conform to SOA principles (very coarse-grained services that achieve a high coupling and low cohesion). In this case, the human expert should exclude the conditional relationships between services to produce a feasible solution. In contrast to this, the search-based framework checks that the produced solutions are feasible and conform to SOA principles. This check is implemented in the framework as a constraint to ensure the feasibility of resulting solutions. This has a direct effect on the quality of the resulting candidate services as well as the quality of these services. The search-based framework outperforms the BPAOntoSOA in this regard.

4.4.2. Static Verification

Static verification uses techniques such as analysis to check for the correctness of the software parts. One of the static verification techniques is to perform an analysis (Bartley et al., 2002). This method of verification applies when performing calculations using classical textbook methods or generally accepted computer methods that perform analysis and mathematical calculations (Whitner et al., 1989). Sampling and correlating measured data could be used as an analytical method by observing the test results and then comparing these results with the expected values (i.e., calculated manually) to show the consistency (i.e., a status where there are no internal contradictions) between them.

To check for the correctness of the service identification approach, a statistical comparison between the original models and the SOA solutions that are produced manually by domain experts will be conducted. This comparison aims to contrast the number of elements in the original models (i.e., by counting the elements in the graphs manually) with the elements in the candidate solutions. By walking through the entire mapping process, all the business elements can be checked if they are mapped correctly on to the corresponding service components. Successful results verify the search-based framework. Table 4.13 reveals the results of the inspection method.

Table 4.13: Inspecting the Inputs and Outputs of BPMiSearch Framework

Component	Input: BPMN Models	Output: Service Components	Remarks
BPMN Models	CCR case study contains 18 Models that capture all the processes at KHCC.	33 Search Elements (i.e., the main building blocks of candidate services). Each SSE encapsulates a set of activities (i.e., operations), events, and	Consistent

		data elements (i.e., service parameters).	
Start and End Events	41 border events to show the beginning or end of a specific business service.	Border events help to identify the borders of SSEs. In some cases, the Lanes can play this role.	Consistent
Lanes and Processes	18 processes comprise 33 lanes. Distributed across 18 models that group activities and elements based on the role.		
Gateway (Complex, Exclusive or Parallel gateways)	65 gateways that allow the conditional statements to be applied.	Gateways encapsulated inside the SSEs. These gateways connect the services by creating the callings according to the sequence flow.	Consistent
Intermediate Events (Throw and Catch)	65 Intermediate Throw/Catch events were identified to connect activities of one role that are distributed across different models.	Intermediate Throw/Catch events link the BMPN model and make it possible to follow the chronological sequence of the workflow and to connect the models.	Consistent
Manual Task	92 manual tasks that should be performed manually (e.g., perform a lab test). These tasks are connected to other user tasks or send/receive tasks and do not perform stand-alone functionalities.	The 33 SSEs comprise 227 total tasks (i.e., service operations). The number of operators equals the total number of tasks in the original CCR models.	Consistent
User Task	67 user tasks that can be fully automated.		
Send or Receive Task	68 conversation tasks were identified. These tasks initiate conversations between different roles.		
Sequence Flow	410 sequence flow relations were identified to connect activities and create the chronological workflow.	Resulting services protect the sequence flows between SSEs with the same structure found in the original BPMN models.	Consistent

Note that the entire process of mapping the business process elements to the corresponding candidate SOA services are performed with no errors, and both the inputs and outputs are consistent. However, since there is no one-to-one mapping from business processes to software services, elements of the original BPMN models are converted into different forms (e.g., border elements help to construct SSEs in the needed level of granularity). Tracing these elements can be done in one direction (i.e., from business process to services).

The outcomes of this test clearly highlight the completeness of the system structure, to confirm that the framework is built in the right way. The dynamic and static verification techniques presented in this section confirm that the search-based SIM is well-designed and error-free, and all the components are performing as expected.

4.5. Summary

This chapter has introduced the BPMiSearch Framework, which is a comprehensive service identification framework that derives SOA candidate services from role-based BPMs. This multi-layered framework comprises three main layers: the input BPM preparation, service identification, and candidate service refinement. The experiment in this chapter demonstrates a successful attempt to derive candidate services for Cancer Care from the BPMN. The main activity of service identification has been performed using domain experts at KHCC based on their knowledge and experience as a proof of concept to show the validity of the framework. However, the service identification phase activities will be performed using an interactive metaheuristic search.

In addressing RQ1, a role-based BPM has been adopted as an input. More specifically, BPMN is utilised in this research as the modelling language. However, across the three phases of the proposed framework, deriving services is aligned with conformance to SOA principles such as coupling and cohesion.

In addressing RQ1 and RQ2, a set of achievements and benefits has been obtained. Firstly, a generic framework for service identification has been developed. In this multi-layered framework, the input BPM models are traversed and refined at an appropriate granularity level to produce generic building blocks (i.e., SSEs) that construct the search space. Secondly, a suitable representation that fits the service identification problem has been prepared along with the corresponding genetic operators and the fitness measures. The representation method is expected to be efficient as it is designed to produce feasible solutions that satisfy the required constraints and conform to SOA principles (i.e., coupling and cohesion). Moreover, decoding the candidate services to the phenotype form is valid in all possible cases. The decoding mechanism maps each business process element in the resulting candidate services to the corresponding service component based on the WSDL file standards.

To arrive at better SOA solutions that show superior fitness values, a set of correction actions should control the mapping method. Using a search-based method, the correction actions will be implemented within the search process activities such that the search engine obtains solutions of better quality.

Building a rigorous understanding of the search-based SIM requires conducting a set of experiments that examine the efficiency and effective of the proposed method. Exploration and exploitation of the search space are important in order to find a good combination of parameters that support the production of high-quality (i.e., with high fitness values) SOA solutions. The next chapter presents the second DSRM iteration and includes rigorous experiments that automate the search process in order to find SOA solutions.

Chapter 5
The Search-based Framework

5.1. Introduction

Experiments in this chapter investigate a related research aim, i.e. how to effectively explore and exploit the SOA solution search space to arrive at useful and innovative SOA services. Thus, firstly, an initial set of experiments is conducted to trial the performance of the integer representation using a combination of parameters introduced in previous studies. Building on the findings of initial experiments, the second set of experiments is then performed to investigate how empirical trial-and-error parameter tuning might more effectively explore and exploit the search space. Although the stochastic approach does not guarantee the discovery of the best solution, it can be applied to automate the process of finding an acceptable solution among a number of candidate solutions.

Selecting an appropriate technique to address the service identification problem is an important task to satisfy the research objectives. It needs to emphasise that this research is not only about automating the service identification using an SBSE technique, but is also about investigating and highlighting the impact of the interactive preference on the search process when deriving candidate services from role-based business process models. Therefore, it is important to consider the human interaction with the algorithm. As discussed in Chapter 2 (Section 2.2.2), some studies have applied search techniques to derive candidate services from the business process (Khoshnevis and Shams, 2017; Jamshidi et al., 2012; Kazemi et al., 2011). Nevertheless, none of these studies has used interactive search for service identification.

Chapter 2 discussed as well the different evolutionary computing techniques and presented the applications of Single, Multiple, and Many-Objective Genetic Algorithms. Nevertheless, MOGAs have some advantages, but considering the requirements of this research, a single-objective GA that uses a weighted-sum aggregation approach has been adopted to develop the interactive search engine and perform the experiments. Two key reasons are behind this selection. Firstly, the simplicity of the algorithm helps to maintain a focus on the context of bridging the gap between the business and the candidate services. Secondly, to develop a technique with high sensitivity to human interaction. It is important to allow the interactive preference to steer the trajectory of the search which requires the selection of a sensitive algorithm that provides the interactive preference with an influence on the search. In addition, if the fitness function includes a small number of design metrics, it is anticipated that this function would be more sensitive to the interactive preference values. As a consequence, just two design metrics are selected to create the objective fitness function (i.e., coupling and cohesion), and subsequently, the subjective feedback value will be aggregated with the fitness value.

The selection of this weighted-sum aggregation function is anticipated to help understand the influence of the interactive preference on the search process which satisfies the objectives of this research. In addition, the main advantage of this technique is the simplicity and elegance of it (Bandyopadhyay and Saha, 2012; Rai, 2006). A key challenge when using a weighted-sum aggregation technique to calculate fitness is the appropriate selection of the weights (Bandyopadhyay and Saha, 2012). The weighted summation essentially provides a convex combination of the different objectives.

The third research question (RQ3) in this thesis states:

> "**Can the services solution space be effectively and efficiently explored and exploited in order to derive candidate SOA solutions from BPMN 2.0 models?**"

Addressing this research question necessitates the adoption of a robust search-engine that uses the representation technique along with the fitness measures presented in the previous chapter to identify candidate SOA services. Subsequently, the representation method should be examined to find how capable it is with regard to mapping the SSEs to the corresponding services. In addition, a testing method should be designed to find an appropriate collaboration of parameters which could be used to explore and exploit the search space. Another important issue is to examine the fitness function and the designated genetic operators. It is essential to check the three DSRM phases of design and development, demonstration, and evaluation to develop and evaluate the search-based framework before proceeding to the next DSRM iteration (i.e., interactive search). It also paves the way for the use of the interactive human preference in the succeeding stages of this research.

The demonstration phase of this iteration (Section 5.3) presents a detailed evaluation procedure to examine all the parts of the proposed search-based framework. A real-life example from the domain of Cancer Care (Yousef, 2010) is utilised for this purpose. Moreover, benchmarking reveals how powerful the search-based framework is in comparison with other techniques (Sections 5.3 and 5.4).

5.2. The DSRM Design and Development Phase

Building on the findings of the previous DSRM iteration (i.e., presented in Chapter 4), this iteration aims to automate the service identification framework. In addition, the search has the capability to optimise the resulting solutions such that the fittest services are produced. The representation method and fitness measures introduced in the previous chapter are used as the fundamental components of the 'search-engine' tool. Furthermore, the genetic algorithm (i.e., denoted by GA) should also consider managing the constraints that ensure the feasibility of resulting solutions.

5.2.1. The Genetic Algorithm

The genetic algorithm (GA) is a particular mathematical search and optimisation technique which is inspired by natural genetic laws (Noever and Baskaran, 1992). Since its development in 1975 (Holland, 1075), it has been applied to optimisation problems in different fields such as engineering, machine learning and physics (Davis, 1991; Goldberg, 1989). Applying the method to partial and un-optimised solutions generates mutated or recombined alternative copies of solutions until a good-enough solution emerges. In each generation step, the individuals are mixed using recombination operations (i.e., crossover and mutation) which exchange partial segments of these individual solutions. The recombination plays a significant role in breeding using the chromosomes that support exchanging and copying the traits of individuals to the next generation. The mathematical mechanism of these operations is based on promoting the privileged solutions at the expense of poor solutions, while at the same time maintaining a robust pool of new alternatives.

What distinguishes GAs from other biological models is the adoption of fixed population size (Smith and Vavak, 1999). The selection mechanism in the GA is thus split into two phases, the parental selection and the replacement strategy. The replacement strategy for the Generational GAs (GGAs) deletes all the members of the previous population. The reproduction operators of the GGAs can preserve and propagate the good solutions. However, this is an optional choice. On the other hand, with Steady State GAs (SSGAs), the members of the previous population are not entirely deleted, instead, few population individuals are replaced (De Jong and Sarma, 1992). Different replacement strategies are applied by SSGAs such as Replace-Worst and Replace-Random (Smith and Vavak, 1999). The two strategies have different reproductive behaviours that affect the performance of the GA, thus both mechanisms will be examined by the GA to choose a suitable replacement strategy.

The GA can be implemented in different ways, for example, one way was the mechanism introduced by Goldberg (1989). This design supports activating all the genetic operators (i.e., selection, mutation, and crossover) that are presented in the previous chapter. Figure 5.1 shows an activity diagram of the search-based GA adopted by this research and inspired by Goldberg (1989). To recap, integer-based representation is adopted, in which the SSEs have static positions, and an integer number that represents the service ID is assigned to each SSE. The genetic operators are applied to the integer service IDs such that changing the integer value that is assigned to a certain SSE indicates that the SSE belongs to a new candidate service that has the corresponding ID.

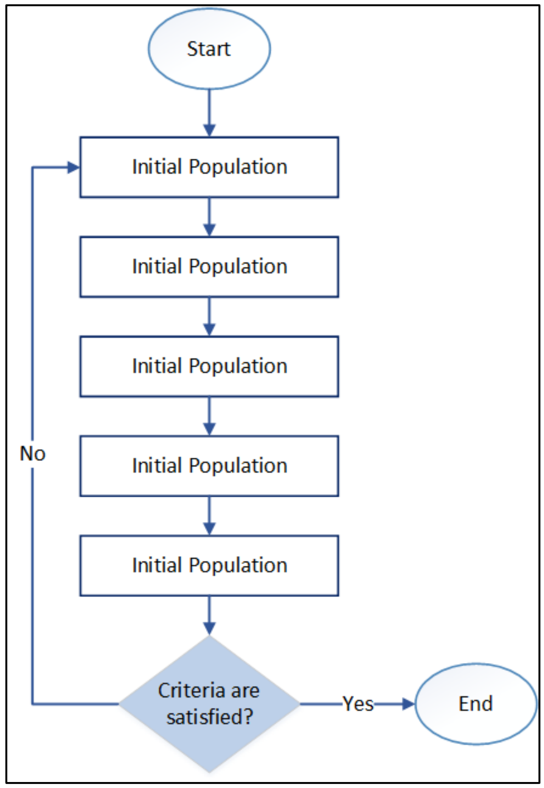

Figure 5.1: GA Activity Diagram adapted from (Goldberg and Holland, 1988), used with publisher's permission.

With regard to the selection, it can be achieved using different techniques such as fitness proportionate selection, roulette-wheel selection, random selection, steady-state selection, and tournament selection (Goldberg and Deb, 1991). The experiments in this research perform the selection by using a tournament selection. This mechanism selects a set of individual solutions randomly and then selects the fittest among them. Although the tournament selection could be slower than other techniques (i.e., as it first selects solutions at random, and then measures the fitness of each solution to find the fittest), it is adopted because it provides a uniform chance to all individuals to be parents.

With reference to the recombination (i.e., crossover), it is achieved using a single pivot point crossover, in which the SSEs of two parents are swapped around the pivot point to produce two offspring individuals.

The first resulting child takes the service IDs from the first parent (i.e., elements before the crossover point) and takes the other IDs from the second parent (i.e., after the point). The second child takes the remainder of these IDs. The key advantage of this crossover operation is the efficiency and flexibility to perform the swapping operation as it works with integer values. In addition, it fulfils the constraints which guarantee that swapping SSEs to another service would not leave the candidate service lacking in elements, or even duplicate an element in more than one candidate service. The probability of applying the crossover operation takes place on the population level, such that if the probability applies to a selected individual the operation is then performed.

The other reproduction operation is the mutation. Different mechanisms are valid when performing this operation such as swapping elements between different individuals, or by flipping the targeted gene value with the complement value (e.g., for integer values from 0-10 the complement is 10 – the value). In the experiments in this research, the mutation is achieved by swapping two SSEs between two services within a single SOA individual (i.e., one generation represents a SOA solution that includes all the candidate services). A mutation probability on the level of population individual is examined before performing this operation.

The genetic operators are applied to the population individuals according to the recombination and mutation probabilities. The breeding pool probability is held at the population level, i.e. population individuals are selected at random according to this probability of recombination and mutation. The implementation details of the genetic operators are presented as follows:

1) Selection: Typically, not all individuals from the current generation would have the chance of surviving before the next offspring (Eiben and Smith, 2003). There is usually a competition between individuals to find a place in the new generation; otherwise, they would die out. Choosing parents for sexual reproduction is applied through selection for individuals from a mating pool (Kazemi et al., 2011).

Selection can be achieved using different techniques such as fitness proportionate selection or roulette-wheel selection, rank selection, steady-state, and tournament selection (Seng et al., 2006). This research adopts the tournament selection mechanism in which a number of individuals are chosen at random from the population for later breeding. The fittest individual among the selected individuals is chosen as a parent. This process is repeated until all the individuals of the new generation are produced. This technique is adopted because it has several benefits over other alternative selection methods for a GA. These benefits include: (i) coding efficiency (Blickle and Thiele, 1996) (ii) the capability to handle either maximisation or minimisation problems without performing any structural changes (Abd Rahman et al., 2016) (iii) the ability to create more diverse populations by providing a uniform probability for all population individuals to be in the new generation (Back and Fogel, 2000; Eiben and Smith, 2003).

2) Crossover: A single point crossover operator is applied to represent the recombination. This function takes two parents and generates two offspring from them. This operator exchanges the allocation of SSEs. It is based on switching the mapping of two individuals around a pivot point that is selected at random (Malhotra et al., 2011). Figure 5.2 presents the single point crossover operator. Note that the position for the crossover point is selected randomly. The first child inherits the ID numbers from the first position to the crossover point in the first parent, i.e., p = {0, 1, 2, 3}. It inherits the reset of IDs from the second parent where p={4, 5, 6, 7}. The second child inherits the remaining service IDs. From two parents, the crossover produces two children.

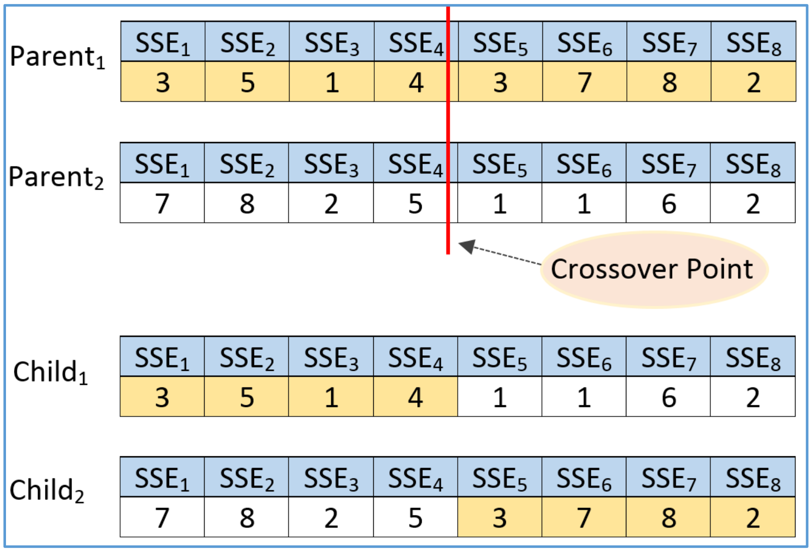

Figure 5.2: Single Point Crossover

3) Mutation: The mutation operator is based on switching the mapping of two SSEs. Two positions are selected at random, and the service IDs at these positions are swapped (Guo et al., 2010). Figure 5.3 illustrates an example of applying the mutation operator. Note that the two service IDs at positions 1 and 5 were swapped.

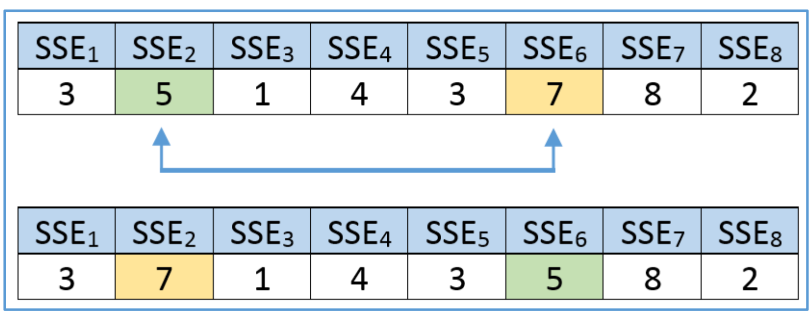

Figure 5.3: Mutation

The genetic operators keep the positions of SSEs fixed, as the operators are applied to the service IDs only such that a new allocation for the corresponding SSEs would be formulated. These operators ensure that no redundant SSEs will be allocated in one service, in addition to no candidate service lacking any SSEs.

With reference to the fitness measures, a single-objective weighted sum fitness function is used. The quantitative measures adopted by this method are coupling and cohesion. Values from the two metrics are accumulated according to the desired weight. Equal weights are to be assigned to the coupling and cohesion through the experiments. The goal is to maximise the fitness value to acquire more effective solutions and minimise the time to arrive at the needed solutions. Equation 4.1 shows the equation of coupling.

With regard to the second component of the genetic algorithm (i.e., the fitness function), the coupling and cohesion design metrics in addition to the weighted sum fitness function are presented in Section 4.2.2, in which the mathematical basis of these metrics is explained. Equations 4.1, 4.2, and 4.3 have been implemented in this chapter to evaluate the quality of solutions and to enable evolving the population and produce more effective solutions (i.e., that achieve high fitness values).

5.3. The DSRM Demonstration Phase

In the course of developing efficient and effective evolutionary computing (EC) applications, it is often necessary to tune parameters by empirical investigation. However, it has been observed that each parameter to be tuned has a large scope of possible values that results in a large number of feasible collaborations which cause an explosion of parameter value combinations at the end (De Jong, 2006). Over many years of using evolutionary computing, it has been found that the performance of the evolutionary algorithm depends to a great extent on the problem case itself (Eiben and Smith, 2003). In this research, empirical trial-and-error investigations are used to tune the genetic parameters. To narrow down the scope of parameters to be tuned, an initial set of parameters that has been used in the literature of EC has been used to demonstrate the search framework. The goal is to find a starting point to begin the tuning experiments while narrowing down the scope of parameters that should be tuned. For the tuning investigation, a set of alternative values is prepared so as to be tested for each parameter.

The first experiment has two key objectives; (i) examine the correctness of the search-engine structure, as well as to verify that the search-engine has no syntax or runtime errors, and (ii) investigate and find the most effective parameters that can effectively and efficiently explore and exploit the search space which indicates that the search is reliable and not random.

The objective of the second experiment is to investigate whether there is a significant effect in using computational support for service identification using a metaheuristic search-based technique. In addition, this experiment aims to show that the metaheuristic search has the power to produce feasible solutions of a high-quality. The second experiment is very important when examining the effectiveness of the search-based method. To fulfil this objective, the search method will be compared to the baseline and manual methods.

5.3.1. Experiment: Find an initial set of parameters for trial-and-error tuning

Initial parameter values have been derived from the literature of evolutionary computing or used by previous research studies (Kim et al., 2015; Li et al., 2015; Goldberg, 1989; Back, 1996; De Jong, 2006). The set of genetic parameters started with in the first experiment and the alternative values for the different parameters are thus as follows:

i. Selection: tournament selection, selection size: 2, 3, 5, 7;

ii. Crossover and mutation probabilities: initially use the following set of crossover and mutation pairs S= {(0.7, 0.1), (0.6, 0.3), (0.0, 1.0), (1.0, 0.0), (1.0, 1.0)}; Note that the first number in each pair represents the crossover probability and the second one represents the mutation. Each experiment is labelled using this naming convention, i.e., GA (crossover, mutation). Next, select the pair that produces the best results for the tuning experiments based on the results of the initial trial.

iii. Number of generations: 250, 500, 1000, and 1500

iv. Breeding pool size: 25%, 50%, 75%, and 100% (i.e. 100 parents generate 100 offspring, and only those offspring become parents of the next generation).

v. Population size: 50, 100, 200, and 300 individuals in the generation.

With respect to the termination condition, the search is not halted until the completion of the execution of the full number of generations. In other words, the termination condition is deactivated. Recording the resulting fitness values of a large number of generations until the performance of the search reaches a fitness plateau supports designing a better strategy for tuning parameters using the empirical trial-and-error technique in the next experiment.

Table 5.1 presents the dependent and independent set of parameters. The dependent parameters (i.e., effectiveness and efficiency) are observed when the set of independent parameters are changed.

Table 5.1: Parameters of the Search Experiments

Experiment	**Parameter Type**	**Values**
Effective and efficient search experiment	Independent	Crossover probability Mutation probability Population size Evolve times Breeding pool size Tournament selection size
	Dependent	Effectiveness (i.e., in terms of Coupling, cohesion, and fitness value) Efficiency (i.e., in terms of time, and number of executions)

To examine reliability, each search is run 50 times to provide average population fitness curves together with standard deviation. The search engine tool is implemented in Java, and the experiments are conducted on a standard desktop PC running the Microsoft Windows operating system.

5.3.2. Results

Figure 5.4 shows the population average fitness curves obtained for the five parameter combinations. Clearly, the combination GA(0, 1) has achieved the lowest fitness values with no substantial improvements, whereas, GA(0.7,0.1) has obtained the highest fitness values among the different combinations. With regard to the rest of the trials, they arrived at fitness values around 0.5 which are intermediate values, and the three sets have obtained slightly similar fitness values. To arrive at the fitness plateau of the best combination, 260 generations were processed.

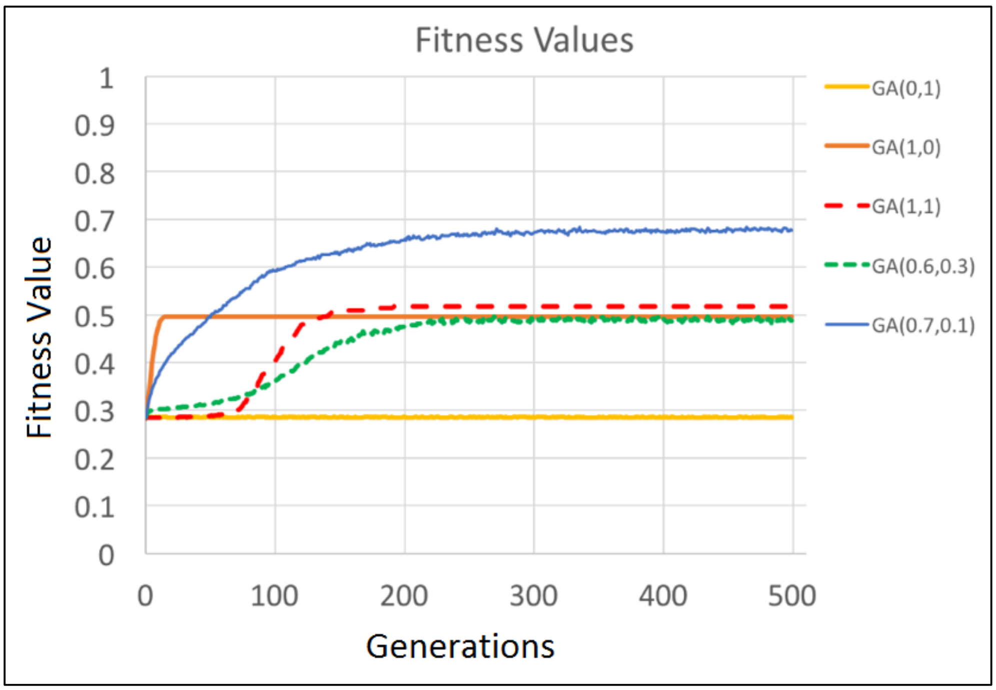

Figure 5.4: Population Average Fitness Curves

Table 5.2 summarises the resulting population average fitness values for the five combinations of parameters when arriving at a fitness plateau. In addition, the standard deviation is recorded as well. The best fitness value is highlighted. Nevertheless, GA(0.6, 0.3) and GA(1,1) arrived earlier at a fitness plateau compared to GA(0.7, 0.1), but the latter appears to reach higher fitness values at the end.

Table 5.2: Population Average Fitness Results at the Fitness Plateau

Experiment	Coupling	Cohesion	Fitness	Std. Deviation	Generations to plateau
GA (1, 0)	0.29	0.2832	284	0.0001	18
GA (0, 1)	0.4466	0.0147	.4966	.0178	1
GA (1, 1)	0.0	0.0341	.517	0.0	192
GA (0.7, 0.1)	0.1649	0.5097	**0.6724**	0.1458	260
GA (0.6, 0.3)	0.1798	0.1647	0.4924	0.1729	308

Figure 5.5 highlights the population fitness curves for the best combination arrived at, i.e., GA(0.7, 0.1). The figure presents the plots of coupling, cohesion, and fitness values. It is observed that there is an obvious improvement of fitness values (i.e., that aggregates the coupling and cohesion) as they started from approximately 0.28 and finished at 0.67.

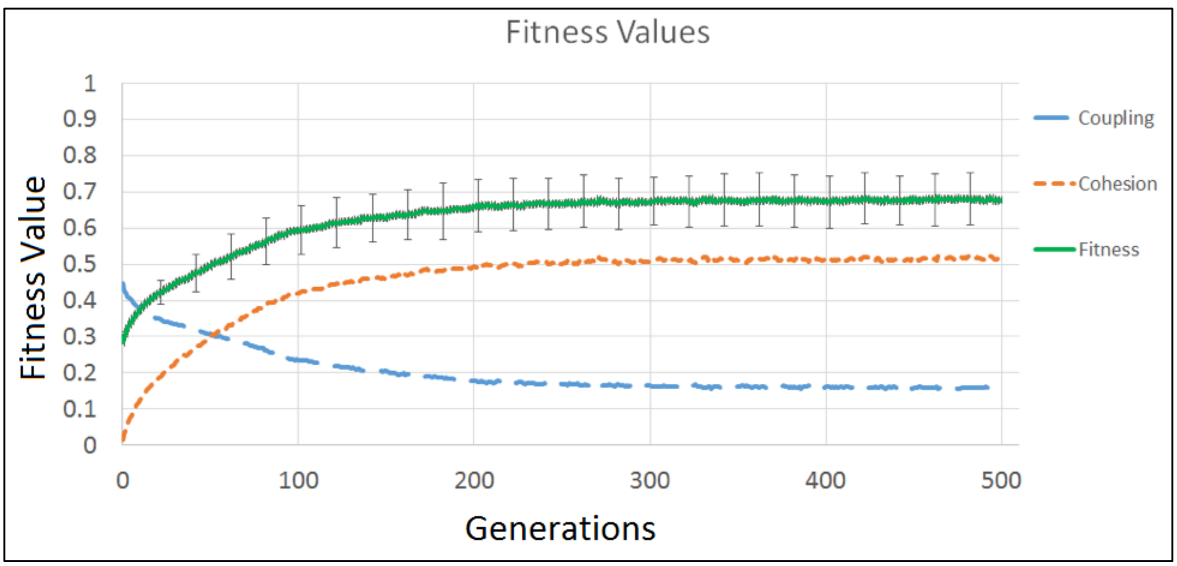

Figure 5.5: Population average fitness and standard deviation, coupling and cohesion measures for GA(0.7, 0.1)

Taking the values from this point and proceeding with more trials while tuning the parameters, the best combination that supports producing solutions with high fitness values is presented in Table 5.3. These parameters create a good balance between exploration and exploitation such that effective solutions can be produced.

Table 5.3: Parameter combination obtained using trial-and-error experiments

Parameter	Value
Population size	100
Number of generations	500
Selection size	7
Replacement Strategy	Generational (100 parents produce 100 offspring)
Crossover probability	0.80
Mutation probability	0.03

The population size is one of the most significant parameters having an influence on GA performance (Alajmi and Wright, 2014). Different population sizes were tested, however, the best size that should be used with this case study is 100 individuals in the population. Smaller population sizes have fewer alternatives. In contrast, compared to larger sizes, there are no improvements in effectiveness.

With regard to the number of generations, the GA would arrive at a fitness plateau in less than 500 generations. Using a larger number of generations has no influence on the effectiveness, but does have a negative effect on efficiency.

With reference to the selection size, although the small selection sizes (e.g., 2 and 3) are time efficient (i.e., consume less time), the experiments show that larger sizes (e.g., 5 and 7) have produced more effective solutions that obtain higher fitness values. However, the population size has a direct effect on the selection size (i.e., a larger population needs to select more individuals). In addition, the scale of the case study plays an important role in choosing the most appropriate selection size. From a practical perspective, with a population of 100 individuals, using the selection size of 7% helps the algorithm to arrive at a fitness plateau more quickly.

Although, some research studies show that a steady-state replacement strategy is more effective than the generational strategy (Vavak and Fogarty, 1996) experiments using the CCR case study with empirical trial-and-error parameter tuning show no significant difference when switching from one strategy to the other in terms of effectiveness and efficiency. However, because it is simpler to implement and trace, a generational replacement strategy has been adopted.

With regard to the crossover and mutation probabilities, the best probabilities found are 0.8 and 0.03 respectively. This combination of parameters has achieved both the highest effective and efficient results.

All results have been obtained using the CCR example problem domain. Figure 5.6 shows population average fitness curves achieved using the best combination of parameters found after the experiment. At the fitness plateau, the average fitness reached a value of 0.785, with a coupling of 0.137 and cohesion of 0.708.

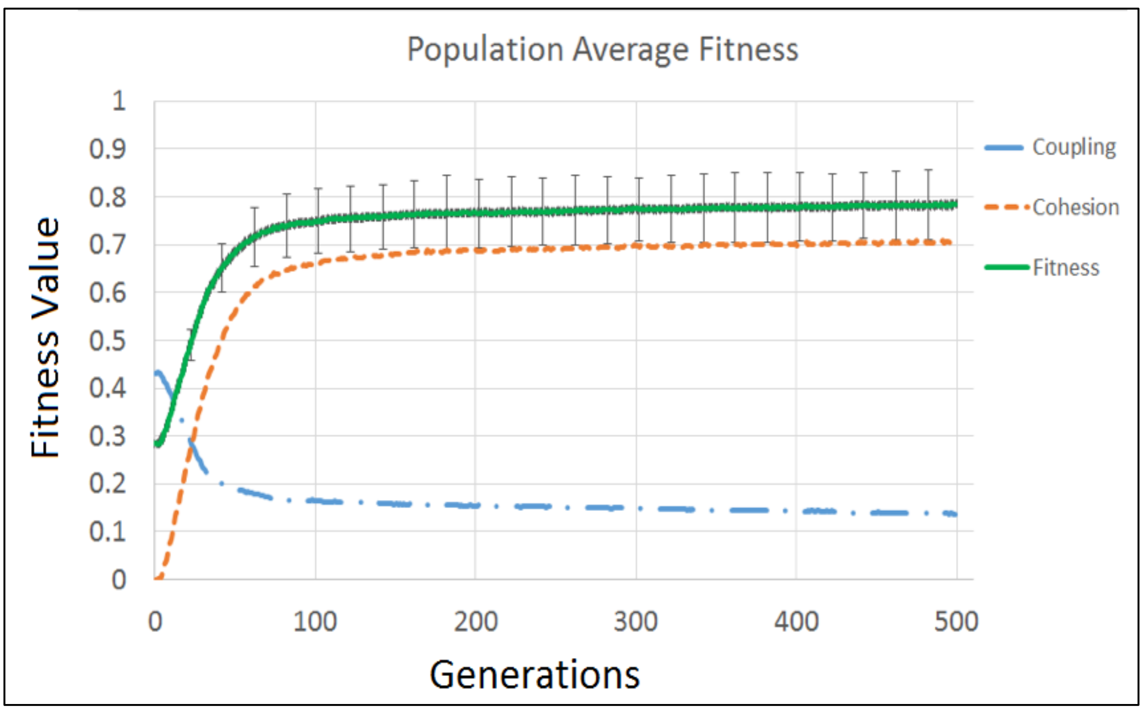

Figure 5.6: Population average fitness and standard deviation for GA (0.8,0.03)

Reflecting on bridging the gap (which is the subject of this research), the obtained results imply that a set of candidate services has been constructed, in which the candidate services are 13.7% are loosely coupled (i.e., close to be stand-alone services), and 78.5% have a clear purpose (i.e., high cohesion). In addition, results indicate that the metaheuristic search can construct candidate services based on the business activities which in result can satisfy the business-IT alignment and reduce the gap between the business models and services. Table 5.4 shows a full solution produced using the GA (0.8, 0.03). This solution includes nine candidate services with a fitness value of 0.87 (i.e., Coupling = 0.099, and Cohesion = 0.839). The first column shows the service number, the second the abstract element name (i.e., as derived from the BPMN models), and the third the detailed activities and functions that comprise each SSE.

Table 5.4: Sample candidate services produced using the search algorithm

Service Number	Abstract Functions	Detailed Functions
Service 1	Patient	(1) Pay (2) Visit imaging department
	Imaging department	(3) Inform patient to visit doctor (4) Inform patient to visit doctor (5) Add and report results (6) Perform test (7) Receive payment (8) Receive patient visiting imaging department (9) Check if patient is medically insured (10) Check if patient has appointment (11) Book appointment for patient
Service 2	Patient	(1) Pay (2) Visit radiotherapy (3) Receive treatment
	Radiotherapy department	(4) Receive payment (5) Transfer patient (6) Ask patient to visit specialist (7) Transfer patient (8) Receive patient visiting radio (9) Check if the patient is medically insured (10) Check if patient has appointment (11) Begin treatment (12) Add results (13) Check if patient needs imaging test (14) Check if patient needs lab tests
Service 3	Patient	(1) Visit department (2) Visit admission department
	Inpatient care specialists and nurses	(3) Begin surgery (4) Check if patient needs surgery (5) Transfer patient to chemo department (6) Check if patient needs chemotherapy (7) Check if patient needs imaging tests (8) Transfer patient to lab (9) Check if patient needs tests (10) Continue treatment (11) Receive patient visiting department and his papers (12) Check if patient needs other treatment (13) Open admission file (14) Transfer patient to radiotherapy department

Table 5.4 (Cont.): Sample candidate services produced using the search algorithm

		(15) Check if patient needs radiotherapy (16) Transfer patient to imaging department (17) Update patient file (18) Add notes to file
Service 4	Receptionist	(1) Request patient's file (2) Check files (3) Return patient's file (4) Find patient's appointments
	Medical Records	(5) Register file's details (6) Find patient's file (7) Save patient's file in library (8) Send patient's file (9) Open file (10) Check files (11) Check if there is a new patient
Service 5	Patient	(1) Handle payment (2) Visit lab
	Lab	(3) Receive patient visiting lab (4) Inform patient to visit doctor (5) Perform test (6) Receive payment (7) Check if patient medically insured (8) Add results
Service 6	Inpatient care specialist and nurses	(1) Follow-up patient (resident doctor) (2) Make appointment in outpatient clinic with patient (3) Check if patient needs to remain in hospital (4) Perform treatment according to patient state (5) Check patient's financial state (6) Transfer patient to imaging department (7) Follow up patient state (resident doctor) (8) Check if patient needs imaging investigation (9) Send sample to lab (10) Check if patient needs lab test (11) (specialist) Review resident doctor's orders, diagnoses patients and review old tests
	Accounts clerk	(12) Approve patient's financial state (13) Check patient's financial state
Service 7	Patient	(1) Pay or come to an agreement (2) Visit doctor (3) Book appointment by phone (4) Visit Clinic
	Receptionist	(5) Inform patient to visit doctor (6) Check if patient medically insured (7) Register Patient details (8) Book appointment (9) Check if the patient is in DB
Service 8	Handle a Patient General Reception	(1) Request appointment (2) Book an appointment by phone (3) Visit clinic
	Handle a Patient General Reception	(4) Transfer patient to emergency (5) Inform patient to visit cancer detection unit (6) Check if patient diagnosed (7) Inform patient to visit specialist (8) Check if patient has appointment (9) Check if patient in DB (10) Register patient's details

Table 5.4 (Cont.): Sample candidate services produced using the search algorithm

		(11) Make appointment (12) Check if emergency
Service 9	Receptionist (outpatient department)	(1) Send the list to registrar (2) Make list of patients who have not attended their appointments
	Medical records clerk	(3) Send reports to managers (4) Generate main statistical report (5) Analyse collected data (6) Collect data from different departments
	Admission clerk	(7) Check if patient is emergency case (8) Inform the patient to visit department (9) Add patient to waiting list (10) Check room availability
	Receptionist (Inpatient care)	(11) Collect patient's files who have been discharged (12) Add collected data to database
		(13) Collect data (14) Send patient's file
	Receptionist (department-specific)	(15) Add collected data to database (16) Collect data (17) Send reports
	Receptionist (Outpatient clinic)	(18) Send patients' files (19) Add results into database (20) Collect data (21) Send list of patients (22) Collect patient files
	Registrar	(23) Add additional information (24) Check for additional information (25) Add primary tumor (26) Extract main details about cancer patient (27) Make copies of pathology reports and death certificates (28) Add required details in JCR form (29) Generate reports about cancer incidents in the hospital (30) Check if primary tumor exists in database (31) Add patient's details to database (32) Check if patient exists in database (33) Check if there is any contradictable data (34) Inform a specialist about contradictable data
	Doctor (Diagnostician)	(35) Check if patient needs admission (36) Receive patient visiting doctor (37) Refer patient to special combined clinic (38) Review lab and imaging results (39) Perform clinical appraisal (40) Check if patient needs imaging investigations (41) Book appointment for patient (42) Take notes and review history (43) Check if patient needs investigations (44) Order test (45) Update patient's file
	Patient	(46) Visit clinic (47) Pay or come to agreement
	Receptionist (outpatient clinic)	(48) Guide patient to combined clinic (49) Check if patient has medical insurance (50) Receive payment (51) Receive patient visiting clinic (52) Check patient's appointment
	Patient	(53) Visit combined clinic

Table 5.4 (Cont.): Sample candidate services produced using the search algorithm

	Combined clinic (specialists)	(54) Check if patient needs tests (55) Order test (56) Check if the patient needs admission (57) Inform patient to visit radio (58) Receive patient visiting combined clinic (59) Book appointment for radiotherapy (60) Inform patient to visit chemo (61) Book appointment imaging department (62) Continue treatment (63) Review patient's history and all investigations (64) Check if patient needs other treatment (65) Check if patient needs chemo
		(66) Request admission (67) Check if patient needs radio (68) Book appointment for chemotherapy (69) Devise plan for treatment
	Patient	(70) Receive information to wait (71) visit specialist
	Specialist	(72) Perform suitable treatment according to patient situation (73) Check if patient needs tests (74) Check if the patient needs admission (75) Perform medical appraisal (76) Book appointment imaging department (77) Request another appointment (78) Send advices and instructions to patient (79) Check if patient needs another appointment (80) Take notes, review history and old tests (81) Update patient file (82) Request admission from admission clerk (83) Order test
	Registrar	(84) Inform patient's specialist to update file (85) Inform patient to visit specialist (86) Check if patient changed hospital (87) Contact patient (88) Update patient's file (89) Find patient's address
	Patient	(90) Pay (91) Visit chemotherapy (92) Receive treatment
	Chemotherapy department	(93) Perform treatment (94) Receive payment (95) Receive patient visiting chemo (96) Ask patient to visit specialist (97) Add results (98) Check if the patient is medically insured (99) Check if patient has appointment

Figure 5.7 shows a visualisation of the connections between the resulting services of the sample candidate solution. As the figure shows, this solution comprises nine services; six of them are stand-alone services (i.e., the green services), and two services have 1-2 connections (i.e., the blue ones). Only one service has high connectivity (i.e., the red service with 5+ connections), and that service contains the core functions

of the Cancer Care, as presented in Table 5.4. Six out of nine candidate services are stand-alone services with specific functionalities (i.e., the green colour), which reflects on the influence of the coupling and cohesion metrics. Two services have low connectivity (blue services) and one core service (red coloured) that comprises the key functionalities in the business process models.

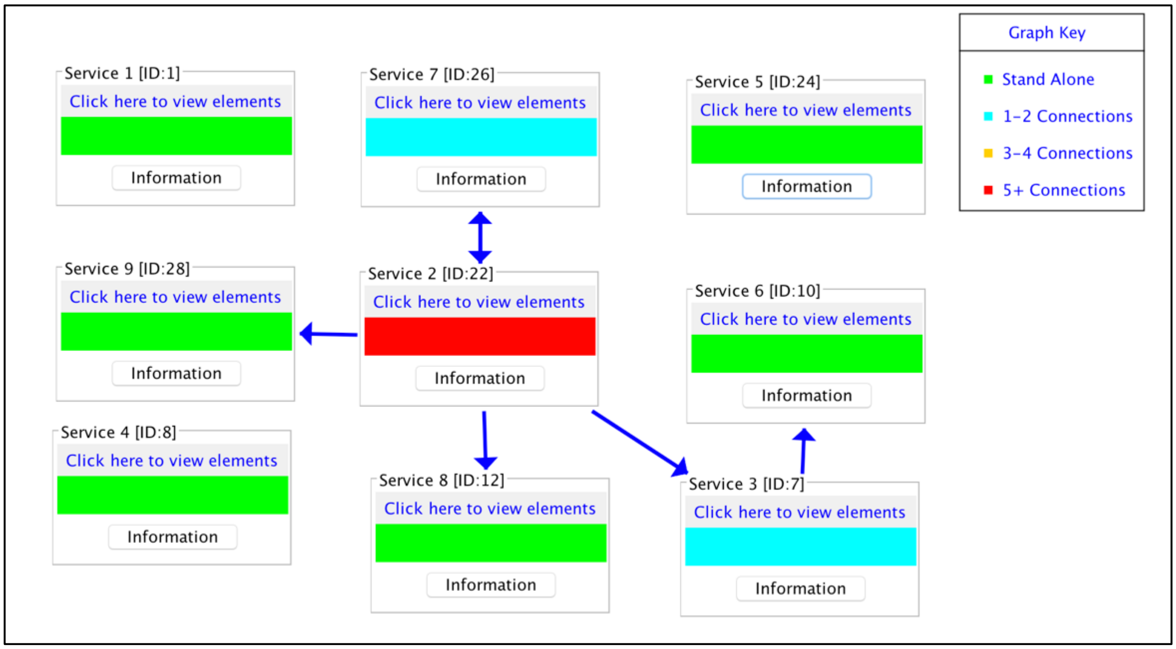

Figure 5.7: Connectivity visualisation diagram for a sample candidate solution

Proceeding to the service refinement layer, the WSDL file of each service is produced. The decoder extracts the detailed functionalities inside the resulting SSEs and maps them to the corresponding service components. This file includes the main components of the standard service interface such as the headers, the operations, and the types. The binding section in the WSDL file allows the service to be accessed by other services. Figure 5.8 shows a part of the WSDL file for a sample service. The unique ID of each element is used to identify the corresponding service component (e.g., SendTask_9). Generating the WSDL files for the resulting solution shows that the list of resulting candidate services has been used to build a SOA solution.

WSDL 2

```
<?xml version="1.0"?>
<wsdl:description xmlns:wsdl="http://www.w3.org/ns/wsdl"
 xmlns:wsoap= "http://www.w3.org/ns/wsdl/soap"
 xmlns:hy="http://www.herongyang.com/Service/"
 targetNamespace="http://www.herongyang.com/Service/">

 <wsdl:documentation>
  KHCC_WSDL_20_SOAP.wsdl
  UWE, SERG Group </wsdl:documentation>  <wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uwe.ac.uk/Service/">
  <xsd:element name="ManualTask_20" type="xsd:string"/>
  <xsd:element name="ManualTask_19" type="xsd:string"/>
  <xsd:element name="ManualTask_21" type="xsd:string"/>
  <xsd:element name="SendTask_9" type="xsd:string"/>
  <xsd:element name="SendTask_9Response" type="xsd:string"/>
  <xsd:element name="ManualTask_23" type="xsd:string"/>
  <xsd:element name="ManualTask_22" type="xsd:string"/>
  <xsd:element name="UserTask_15" type="xsd:string"/>
  <xsd:element name="UserTask_16" type="xsd:string"/>
</xsd:schema>
</wsdl:types>  <wsdl:interface name="service 2Interface" >
  <wsdl:operation name="Process_8"
  <wsdl:operation name="Process_9"
   pattern="http://www.w3.org/ns/wsdl/in-out"
   style="http://www.w3.org/ns/wsdl/style/iri">
   <wsdl:input messageLabel="In"
    element="hy:MyElement" />
   <wsdl:output messageLabel="Out"
    element="hy:service 2Response" />
   </wsdl:operation>
  </wsdl:interface></wsdl:description>
```

Close

Figure 5.8: Sample WSDL file for a candidate service

5.3.3. Analysis

Conducting a positive experiment using different parameter combinations results in producing different solutions, which shows that the search is capable of constructing feasible SOA solutions. However, behaving differently according to the set of parameters used verifies that the genetic operators run as anticipated. Changing the behaviour according to the current set of parameters is necessary to validate the search engine tool. The search-based service identification framework has successfully passed all the phases of service identification with no syntax or runtime errors, which implies that this framework is well-constructed and fulfils the first objective of this experiment (i.e., examine the correctness of the search-engine structure).

With regard to the effectiveness, the resulting solutions can be described using the balance between the exploration and exploitation produced by different parameter combinations. The probabilities of crossover and mutation play a significant role in the formulation of the search outcomes. Results reveal that using a combination with an exploitative nature (i.e., with a high crossover probability) is more effective and obtains high fitness values, whereas, the combination that is more explorative in nature (i.e., with a high mutation value) arrives at a fitness plateau more quickly, e.g., GA (1,1). However, disabling one of the genetic operators (i.e., mutation or crossover) has limited the fitness from reaching high values because the GA is unable to exploit the existing individuals to produce more effective offspring. It is concluded that a combination of a large crossover probability and a small mutation probability has a more powerful effect.

Through observation, it is advocated that a high mutation probability does not result in higher fitness values. For example, GA(0,1) has the highest possible mutation probability but no actual evolution is noticed. Furthermore, although GA(1,1) has a high mutation probability, the search does not produce high fitness values compared to a combination that has a smaller mutation probability, e.g., GA(0.7,0.0.1). The same observation is applied to the crossover probability; a high crossover rate does not necessarily mean high fitness values. Nevertheless, when compared to mutation, a crossover more inherently respects the reliability of the search element allocation in candidate services and so is computationally more straightforward and efficient at maintaining diversity in a population of SOA solutions.

Regarding the characteristics of the search space, it is recommended that all the potential optimum configurations should be included in the search space to be comprehensive (Giridhar and Agrawal, 2010). However, only useful configurations without redundancy should comprise the search landscape (i.e., a smaller number of configurations is preferred for better performance) (Giridhar and Agrawal, 2010). A trade-off should be considered to strike a balance between these two characteristics. The configurations that have this balance are more capable of obtaining solutions with higher fitness values, e.g., GA (0.8, 0.03) since these configurations are not likely to be trapped in a local optimum. Producing solutions with low performance indicates that the available configurations lack these key characteristics such that they are trapped in a local optimum.

The takeaway from this experiment is that creating a balance between exploration and exploitation does not mean equal percentages of crossover and mutation probabilities. In the beginning, the tuning strategy is based on fixed mutation and crossover probabilities that achieved a high performance. To provide more effective SOA solutions, it is recommended that mutation values smaller than 0.1 are used. On the other hand, higher ratios should be assigned to the crossover probability, as it has been noticed that a higher crossover probability ratio may result in a greater exploitation capability and, accordingly, greater fitness values. Therefore, the experiment outcomes suggest using crossover values ranging from 0.5 to 1.0. It is also concluded that different problem contexts need a different combination of parameters. Having a set of parameters that works efficiently for some problems does not necessarily mean that this combination would work well for different problems. The scale and complexity of the problem play a significant role in finding the appropriate set of parameters.

To highlight the effectiveness and efficiency of the search-based method when using these parameters, a comparison with the baseline solutions is conducted. In a baseline solution, individuals are selected randomly for the breeding (i.e., the tournament selection selects a random member rather than the fittest). Table 5.3 presents the set of parameters that is used to conduct the experiment in the two methods. This test aims to highlight the actual performance of the representation method with the associated genetic operators and the genuine role of the fitness function in the evolution. Therefore, the null and alternative hypotheses are as follows:

H_0: Using the search method does not improve the outcomes in comparison with the baseline method.

H_1: Using the search method improves the outcomes in comparison with the baseline method.

The outcomes are the effectiveness and efficiency of the resulting candidate SOA solutions. Figure 5.9 contrasts the fitness curves for GA(0.8,0.03) and Baseline. There is a large difference between the two plots.

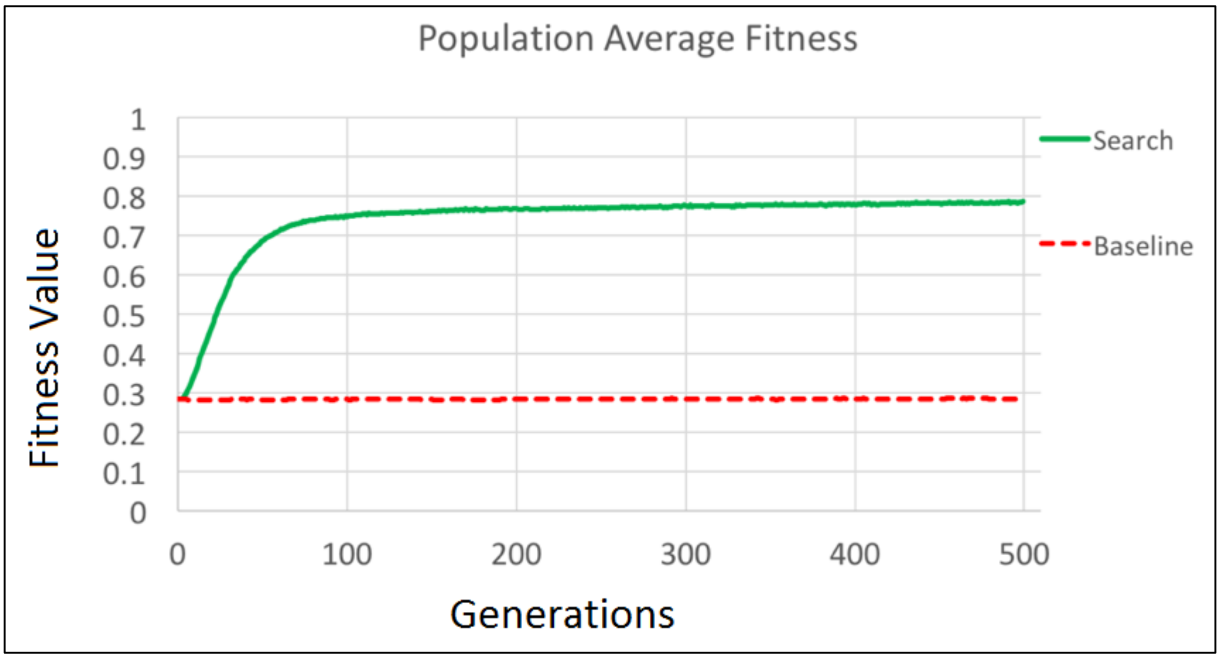

Figure 5.9: Population average fitness curves for the Search-based algorithm and the Baseline random algorithm

Figure 5.10 shows the time required to complete the search process. The Search method requires 1.018 seconds (SD= 0.284) to complete processing the 500 generations, whereas the baseline search requires 0.692 seconds (SD = 0.192) to complete the same full process.

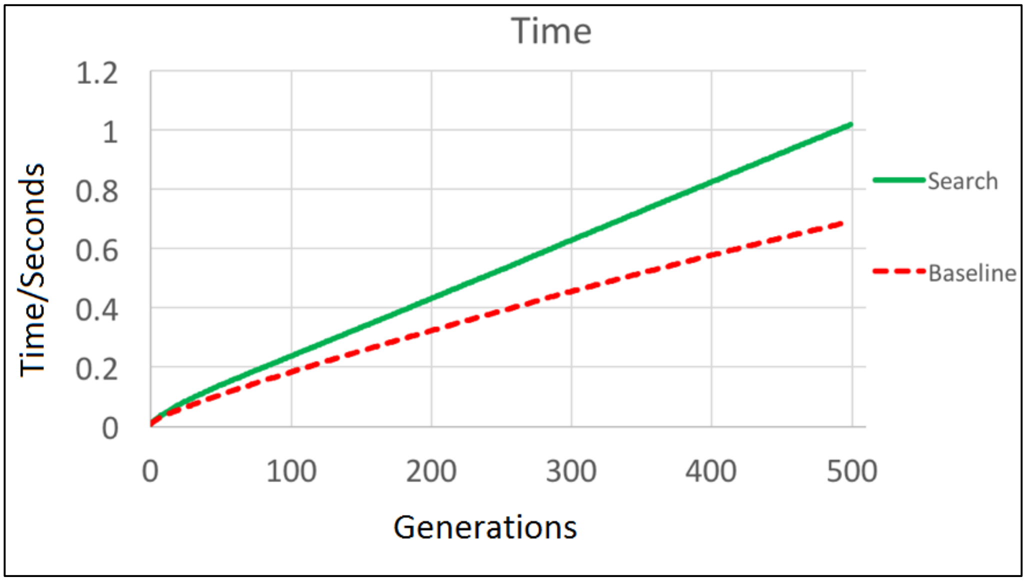

Figure 5.10: Time to finish the search

The Search sample is associated with a fitness value (M = 0.741, SD = 0.093). By comparison, the Baseline sample is associated with a numerically smaller fitness value (M= 0.284, SD = 0.001). To test the hypothesis that the Search sample in comparison with the Baseline sample is associated with statistically significant different mean fitness values, an independent samples t-test is performed. The independent samples t-test is associated with a statistically significant effect in comparison with the two methods; the Fitness $t(998) = 110.263$, $p < 0.05$. Thus, the Search outcome is associated with a statistically significant larger mean fitness value than the Baseline outcome. Cohen's d is estimated at 6.974, with effect-size r = 0.961, which is a large effect based on Cohen’s (1992) guidelines.

Table 5.5: Independent samples t-test

Method	Group Statistics			Independent Samples Test				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	Effect-size r
$Search_{Fitness}$	0.741	0.093	0.004	110.263	998	0.000 (<0.05)	6.974	0.961
$Baseline_{Fitness}$	0.284	0.001	0.000					

With regard to the computational speed of execution of the evolutionary search, results show that the search method requires approximately 1.018 seconds (SD= 0.284) to run the 500 generations using an average population fitness. However, arriving at a fitness plateau requires 0.4 seconds. This suggests that with respect to computational speed at least, initial parameter tuning achieves sufficient performance with the integer representation to provide a basis for the interactive search.

The baseline method requires approximately 0.692 seconds (SD = 0.192) to compute the full 500 generations. The computational time is smaller than the search method because with the search method the fitness value is calculated multiple times with every selection to find the fittest individual. Table 5.6 shows that $Search_{Time}$ (M=0.529, SD=0.284) is significantly larger than $Baseline_{Time}$ (M=0.38, SD=0.192). To test whether the Search sample in comparison with the Baseline sample is associated with statistically significant different mean computational speed values, an independent samples t-test was performed. The independent samples t-test is associated with a statistically significant effect in comparison with the two methods; the execution time $t(998) = 9.72$, $p < 0.05$. Thus, the baseline method is associated with a statistically significant smaller mean execution time value than the search outcome. Cohen's d is estimated at 0.615, with effect-size r = 0.294, which is a small effect based on Cohen's (1992) guidelines.

Table 5.6: Independent samples t-test for computational speed

Method	Group Statistics			Independent Samples Test				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	Effect-size r
$Search_{Time}$	0.529	0.284	0.013	9.72	998	0.000 (<**0.05**)	0.615	0.294
$Baseline_{Time}$	0.38	0.192	0.009					

In contrast, arriving at a fitness plateau requires about 0.4 of a second, which is a small value that supports the human-computer interaction.

It is concluded that the search method outperforms the baseline (i.e., random) method. This necessarily means that the fitness function does not perform at random. It is also concluded that the integer representation along with the associated genetic operators provides an effective basis for an evolutionary search of service identification. In addition, the visualisation tool has significant importance in supporting human comprehension as it reveals the connectivity between services in a simple way. These results indicate that grouping business process elements randomly can construct services with high connectivity and no specific purpose. In contrast, the metaheuristic search can group the business elements based on the real interconnection between these elements. The following section describes further experimentation that compares the effectiveness of the search approach to the results obtained by domain experts as presented in the previous chapter.

5.3.4. Experiment: Comparison with the Manual Mapping Method

This experiment aims to examine the effectiveness (i.e., fitness values) of the search method in comparison with the solutions produced manually by domain experts (denoted by "manual" solutions). The null and alternative hypotheses for this experiment are formulated as follows:

H_0: There is no significant difference between the search outcomes in comparison to the manual method outcomes.

H_1: There is a significant difference between the search outcomes in comparison to the manual method outcomes.

Manual fitness values taken from the previous chapter are provided for comparison. A set of search method samples is prepared for the comparison using the genetic parameters we arrived at in the previous experiment. All runs are conducted on a standard desktop PC running the Microsoft Windows operating system.

5.3.5. Results

Table 5.7 shows the population average fitness, coupling, and cohesion values after arriving at a fitness plateau using the search method. The average fitness value is 0.761 (Coupling=0.152, and Cohesion=0.673).

Table 5.7: Population average fitness values at a fitness plateau using the search method

Solution	Coupling	Cohesion	Fitness
Solution 1	0.1555	0.6766	0.7605
Solution 2	0.1646	0.6646	0.75
Solution 3	0.1333	0.6574	0.7621
Solution 4	0.1549	0.6712	0.7581
Solution 5	0.1515	0.6669	0.7577
Solution 6	0.1514	0.6914	0.77
Solution 7	0.1528	0.6836	0.7654
Average	***0.152***	***0.673***	***0.761***

Table 5.7 shows the fitness values for manual solutions as presented in the previous chapter. The average fitness value is 0.498 (Coupling=0.211, and Cohesion=0.208).

Table 5.8: Fitness values for Manual solutions

Solution	Coupling	Cohesion	Fitness
Solution 1	0.15	0.189	0.5195
Solution 2	0.195	0.116	0.4605
Solution 3	0.181	0.355	0.587
Solution 4	0.258	0.308	0.525
Solution 5	0.208	0.147	0.4695
Solution 6	0.263	0.134	0.4355
Solution 7	0.224	0.207	0.4915
Average	0.211	0.208	0.498

Figure 5.11 highlights the curves of average fitness values for both the search and manual samples. Note that all the search fitness values are superior to manual fitness values.

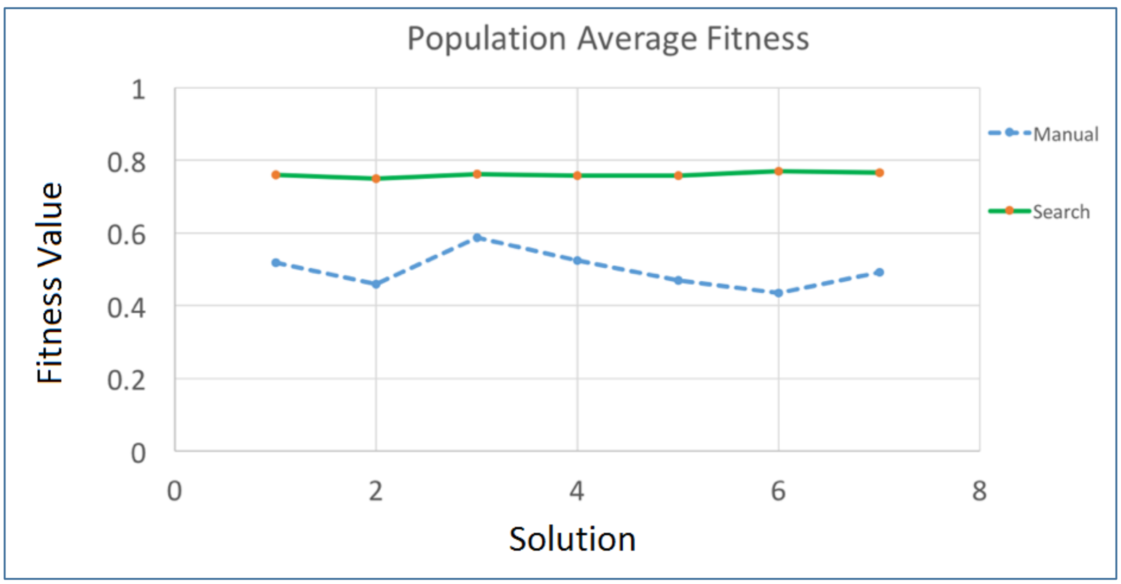

Figure 5.11: Population fitness plateau average fitness values for search and manual samples

5.3.6. Analysis

The outcomes of the Search method achieve higher fitness values in comparison with the fitness values obtained by the manual method. The larger number of configurations that are available in the search landscape allows for the production of more candidate solutions with better fitness values. In contrast, the landscape for solutions produced manually by domain experts is limited and relies on the knowledge and experience of the domain experts. This observation reflects the influence of the large scale and high complexity of the CCR case study, which make the service identification a challenging task for software engineers and domain experts. In contrast, the capability of the metaheuristic search method to find a better optimisation amongst a large population is much higher than the capability of a domain expert to find such an optimisation, which explains the large difference between the fitness values obtained using the two methods.

To compare the means of the two populations produced by two independent methods (i.e., the Search and manual), a test for the distribution of the data should be conducted to confirm that the sample data have been drawn from a normally distributed population. Due to the small sample size, a Shapiro-Wilk test for normality is performed at an alpha level of 0.05. The null and alternative hypotheses for this test are formulated as follows:

H_0: The population of data samples is normally-distributed.

H_1: The population of data samples is not normally-distributed

If the p-value is less than or equal to the chosen alpha level (i.e., 0.05), then the null hypothesis is rejected. Failing the normality test indicates that by 95% confidence the data are not normally distributed.

Results of applying the Shapiro-Wilk normality test are presented in Table 5.9. Note that for both the search and manual data samples, the p-values (denoted by Sig. column), are greater than the alpha level. Moreover, the skewness and kurtosis are within acceptable limits of ±2 (Trochim and Donnelly, 2006; Field, 2000; Field, 2009; Gravetter and Wallnau, 2014). Thus, the null hypothesis is accepted, which implies that the data are normally-distributed and that indicates the possibility of performing a parametric test such as the t-test.

It is recommended that a *t*-test is used to compare the means of two populations from different methods (i.e., independent methods).

Table 5.10 presents the results of conducting the statistical analysis test on the two samples sets.

Table 5.9: Normality test results

Method	**Descriptives**				**Shapiro-Wilk**		
	Mean	**SD**	**Skewness**	**Kurtosis**	**Statistic**	**df**	**Sig.**
$Search_{Fitness}$	0.7605	0.0063	-0.225	0.697	0.980	7	0.961
$Manual_{Fitness}$	0.4984	0.0505	-0.723	0.416	0.958312	7	0.804

Table 5.10: Independent samples t-test

Method	**Group Statistics**			**Independent Samples Test**				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	Effect-size r
$Search_{Fitness}$	0.761	0.006	0.002	13.64	12	.000 (< **0.05**)	7.39	0.965
$Manual_{Fitness}$	0.498	0.050	0.019					

The $Search_{Fitness}$ group (N = 7) is associated with a fitness value (M= 0.761, SD=0.006). By comparison, the $Manual_{Fitness}$ group (N = 7) is associated with a numerically smaller fitness value (M=0.498, SD=0.05). To test the hypothesis that the $Search_{Fitness}$ in comparison with $Manual_{Fitness}$ is associated with statistically significantly different mean fitness values, an independent samples *t*-test is performed. The independent samples *t*-test is associated with a statistically significant effect in comparison with the two methods; the Fitness $t(12) = 13.64$, $p < 0.05$. Thus, the $Search_{Fitness}$ samples are associated with a statistically significant larger mean fitness value than the $Manual_{Fitness}$ samples. Cohen's d is estimated at 7.39 with effect-size r estimated at 0.965, which is a large effect based on Cohen’s (1992) guidelines.

Based on the results and analysis, the metaheuristic search method outperforms the manual method. The large scale and high complexity of the case study play a significant role to highlight the advantages of the metaheuristic search over the manual method. The search is a fully automated comprehensive top-down framework that comprises all the phases of the service identification. It is concluded that the representation method, with the associated genetic operators, has implemented the service constraints such that the resulting solutions are feasible and adhere to the SOA principles of low coupling and high cohesion. The outcomes verify that the search-based method represents a robust basis for an interactive search.

5.4. Evaluation

In this section, the search method will be evaluated by means of triangulation that combine quantitative and qualitative techniques (Breitmayer, Ayres and Knafl, 1993; Greene and McClintock, 1985). Triangulation, however, helps researchers to capture a more complete, holistic representation of some parts of the problem that could be neglected by a single method (Greene, 1989). Qualitative methods, in general, play a significant role in collecting data and suggesting conclusions to which other methods are blind. Moreover, an additional technique could be used not just to assess the results of a method, but also to enrich understanding by allowing a deeper dimension to emerge. For greater accuracy to improve the results of the evaluation, a case study is adopted with the search-based method (Prat et al., 2014).

A generic evaluation approach is proposed by Harman and Jones (2001) to evaluate SBSE techniques such as genetic algorithms, simulated annealing, and tabu search. As presented in chapter 3 of this thesis, the evaluation includes different techniques to validate the search method. These evaluation techniques include a comparison with baseline methods, benchmarking, and a consideration of the aesthetic side of the outputs. The comparison with both the baseline and the manual methods is presented as a part of section 5.3. Benchmarking and the evaluation by domain experts at KHCC techniques will now be discussed.

Some of the evaluation activities focus on inspecting the correctness of the framework structure (i.e., correctness means consistency and completeness). Other methods include, for example, a unit test, and checks to see whether the constructed functions and genetic operators (e.g., crossover, mutation, and selection) are free of errors (Sharma et al., 2014). The experiments presented in section 5.3 confirm that these functions have successfully passed the unit tests since the search was run with no syntax or run-time errors. The different experimental trials conducted in this chapter and the previous one suggest that the search method is fully verified. These tests include the observation of outcomes which are produced precisely as anticipated, the inspection check conducted in the previous chapter to highlight the consistency of inputs and outputs, and the observation of the behaviour of GA when using different parameters.

5.4.1. Validation by Domain Experts at KHCC

Using the quantitative design metrics is not enough to reveal the quality of the candidate solutions. Therefore, using the implicit knowledge and experience of domain experts is anticipated to provide a suitable assessment of the quality of the search outcomes. This evaluation is utilised to ensure that the search framework is capable of producing good-enough SOA solutions.

Five domain experts at KHCC were recruited to study the resulting services of sample solutions and then evaluate them using the Likert satisfaction scale (Likert, 1932). The key driver of the evaluation is the implicit and explicit knowledge and experience of domain experts such that they know the actual relationships between the entities. These experts are capable of confirming whether the candidate solutions are feasible or not. Using the assessment by domain experts has an additional advantage in that the correctness of the candidate SOA solutions can be diagnosed. Lastly, at the conclusion of the evaluation session, each participant is invited to provide any comments on their overall solutions. Such comments might include any detailed justifications and any suggestions for enhancement of the overall experience. Table 5.11 presents the Likert-type satisfaction scale that was used in this evaluation.

Table 5.11: Likert-type Satisfaction Scale

Scale Number	Description	Value (%)
10	Very Satisfied	100
9		90
8	Satisfied	80
7		70
6	Neutral	60
5		50
4	Dissatisfied	40
3		30
2	Very Dissatisfied	20
1		10
Void Response	N/A	0

To confront the challenge that each domain expert has knowledge of his/her domain but not the entire process model of the case study, the experts are asked to work in two groups. Two sample SOA solutions are chosen at random for the evaluation after performing the search for 50 runs. Table 5.12 presents the first sample solution that was given to the first group for evaluation. The first column represents the service number, the second column shows the SSEs that construct each service and the last column shows the feedback provided by the participants on each candidate service in a scale of 0-100. This solution comprises 11 services. The experts in the first group evaluated the first solution such that the average evaluation value is 81.82%.

Table 5.12: Evaluation values by the first group of domain experts at KHCC

Service ID	Main Role and Functions	1st Group Feedback
Service 1	. Medical Records . Receptionist {Patient Registration} . Receptionist (department-specific) {DB and Reports}	70%
Service 2	. Registrar {Connection with JCR DB}	100%
Service 3	. Patient {Visit Admission Department} . Combined clinic (specialists) . Patient {Visit Specialist} . Admission clerk . Specialist	80%
Service 4	. Lab . Patient {Lab visit and payment}	70%
Service 5	. Patient {Radiotherapy visit and payment} . Radiotherapy department	90%
Service 6	. Receptionist (outpatient clinic) . Patient {visit clinic and payment} . Receptionist (Inpatient care) . Patient {visit combined clinic}	90%
Service 7	. Doctor (Diagnostician) . Patient {book an appointment and visit doctor} . Receptionist (Cancer detection unit) . Medical records clerk . Chemotherapy department . Patient {Pay and receive chemotherapy treatment}	60%
Service 8	. Handle a Patient General Reception {handle appointment requests} . Handle a Patient General Reception {register patients and book appointments}	80%
Service 9	. Receptionist (outpatient department) {Patient lists} . Receptionist (Outpatient clinic) {Patient files} . Registrar {Updates on patient records and files}	90%
Service 10	. Accounts clerk {Patient financial state} . Inpatient care specialists and nurses	70%
Service 11	. Patient {Pay and visit imaging department} . Imaging department	100%
	Mean (Fitness)	81.82%

Table 5.13 shows the second sample solution that is revealed to the second group of experts for evaluation. Note that this solution includes ten candidate services. The detailed functions comprising each service are presented in the second column. The results of the evaluation reveal that the average evaluation value for this solution is 78%.

Domain experts have confirmed that the resulting services are feasible and these services conform to SOA principles. This shows that the search-based framework is capable of producing valid SOA candidate solutions that can be accepted by experts.

Table 5.13: Evaluation values by the second group of domain experts at KHCC

Service ID	Main Role and Functions	2^{nd} Group Feedback
Service 1	. Combined clinic (specialists) . Patient {Visit Specialist}	80%
Service 2	. Registrar {Connection with JCR DB} . Registrar {Updates on patient records and files} . Medical records clerk	90%
Service 3	. Patient {Visit Admission Department} . Admission clerk . Medical Records . Receptionist {Patient Registration} . Receptionist (department-specific) {DB and Reports} . Specialist	60%
Service 4	. Lab . Patient {Lab visit and payment}	100%
Service 5	. Patient {Radiotherapy visit and payment} . Radiotherapy department	70%
Service 6	. Receptionist (Inpatient care) . Patient {visit combined clinic}	80%
Service 7	. Doctor (Diagnostician) . Patient {book an appointment and visit doctor} . Receptionist (Cancer detection unit) . Receptionist (outpatient clinic) . Patient {visit clinic and payment}	60%
Service 8	. Handle a Patient General Reception {handle appointment requests} . Handle a Patient General Reception {register patients and book appointments} . Accounts clerk {Patient financial state}	80%
Service 9	. Receptionist (outpatient department) {Patient lists} . Receptionist (Outpatient clinic) {Patient files} . Chemotherapy department . Patient {Pay and receive chemotherapy treatment}	70%
Service 10	. Patient {Pay and visit imaging department} . Imaging department . Inpatient care specialists and nurses	90%
	Mean (Fitness)	78%

The average evaluation for the two solutions is 79.91% which indicates a good level of satisfaction according to the Likert scale. This evaluation suggests that the resulting candidate services are close to the expectations as it reflects a good satisfaction level about the quality of the resulting SOA solutions.

Grouping different domain experts in order to collaborate reduced the bias and developed a wider comprehension of the workflow. Domain experts justified the low evaluation value assigned to some services (e.g., service 7 in the 1st group, and service 3 in the 2nd group) due to the lack of harmony between the functions that comprise these services. In addition, some of the activities that run in the background are not very obvious to all domain experts. Thus, low evaluation values are assigned to some services for that reason. Senior domain experts justified the difficulty of assessing some services by stating that some changes had been applied to the original workflow. In these special cases, the experts evaluated the services based on the abstract functionalities rather than the modified activities inside the services. This observation emphasises the importance of interactive search, as there is a need for domain experts to steer the trajectory of the search in order to reach a higher level of satisfaction.

However, the highest possible evaluation value has been assigned to some services (e.g., services 2 and 11 in the 1st group, and 4 in the 2nd group). It was justified by domain experts because of the close relationship between the different activities comprising these services and the capability of these services to fulfil their functions without the need for other services. This reasoning reflects the quality of the candidate services that can be produced and shows that embracing the coupling and cohesion design metrics produces satisfactory solutions.

A further observation is that the 10-number scale is helpful as it shows the level of satisfaction in detail and enables the clustering of answers into groups (i.e., 7-10 high level of satisfaction, 5-6 is a normal level, and below 5 is a low level). However, using a long scale forces experts to spend more time finding the most accurate result. This issue will be taken into account when allowing participants to interact with the search engine to reduce human fatigue.

5.4.2. Benchmarking

This section continues to compare the search framework with the BPAOntoSOA. In Chapter 4, the proposed framework was compared to the BPAOntoSOA framework in aspects such as the abstraction level of inputs and conformance to the SOA principles. In this iteration, different aspects are compared such as the level of automation, the multiple runs, the feasibility of resulting solutions, and the format of the outcome. Table 5.14 presents the differences between the two frameworks according to the different characteristics.

Table 5.14: Comparison between BPAOntoSOA and Metaheuristic Search frameworks when identifying services from BPMs

Comparison	BPAOntoSOA	Search-Based
Level of Automation	Semi-automated.	Fully automated.
Multiple Runs and Correction Actions	The static algorithm produces only one solution.	The stochastic algorithm produces multiple solutions at each run.
Feasibility of Solutions	The software engineer should have the experience to check whether the resulting solution is feasible or not.	The representation technique implements the constraints that guarantee the feasibility of the resulting solutions.
Format of Outcomes	List of SOA candidate services.	Full SOA solutions, and the WSDL files.

Automation is an important characteristic that is enabled by the metaheuristic search framework for service identification. Although, as the general purpose of this research is not the automation as the service identification will use the human interactive preference, the other phases are fully automated. That includes the data preparation phase, which is considered a challenging task because of the complex nature of input models. In addition, the activities of service identification phase that include grouping SSEs based on their relationships and calculating the fitness values have been automated. The search engine searches for solutions, measures the fitness, and evolves the solutions until arriving at solutions with high fitness values. Furthermore, the third phase (i.e., refinement of candidate services) in which WSDL files are produced from the candidate services, is automated as well. In contrast, the BPAOntoSOA uses an algorithm to identify services, but this algorithm relies on human involvement to prepare the input data and the outcomes in the final form. The search method outperforms the BPAOntoSOA in terms of automation as the search fully automates all the activities of service identification.

With reference to the single or multiple runs of the algorithm, on the one hand, BPAOntoSOA uses a static algorithm (i.e., denoted by RPA clustering algorithm) that produces the same exact solution each time it runs. The main drawback here is the limited solutions provided by the RPA algorithm. If the software engineer is not satisfied by the resulting candidate solution, then the resulting candidate services of BPAOntoSOA framework become useless. On the other hand, the stochastic search-based method has the capability of producing different feasible solutions with each run, even when using the same combination of parameters (e.g., crossover and mutation probabilities). This superb advantage allows the search method to produce numerous feasible candidate solutions, such that the expert is most likely to accept them.

Respecting the feasibility of resulting solutions, this issue has a relation with the conformance to the SOA principle point discussed in the previous chapter. The additional point here relates to how each method applies the constraints that guarantee the feasibility of the resulting solutions. The BPAOntoSOA defines the targeted SOA principles and the RPA algorithm tries to stick to these principles but in the end, it relies on the experience of the software engineer to confirm that the resulting solutions are feasible. In the event that the resulting solution is not feasible, the targeted SOA principles will be affected. This needs the involvement of the software engineer to take action to ensure the solution is feasible, e.g., to delete the conditional relationships between services. However, in the search method, the representation technique with the associated genetic operators have implemented the constraints that make the resulting services feasible. Consequently, the check for solution feasibility is done automatically. This advantage makes the search more powerful and efficient. Therefore, regarding the feasibility of solutions, the metaheuristic search method has an advantage over BPAOntoSOA method.

With regard to the format of the outcomes, the BPAOntoSOA produces a list of SOA services. Producing a full SOA solution from these candidate services requires a further step in which the detailed processes should be presented. This limitation refers to the abstraction level of inputs. In contrast, having the BPMN as an input provides the capability to produce candidate services with the full detail required to produce SOA solutions. As a result, the decoder converts the candidate services into a SOA solution with the WSDL files. With regard to the format of outcomes, the search outperforms the BPAOntoSOA.

Performing this comprehensive comparison between the search-based and the BPAOntoSOA SIMs, it is concluded that the search outperforms BPAOntoSOA in terms of automation, the ability to generate multiple novel solutions, the feasibility of resulting services, and the format of the outcomes. These improvements satisfy the benchmarking part of the SBSE evaluation criteria (Harman and Jones 2001).

5.5. Summary

This chapter has addressed RQ3 in the fulfilment of two objectives: (i) test the representation method to inform whether it has the capability of mapping the BPMN on to corresponding SOA solutions (ii) and explore and exploit the search space in an effective and efficient way until arriving at a good combination of parameters.

With regard to the first objective, this chapter presented the development and testing of the GA-based search engine. The first experiment presented in the demonstration phase had successfully tested the search engine using a set of initial parameters. That experiment informed the validity of the search method, and in addition, it helped to tune the genetic parameters for the next experiment. Moreover, it has encouraged more investigations in order to find better parameter combinations to help in creating more knowledge about the effect of the genetic parameters.

Regarding the second objective of this chapter, the aim of the first experiment was to arrive at a good combination of genetic parameters that produced efficient and effective solutions. Empirical trial-and-error parameter tuning was utilised to strike a balance between exploration and exploitation. The experiments in this chapter have revealed that a good combination of parameters would help to explore and exploit the search space more effectively. The statistical analysis clearly emphasises that the search method outperforms both the baseline and manual methods.

The metaheuristic search-based framework has also been evaluated using benchmarking and domain experts at KHCC. Evaluation results confirm the capability of the search method to produce feasible SOA solutions. In addition, the entire process of service identification was automated. The optimisation method helps to apply the correction actions that generate more optimised solutions. The feedback from domain experts provided an acceptable degree of satisfaction. However, feedback from domain experts also revealed that human preference is anticipated in order to construct more acceptable solutions.

Overall, the findings of this chapter suggest that the search-based framework presented in this chapter has provided the appropriate characteristics that represent an effective and robust basis for an evolutionary interactive search of candidate SOA solutions. However, it is necessary to investigate techniques to facilitate the utilisation of user evaluation to enable the interactive search capability. Therefore, the following chapter describes experiments in the interactive search that are used to enrich the quality of the search outcomes.

Chapter 6
The Interactive Search

6.1. Introduction

Experiments in this chapter investigate the impact of the human interactive preference on the search outcomes. It is undeniable that software engineers struggle with numerous trade-off judgments as they formulate candidate software services as a basis for service-oriented architecture (SOA) solutions. Therefore, to assist the user with decision making, interactive metaheuristic search techniques (e.g., evolutionary algorithms and ant colony optimisation) with the human 'in-the-loop' have been recently studied and shown promising results (Simons et al., 2014). Different research attempts combined the qualitative user evaluation with the quantitative machine-calculated values to steer the direction of the search (Simons and Parmee, 2012). Furthermore, it has been found that service identification methods (SIMs) lack this combination of assessment that combines the evaluation of the software engineer with quantitative fitness values. Therefore, the objective of this chapter is to address the fourth research question (i.e., RQ4):

"What is the impact of the human interactive preference on the search outcomes?"

Addressing this research question requires the design and development of an interactive method that provides the capability to combine participant's feedback with the quantitative values in order to enrich the quality of the resulting solutions and enhance the performance of the search (Section 6.2). RQ4 triggers a set of sub-questions, such as: what are the appropriate techniques when asking for the participants' feedback? How can that value be used to steer the trajectory of the search? How can human fatigue be reduced when inquiring for feedback? And would human interaction help the search to arrive more quickly at good-enough results? Therefore, the major contribution of this chapter is to explore the potential impact of using the interactive search-based framework (i.e., BPMiSearch) on the resulting outcomes. These outcomes include the effectiveness (i.e., the fitness values), the efficiency (i.e., time of arriving at good-enough solutions), and the level of satisfaction regarding the resulting solutions.

A case study is used to demonstrate the BPMiSearch (Section 6.3), which provides an opportunity to evaluate the novel framework. In addition, domain experts have been recruited to interact with the BPMiSearch to satisfy the requirements of the corresponding research question. In addition, the domain experts play an important role in some evaluation techniques as presented in Section 6.4.

6.2. DSRM Design and Development Phase

This section explains the design and development stage of the proposed interactive method for service identification. Building on the outcomes of the previous chapter, the same representation along with the genetic operators has been adopted. In addition, the same quantitative fitness measurements are utilised after tuning the weights of the design metrics such that the fitness function would combine the qualitative feedback with the quantitative values. Expressing human feedback and then using it to steer the trajectory of the search is not a trivial task. This section presents the difficulties and challenges that should be managed in order to conduct a successful interactive experiment.

6.2.1. The Proposed Framework

The focus of this section is to find an appropriate mechanism to integrate human judgment into the search process in order to explore and exploit the search space within an interactive, human-in-the-loop search. It is important to combine the preference values with the quantitative design metrics (i.e., coupling and cohesion) to allow the participant to steer the trajectory of the search. It is hypothesised that human knowledge and experience will help the search algorithm to find more effective and efficient solutions that satisfy the participants.

Human preference can be distinguished into three categories; priori preference (i.e., before the search process), posteriori preference (i.e., after the search), and interactive preference (i.e., during the search) (Aljawawdeh, Simons and Odeh, 2015). The priori preference includes activities such as configuring the search engine with the needed set of parameters, therefore, this requires that the participant should be aware of the nature of the genetic algorithm and have knowledge and experience in the configuration of the search algorithm along with its parameters. The second type is the posteriori, and it includes for activities such as selecting a good-enough solution among a set of candidate solutions. These two types of preference are out of the scope of this research, which indicates that knowledge of the nature of the search and genetic algorithm is not required in the participants. The focus of the experiments is on conducting an interactive search in which the participant provides his/her feedback while the search process is still active. The computational algorithm searches for good-enough solutions and during this search process, some solutions should be presented to the participant for evaluation. The algorithm should use the participant's feedback to learn and find better solutions. This responsive process allows the participant to drive the trajectory of the search to find solutions with better quality.

Participants' feedback (i.e., the implicit knowledge and experience) suggests that positive interaction with the search might enrich the quality and appearance of the search outcomes (Simons and Parmee, 2012). This section conducts an investigation into how to incorporate domain experts in order to evaluate the SOA services, and then integrate their feedback with the quantitative evaluation of the targeted services. The human preference can be expressed before, during or after the search (Aljawawdeh, Simons and Odeh, 2015); however, the scope of this research investigates only the interactive search (i.e., during the search process).

Because of the large number of candidate solutions that comprise the search space, selecting presentation individual solutions to be evaluated by the participant is challenging. The metaheuristic search is ordinarily population-based; however, selecting all the population individuals for evaluation leads to the production of inconsistent and ineffective results. This causes evaluation fatigue and places a heavy load on participants rather than supporting them to make the right decisions (Shackelford and Simons, 2014).

Four key challenges need to be overcome to develop an interactive search-based framework. Firstly, there is the selection of representative solutions for presentation. Secondly, it is necessary to find a proper mechanism to derive the subjective feedback from the participants, and quantify the preference values such that the evaluation is as accurate as possible. Thirdly, it is important to find an appropriate mechanism to evaluate the other individuals in the population in light of the value assigned to the presentation individuals. Finally, it is necessary to reduce human fatigue while interacting with the BPMiSearch tool. A flexible and friendly graphical user interface (GUI) for the visualisation tools is a significant factor that supports addressing the four key challenges more efficiently. A well-designed GUI should present the services with all the required data, reveal the search progress, and encourage the participant to make the right decisions. In addition, using colourful graphs to represent service solutions would help the participants to understand the relationships between services and make accurate decisions.

With regard to the first challenge, i.e., the selection of representative individuals for presentation, Shackelford and Simons (2014) suggest a set of solutions based on the context of the problem and the experiment's variables (e.g., population size). The suggested solutions include reducing the population size and presenting all individuals or selecting some individuals for presentation based on a specific criterion (e.g., randomly, sequentially, or based on a clustering method). An alternative solution is to select presentation individuals on a regular basis after a fixed number of generations or according to the fitness value achieved by the individuals (Shackelford and Simons, 2014). The scale and complexity of the case study are important factors when choosing the appropriate criterion in selecting presentation individuals.

Presenting all the population individuals is not a suitable technique in this research as the case study is classified as medium/large. Using a uniform mechanism to select the presentation individuals is more suitable. In addition, showing presentation solutions iteratively as the search progresses is expected to reflect the progress of evolution which is beneficial in the interactive context. Therefore, the technique of adoption is based on selecting presentation individuals frequently, but using a special selection criterion (i.e., tournament selection). A solution will be selected after executing 100 generations by selecting seven samples at random, then choosing the fittest amongst them. The advantage of a frequent selection is that it provides participants with a sense of evolution over time, which increases positive feedback as they experience progress, whereas the tournament selection which provides a uniform way for selection provides an opportunity for each solution in the population to be selected. Consequently, the tournament selection is preferred in combination with the frequent selection.

Secondly, there is the challenge of eliciting and quantifying the participant's feedback efficiently. In the BPMiSearch framework, the participant evaluates a candidate service by assigning a number (i.e., from the Likert scale) to that service. Using a number-scale provides an excellent simplicity advantage to the technique of eliciting user evaluation since almost all cultures around the world are familiar with the number-scales. In addition, number-scales support the use of values to conduct a statistical analysis which is considered a strength point. Selecting the suitable size for the number-scale is very important to help participants justify their opinions accurately.

A medium size satisfaction scale is considered one of the most reliable scales, as based on various research studies it performs the most effectively (Van et al., 2004). In addition, it provides the required level of detail that helps to analyse what a user is trying to say (Snelick et al., 2005). Based on these facts, the Likert scale rating (i.e., medium-size satisfaction scale) has been chosen. Table 6.1 presents the interpretation of the scale values that allows participants to share their experience and evaluate each service. This scale ranges from "Very Satisfied" to "Very Dissatisfied". Each choice is converted to the corresponding value as the table suggests. This satisfaction scale will be implemented in the search engine tool and will be used by the participants to evaluate the services.

Table 6.1: Likert Satisfaction Scale

Scale Number	Description	Evaluation Value (%)
5	Very Satisfied	100
4	Satisfied	80
3	Neutral	60
2	Dissatisfied	40
1	Very Dissatisfied	20
Void Response	N/A	0

Figure 6.1 shows the GUI screen that will be used to present the services. This "Main Evaluation Window" has been designed to show the presentation solution and allow the participant to evaluate it. Each service in the SOA solution is presented in a box labelled by the service number and service ID. In each service box, the major abstract functionalities are presented. The participant can view the details of each specific function by clicking on it with the mouse pointer after which a small window will appear that reveals the set of the tasks and activities that comprise the targeted function as presented in Figure 6.2.

The main evaluation window also shows the total fitness of the solution which indicates the quantitative values of that solution. In addition, it offers two ways to evaluate a solution: (i) evaluate the candidate solution as a whole using the slider at the bottom of that main evaluation window, or (ii) evaluate each service individually.

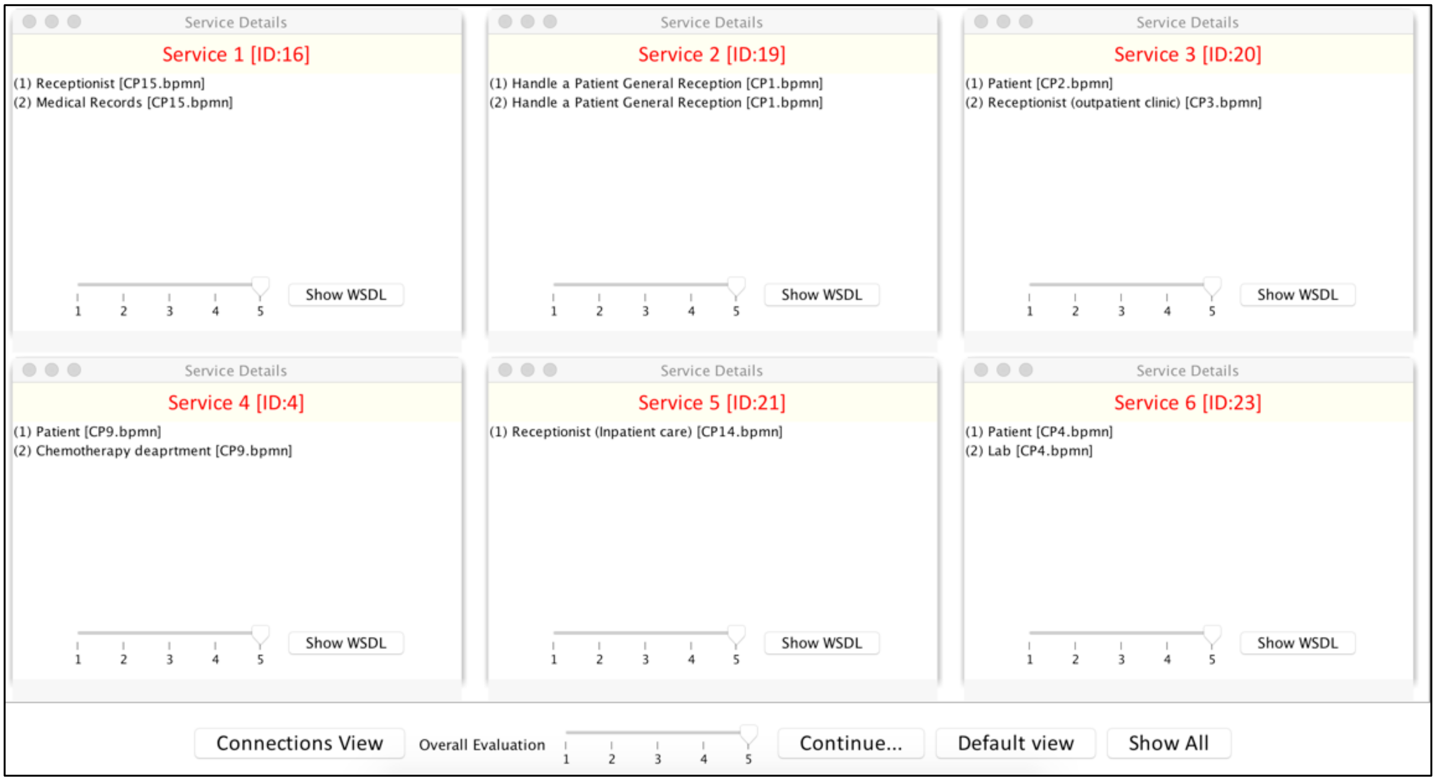

Figure 6.1: Service Evaluation Screen

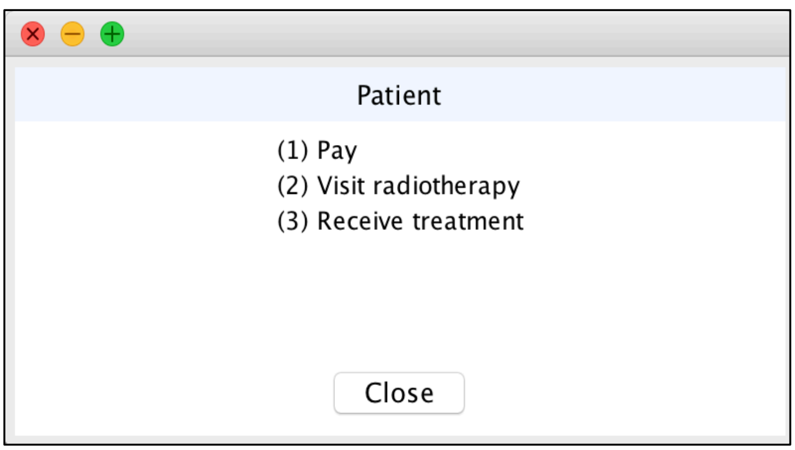

Figure 6.2: Service Details GUI

Another useful tool that helps the participant to make the right decision before evaluating the services is the "Connections-View". This tool (it can be accessed from the main evaluation window) is a graphical visualisation tool that demonstrates the connections between the services and uses different colours for services based on their level of connections. Figure 6.3 depicts a sample connections-view for a presentation solution. The direction of each arrow reveals the source (i.e., sender) and destination (i.e., receiver) of the connection. In addition, the colours of the services indicate the range of connections that that service initiates (i.e., green: stand-alone or 0 connections; blue: up to 2 connections; orange: up to 3; and red: 5 or more connections). Note that the red-coloured service that has a high level of connectivity implies that this service comprises the most important functions in the resulting solution. In other words, this service can be the core candidate service that includes the most important functionalities in the resulting SOA model.

Using the connections-view and the fitness values before evaluating the services is recommended as it helps to make an accurate assessment of the entitled services. The evaluation value of each service is scaled to a value between 0 and 1. The feedback values are recorded in order to be integrated with other fitness measures in order to steer the trajectory of the search.

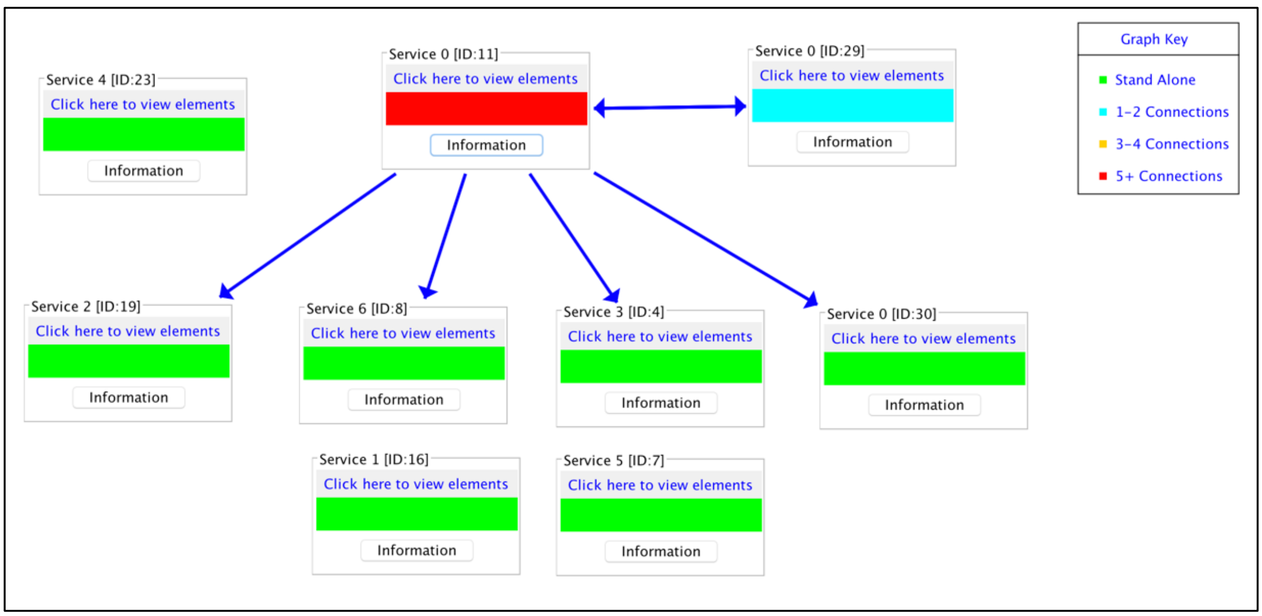

Figure 6.3: Connections-View

The third challenge is to find a suitable mechanism to evaluate all the individuals in the population by using the evaluation assigned to the presentation individuals. As explained previously, evaluating every single individual in the population is not an option. However, in light of the alternative solutions discussed by Shackelford and Simons (2014), different approaches can be developed. For example, same feedback value on all population individuals could be assigned considering that the presentation individuals represent the whole population. This solution indicates that the participant considers all the individuals of a population as having the same quality. The main drawback to this solution is that it does not distinguish between good or bad solutions. The goal is to assign a higher value to some solutions such that they would have a higher chance of living in the next generation. Therefore, it is concluded that this solution is ill-suited for this research. Another alternative approach is to assign the evaluation value that was obtained by the presentation individuals to the identical solutions in the population. This solution is more accurate as it considers the similarities between the presentation individuals and other individuals in the population. However, this solution has two weaknesses; (i) it is inefficient as it takes a long period of time to compare presentation solutions to other individuals in the population, (ii) the existence of identical solutions is not guaranteed as the number of elements that comprise the services is large, which results in a considerable number of differences among the candidate solutions.

To overcome this challenge, a trade-off should be made. By neither assigning the feedback value to all solutions in the population nor searching for identical solutions to assign them with an assessment value, the problem is hypothesised to effectively and efficiently be solved. The solution is to strike a balance between the two techniques. Therefore, a novel technique is proposed to combine the advantages of the two approaches. The proposed technique is based on assigning the evaluation value to a group of selected individuals with similarities (i.e., not necessarily identical solutions, but there are some identical services). These individuals are the sample solutions that are selected using the tournament selection in the previous step. These individuals probably have some similarities in terms of fitness values and the distribution of SSEs. Consequently, the assessment value is assigned to individual services (i.e., not the whole solution) according to the similarities between the enclosed SSEs. If two solutions have some identical services (i.e., with the exact SSEs), then these services will be assigned the same fitness values. The qualitative feedback value is integrated with the quantitative fitness measures (i.e., coupling and cohesion) as depicted in Equation 6.1. Two factors support this method: efficiency and accuracy. Since the two factors may work in opposition to each other, a balance between them should be considered. For example, a small selection sample (e.g., 2 or 3) supports efficiency but reduces accuracy, whereas large selection samples (e.g., 10 or 12) are more accurate but less efficient. However, empirical parameter tuning using a trial-and-error mechanism can be used to determine the appropriate selection size.

$$Fitness(x) = (Weight_{coupling} * \left(1 - CpF(x)\right) + (Weight_{cohesion} * ChS(x) + (Weight_{human} * HumanEvaluation(x)) \quad (6.1)$$

Finally, reducing human fatigue while performing the interactive process, could be one of the most challenging tasks. Clearly, human fatigue may cause systematic errors and biases. If the participants feel tired, the results will not be accurate. Therefore, to overcome this issue, a set of procedures should be considered such as minimising the number of interaction times when completing the interactive search. Generally, the number of interactions is preferred to be of medium size (i.e., between 3-7 times) (Van et al., 2004). However, the size and complexity of the case study and ensuring that participants, who have a busy schedule, are not overloaded by the number of time slots also play an important role in determining the number of interactions. A further factor that should be considered is the efficiency of the search process; this can be managed using an appropriate set of genetic parameters. Well-implemented software that has a short execution time would help to reduce human fatigue. Another important factor is to ensure that the participants feel comfortable about the physical environment in the lab. It is recommended that the length of a single experiment session should be between 60 to 90 minutes or shorter, otherwise participants may become frustrated (Nielsen, 2005). During the experimental sessions, the participants should be allowed to take appropriate breaks as required. Finally, a “friendly” design for the GUI allowing for easy access to all the required information and without any complications would certainly minimise human fatigue (Simons and Parmee, 2009).

The BPMiSearch framework has been designed to address the four challenges. Next section demonstrates this interactive BPMiSearch framework using the CCR case study.

6.3. DSRM Demonstration Phase

This section represents the demonstration phase of the DSRM third iteration (i.e., the interactive search). Demonstrating the BPMiSearch engine using the CCR case study with the human in the loop is required to show the impact of the human interaction on the search results.

The experiment presented in this section aims to examine the algorithm behaviour in an interactive context. The experiment investigates the quantitative outcomes (i.e., effectiveness and efficiency) of the search algorithm with the human-in-the-loop in order to examine the impact of the human interaction on the search outcomes in comparison with the experiment conducted in the previous chapter.

The full plan of this experiment (i.e., includes details about the project, data segments, participants, data protection, the privacy of participants, and an action plan) is presented in Appendix E. The instructions material and details that introduce this experiment to the participants is presented in Appendix F.

6.3.1. Experiment: The Interactive Search

This experiment is designed to measure the difference between the interactive search (i.e., denoted by BPMiSearch) and the non-interactive search (i.e., denoted by search) with reference to the search outcomes. Typical types of independent variables that can be observed include the efficiency, the effectiveness and subjective satisfaction (Lazar et al., 2010). Effectiveness describes the fitness values, while the search efficiency describes the speed of completing a task (i.e., or speed of arriving at good solutions), and can be measured in terms of the time, or number of generations, to complete the search. Therefore, the focus of this experiment is to measure the influence of the interactive preference on the effectiveness and efficiency of the resulting solutions.

It is anticipated that human interaction with the search may affect the quality of the resulting solutions as the implicit knowledge of experienced participants enables the search to construct better-optimised solutions. Furthermore, with the support of human feedback to steer the search directions, arriving at candidate solutions may quicken. The null and alternative hypotheses are thus formulated as follows:

> **H_0: there is no difference in solution outcomes between the BPMiSearch (i.e., with human interaction) and the search (i.e., without human interaction).**
>
> **H_1: there is a difference in solution outcomes between the BPMiSearch (i.e., with human interaction) and the search (i.e., without human interaction).**

The outcome of the interactive experiment depends on both the behaviour of the search algorithm in addition to the feedback from the participant who evaluates the presentation solutions. Clearly, the level of experience and knowledge of the participants play an important role in directing the search to find better solutions. The independent and dependent set of parameters for the experiment are presented in Table 6.2.

Table 6.2: Parameters of the Search Experiments

Experiment	**Parameter**	**Values**
BPMiSearch vs search	Independent	Search GA, BPMiSearch GA, Genetic Parameters (Population size, number of generations, crossover probability, and mutation probability)
	Dependent	- Effectiveness (i.e., fitness values) - Efficient (i.e., generations required to arrive at a fitness plateau)

The independent variables include the values of the genetic parameters obtained in the experiments conducted in Chapter 5. Using these parameters, the quantitative search has successfully produced effective and efficient solutions as presented in Chapter 5. Therefore, the same parameters will be used with the interactive search to enable a valid comparison.

6.3.2. Data Segments and Participants

The participants are authorised to retrieve all the necessary details such as details about the SSEs, the original BPMN documents, and the links between services through the connections-view tool. Participants have access to the data segments for the entire case study to ensure that their decisions are accurate.

Seven professional domain experts from different departments and with different experience levels at the King Hussain Cancer Centre (KHCC) were recruited to participate in the trials using the BPMiSearch. The participants occupy managerial positions in the organisation and consequently have a good comprehension of the business processes of KHCC, and were therefore selected for this experiment. To examine the effectiveness and efficiency of the interactive method, at each iteration, a record is stored containing sufficient details about the population fitness values along with the standard deviation. At the time of evaluation, the participants were invited to evaluate the presentation solutions.

After each interaction, the qualitative assessment is integrated with the quantitative fitness values to steer the trajectory of the search. This process is repeated multiple times to ensure the reliability of the results. However, to prevent user fatigue, a short break is taken after one hour and each participant session is concluded after a maximum of two hours whether or not the planned schedule has been completed or not.

Lastly, at the end of the evaluation session, each participant is invited to provide any comments on their overall human experience of the trial. Such remarks might include any detailed justifications, any satisfying qualities, and any recommendations for enhancement of the overall human experience.

6.3.3. The Experimental Setup

The genetic parameters introduced in Chapter 5 were used to run this interactive experiment. To briefly recap, these parameters are presented in Table 6.3.

Table 6.3: Genetic Parameters

Parameter	Value
Selection	7
Breeding pool size	1.0
Number of generations	500
Population size	100
Crossover probability	0.8
Mutation probability	0.03
Human feedback weight	20%
Coupling Weight	40%
Cohesion Weight	40%

Using the BPMiSearch, 20% of the total weighted fitness value is assigned to the human assessment. Different factors and experiments justify the selection of the weight values. On the one hand, trial-and-error experiments show that small weights (e.g., 5%-15%) have a weak influence on the search direction. To feel the effect of the qualitative assessment, more evaluations should be conducted, but this takes more time and effort and causes human fatigue. On the other hand, experiments show that having a large weight for the qualitative value (e.g., 30% or 40%) gives the participant a powerful influence to steer the trajectory of the search. However, this strong effect results in rapid changes in the search direction and produces more subjective solutions. To assign a high weight value to a participant it is recommended that the experiment be controlled using strict conditions. These conditions include recruiting eligible participants who have a high level of experience in SOA service design and a good understanding of the organisation's BPMs. These fundamental conditions should be available in order to give the participant high-control when steering the trajectory of the search. Nevertheless, the majority of participants who were recruited for this experiment have considerable experience in the organisation's BPMs, but not in SOA development. It is concluded that an intermediate level of weight (i.e., 20%) is a good start for this experiment.

Coupling and cohesion metrics are assigned 40% each as it is anticipated that equal weights would create balanced results. In addition, equal weights have been assigned to these metrics in previous experiments (see Chapter 5), therefore, having equal weights is required to validate this experiment. Using static weights for human value, coupling, and cohesion at this stage is the preferred choice as part of the trial-and-error parameter tuning experiments. However, utilising self-adaptive weights based on the experience of participants could be useful in the future.

With regard to the selection value, a tournament selection is adopted with the value of seven individuals to be selected randomly from the population. As discussed in section 6.2.1, the evaluation value is reflected in the selected samples for the tournament selection. The feedback on the presentation individual is assigned to the seven individuals based on similar traits. It is found that the effect of reflecting the evaluation value on the sample of seven individuals has an effective level of influence on the trajectory of the search. Whereas, by selecting a smaller number of individuals the effect is very small and fades very quickly. Experiments show that a larger number enables the search to converge to the fitness plateau more quickly.

To examine the reliability of the experiment five runs are conducted for each experiment and the average values are then calculated. During each run, a presentation solution is presented to the participant for evaluation after 100 generations. Therefore, the total number of evaluations is 20 times for each participant. Many considerations such as the following are taken into account to calculate the most appropriate number of interactions: the complexity and scale of the case study, the time needed to evaluate each presentation individual, the amount of time the participants can devote to participate in this experiment, and the human fatigue issue. It is hypothesised that 20 evaluation times can be a sufficient number. However, the option of stopping the evaluation after completing fewer evaluations has been excluded to provide equality and uniformity to the results.

Each participant will work individually on a separate machine in the lab. The search engine tool is implemented in Java and the experiments are conducted on a standard desktop PC running the Microsoft Windows operating system.

6.3.4. Results

Each participant produced a SOA solution using the BPMiSearch tool. Domain experts, with the aid of the BPMiSearch tool, have successfully produced a set of SOA solutions. Table 6.4 presents a candidate solution that has been constructed using the interactive method. The first column shows the service number. The second column shows the SSEs that comprise each service, while the third column reveals all the detailed functions that comprise each SSE. This solution contains nine services and each service includes one or more SSEs.

Table 6.4: Candidate service solution produced using the interactive BPMiSearch method

Service ID	Search Space Elements (SSE)	Detailed functions
Service 1	Receptionist	(1) Request patient's file (2) Check files (3) Return patient's file (4) Find patient's appointments
	Medical Records	(5) Register file's details (6) Find patient's file (7) Save patient's file in library (8) Send patient's file (9) Open file (10) Check files (11) Check if there is a new patient
Service 2	Patient	(1) Pay or come to an agreement (2) Visit doctor (3) Book appointment by phone (4) Visit Clinic

	Receptionist (Cancer detection unit)	(5) Inform patient to visit doctor (6) Check if patient medically insured (7) Register Patient details (8) Book appointment (9) Check if the patient is in DB
Service 3	Patient	(1) Pay (2) Visit imaging department
	Imaging department	(3) Inform patient to visit doctor (4) Inform patient to visit doctor (5) Add and report results (6) Perform test (7) Receive payment (8) Receive patient visiting imaging department (9) Check if patient is medically insured (10) Check if patient has appointment (11) Book appointment for patient
Service 4	Inpatient care specialist and nurses	(1) Follow-up patient (resident doctor) (2) Make appointment in outpatient clinic with patient (3) Check if patient needs to remain in hospital (4) Perform treatment according to patient state (5) Check patient's financial state (6) Transfer patient to imaging department (7) Follow up patient state (resident doctor) (8) Check if patient needs imaging investigation (9) Send sample to lab (10) Check if patient needs lab test (11) (specialist) Review resident doctor's orders, diagnoses patients and review old tests
	Accounts clerk	(12) Approve patient's financial state (13) Check patient's financial state
Service 5	Patient	(1) Pay (2) Visit radiotherapy (3) Receive treatment
	Radiotherapy department	(4) Receive payment (5) Transfer patient (6) Ask patient to visit specialist (7) Transfer patient (8) Receive patient visiting radio (9) Check if the patient is medically insured (10) Check if patient has appointment (11) Begin treatment (12) Add results (13) Check if patient needs imaging test (14) Check if patient needs lab tests
Service 6	Medical records clerk	(1) Send reports to managers (2) Generate main statistical report (3) Analyse collected data (4) Collect data from different departments

	Patient	(5) Visit department (6) Visit admission department
	Admission clerk	(7) Check if patient is emergency case (8) Inform the patient to visit department (9) Add patient to waiting list (10) Check room availability
	Inpatient care specialists and nurses	(11) Begin surgery (12) Check if patient needs surgery (13) Transfer patient to chemo department (14) Check if patient needs chemotherapy (15) Check if patient needs imaging tests (16) Transfer patient to lab (17) Check if patient needs tests (18) Continue treatment (19) Receive patient visiting department and his papers (20) Check if patient needs other treatment (21) Open admission file (22) Transfer patient to radiotherapy department (23) Check if patient needs radiotherapy (24) Transfer patient to imaging department (25) Update patient file (26) Add notes to file
	Receptionist (Inpatient care)	(27) Collect patient's files who have been discharged (28) Add collected data to database (29) Collect data (30) Send patient's file
	Receptionist (department-specific)	(31) Add collected data to database (32) Collect data (33) Send reports
	Receptionist (Outpatient clinic)	(34) Send patients' files (35) Add results into database (36) Collect data (37) Send list of patients (38) Collect patient files
	Registrar	(39) Add additional information (40) Check for additional information (41) Add primary tumour (42) Extract main details about cancer patient (43) Make copies of pathology reports and death certificates (44) Add required details in JCR form (45) Generate reports about cancer incidents in the hospital (46) Check if primary tumour exists in database
		(47) Add patient's details to database (48) Check if patient exists in database (49) Check if there is any contradictable data (50) Inform specialist about contradictable data

	Doctor (Diagnostician)	(51) Check if patient needs admission (52) Receive patient visiting doctor (53) Refer patient to special combined clinic (54) Review lab and imaging results (55) Perform clinical appraisal (56) Check if patient needs imaging investigations (57) Book appointment for patient (58) Take notes and review history (59) Check if patient needs investigations (60) Order test (61) Update patient's file
	Patient	(62) Visit clinic (63) Pay or come to agreement
	Receptionist (outpatient clinic)	(64) Guide patient to combined clinic (65) Check if patient has medical insurance (66) Receive payment (67) Receive patient visiting clinic (68) Check patient's appointment
	Patient	(69) Visit combined clinic
	Combined clinic (specialists)	(70) Check if patient needs tests (71) Order test (72) Check if the patient needs admission
		(73) Inform patient to visit radio (74) Receive patient visiting combined clinic (75) Book appointment for radiotherapy (76) Inform patient to visit chemo (77) Book appointment imaging department (78) Continue treatment (79) Review patient's history and all investigations (80) Check if patient needs other treatment (81) Check if patient needs chemo (82) Request admission (83) Check if patient needs radio (84) Book appointment for chemotherapy (85) Devise plan for treatment
	Patient	(86) Receive information to wait (87) visit specialist
	Specialist	(88) Perform suitable treatment according to patient situation (89) Check if patient needs tests (90) Check if the patient needs admission (91) Perform medical appraisal (92) Book appointment imaging department (93) Request another appointment (94) Send advices and instructions to patient
		(95) Check if patient needs another appointment (96) Take notes, review history and old tests (97) Update patient file (98) Request admission from admission clerk (99) Order test
Service 7	Patient	(1) Visit chemotherapy

		(2) Receive treatment
	Chemotherapy department	(3) Perform treatment (4) Receive payment (5) Receive patient visiting chemo (6) Ask patient to visit specialist (7) Add results (8) Check if the patient is medically insured (9) Check if patient has appointment
Service 8	Patient	(1) Handle payment (2) Visit lab
	Lab	(3) Receive patient visiting lab (4) Inform patient to visit doctor (5) Perform test (6) Receive payment (7) Check if patient medically insured (8) Add results
Service 9	Receptionist (outpatient department)	(1) Send the list to registrar (2) Make list of patients who have not attended their appointments
	Registrar	(3) Inform patient's specialist to update file (4) Inform patient to visit specialist (5) Check if patient changed hospital (6) Contact patient (7) Update patient's file (8) Find patient's address

Figure 6.4 presents a partial sample WSDL file decoded by the service decoder. This XML-based file comprises the interface components based on the standard WSDL specifications (Farrell et al., 2006 W3schools, 2015). The tags include the documentation, the types, the operations, and the input and output elements. The name (i.e., ID) of each element in the WSDL file (e.g., operations, parameters, and messages) are presented in a generic form that follows a standard naming convention. In this naming convention, each element ID comprises two parts: the type name and the sequence number. These two fields are concatenated to build a name such as, for example, UserTask_9, or SendTask_2. The unique ID of each element used in the BPMN models is not necessarily a readable name. Thus, the components and elements of the WSDL are prepared so as to be used in the source code rather than being used as simple plain language.

If the service is not a stand-alone, i.e., it has one or more connections with other services, then a binding section will include ports to connect to the targeted services. In addition, the ports that allow other services to connect to the service are included in the binding section. Figure 6.5 highlights the binding section in a sample WSDL file.


WSDL 2

```
<?xml version="1.0"?>
<wsdl:description xmlns:wsdl="http://www.w3.org/ns/wsdl"
 xmlns:wsoap= "http://www.w3.org/ns/wsdl/soap"
 xmlns:hy="http://www.herongyang.com/Service/"
 targetNamespace="http://www.herongyang.com/Service/">

 <wsdl:documentation>
  KHCC_WSDL_20_SOAP.wsdl
  UWE, SERG Group </wsdl:documentation>  <wsdl:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.uwe.ac.uk/Service/">
  <xsd:element name="ManualTask_20" type="xsd:string"/>
  <xsd:element name="ManualTask_19" type="xsd:string"/>
  <xsd:element name="ManualTask_21" type="xsd:string"/>
  <xsd:element name="SendTask_9" type="xsd:string"/>
  <xsd:element name="SendTask_9Response" type="xsd:string"/>
  <xsd:element name="ManualTask_23" type="xsd:string"/>
  <xsd:element name="ManualTask_22" type="xsd:string"/>
  <xsd:element name="UserTask_15" type="xsd:string"/>
  <xsd:element name="UserTask_16" type="xsd:string"/>
</xsd:schema>
</wsdl:types>  <wsdl:interface name="service 2Interface" >
  <wsdl:operation name="Process_8"
  <wsdl:operation name="Process_9"
   pattern="http://www.w3.org/ns/wsdl/in-out"
   style="http://www.w3.org/ns/wsdl/style/iri">
   <wsdl:input messageLabel="In"
    element="hy:MyElement" />
   <wsdl:output messageLabel="Out"
    element="hy:service 2Response" />
   </wsdl:operation>
  </wsdl:interface></wsdl:description>
```

Close

Figure 6.4: WSDL file for the header of a sample service

```xml
<wsdl:operation name="pdfMaker">
  <soap12:operation soapAction="http://tempuri.org/pdfMaker" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ESB">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    This webService is just an example used to show how to invoke WebServices from BizAgi.
  </wsdl:documentation>
  <wsdl:port name="ESBSoap" binding="tns:ESBSoap">
    <soap:address location="http://www.bizagi.com/ESB/ESBServices.asmx"/>
  </wsdl:port>
  <wsdl:port name="ESBSoap12" binding="tns:ESBSoap12">
    <soap12:address location="http://www.bizagi.com/ESB/ESBServices.asmx"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Figure 6.5: WSDL binding sample

Basically, this experiment focuses on two factors; effectiveness, which means achieving the highest possible fitness value (i.e., maximisation problem) and efficiency, which means to achieve the needed fitness value in a shorter time (i.e., minimisation problem). Finding an effective solution that takes a long time is not very useful because it is important for a responsive search to be collaborative between the participant and the intelligent search algorithm. On the other hand, if the search is fast but cannot produce solutions with good-enough fitness values, then the search technique become useless.

Figure 6.6 shows the average population fitness curves for two sample solutions; one is produced by the BPMiSearch (i.e., calculated using equation 6.2) and the other is produced using the non-interactive search (i.e., calculated using equation 4.3). The graph shows the evolution of the population with respect to fitness values against the generations. The fitness curve for the non-interactive search starts faster than the BPMiSearch curve, but, after 100 generations the BPMiSearch curve jumps up to the same level. Then slightly, the BPMiSearch fitness curve starts rising above that level, which makes BPMiSearch superior after a few generations. Since the interactive framework quantifies the subjective feedback of participants, the first step is to evolve solutions without human interaction. Next, these solutions will be presented to the participant for assessment. The feedback values will be integrated with the quantitative fitness values. The weight of the human feedback value is tuned to be 20%, and the first interaction will be after 100 generations. Hence, the graph should be drafting a maximum of 80% of the fitness value for the first 100 generations. Once the human feedback is integrated with fitness value, a steep curve can be noticed. The BPMiSearch curve jumps to a higher point after 100 generations and this happens as the first evaluation assessment takes place after the first 100 generations. The value before that represents only 80% of the fitness value (i.e., values of coupling and cohesion only). Adding human feedback to the quantitative fitness values shows the actual fitness values of the interactive solutions (after 100 generations).

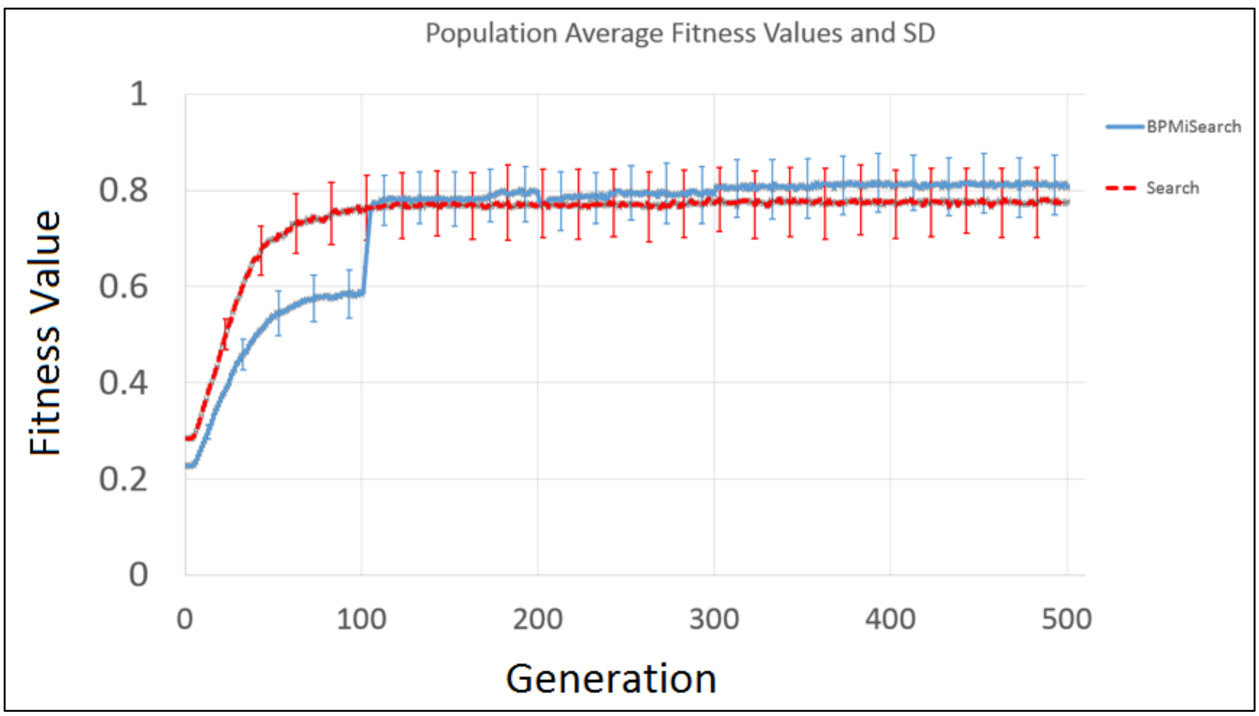

Figure 6.6: Population average fitness curves with standard deviation for a BPMiSearch vs. search sample

As each of the seven participants produced a different solution using the BPMiSearch tool, fitness values when arriving at the fitness plateau are observed.

Table 6.5: Population average fitness value at fitness plateau using BPMiSearch

Solution	Fitness (SD)	Generations (SD)
Solution 1	0.8047 (0.0571)	300 (67.89)
Solution 2	0.7849 (0.0503)	213 (62.95)
Solution 3	0.7961 (0.0532)	301 (68.13)
Solution 4	0.8105 (0.0546)	302 (67.05)
Solution 5	0.8134 (0.0545)	301 (67.97)
Solution 6	0.8138 (0.0586)	196 (62.96)
Solution 7	0.7877 (0.0512)	302(68.09)
Average	**0.802**	**274**

Table 6.5 presents the population average fitness values when arriving at a fitness plateau (i.e., the higher the fitness the more effective solution) and also the number of generations needed to arrive at that fitness plateau (i.e., the shorter the time to arrive at good solutions the more efficient). Each value represents the average after conducting five runs and each row represents a trial by a different participant. The standard deviations are shown in parenthesis. Overall, the average fitness value when arriving at a fitness plateau is 0.8016 and requires approximately 274 generations to arrive at that value.

Table 6.6 presents the resulting outcomes when using the non-interactive search algorithm with the same genetic parameters when arriving at a fitness plateau. These results are presented in chapter five and they are used here to compare the two methods. It is noticed that the population average fitness value when arriving at a fitness plateau is 0.7605. Arriving at a fitness plateau requires approximately 157 generations on average.

Table 6.6: Populating average fitness values at the fitness plateau using the non-interactive search

Solution	Fitness (Std. Dev)	Generations
Solution 1	0.7605 (0.067)	94 (26.07)
Solution 2	0.75 (0.0681)	98 (25.16)
Solution 3	0.7621 (0.0672)	158 (48.59)
Solution 4	0.7581 (0.0682)	140 (21.12)
Solution 5	0.7577 (0.0689)	237 (65.95)
Solution 6	0.77 (0.0672)	209 (59.81)
Solution 7	0.7654 (0.0679)	166 (45.89)
Average	**0.7605**	**157**

Table 6.7 shows the population average fitness values at the fitness plateau for the interactive BPMiSearch (i.e., with the human-in-the-loop) and the non-interactive search methods. It is observed that higher population average fitness values have been obtained using the interactive BPMiSearch compared to the search.

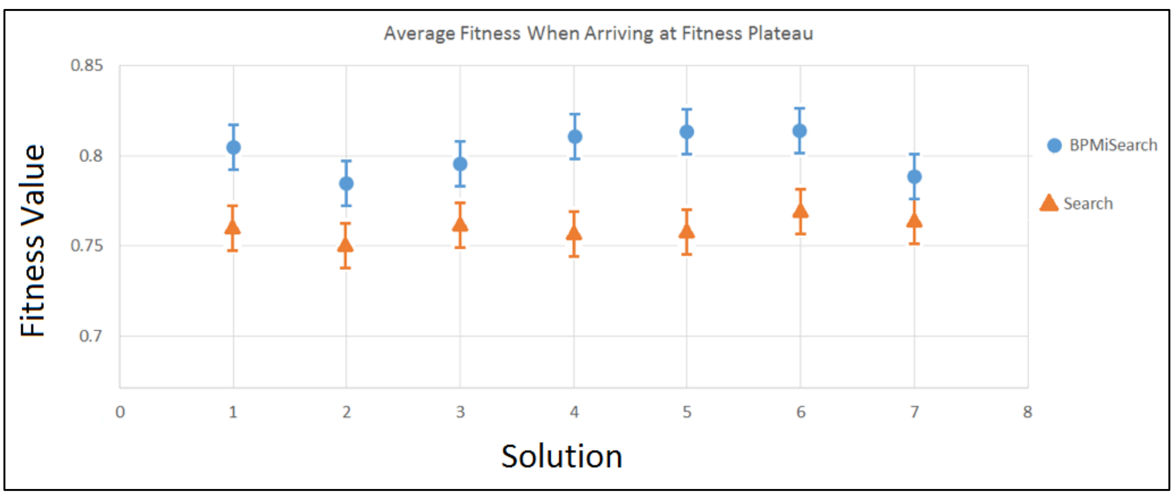

Figure 6.7: Population average fitness values and standard deviation when arriving at a fitness plateau

Figure 6.8 shows the average number of generations needed to arrive at a fitness plateau. Each point represents the average of the five runs of the participant interactive experiment. The majority of the search samples have lower values compared to the BPMiSearch samples. Arriving faster at good-enough solutions is considered as a minimisation problem (i.e., lower values indicate faster arrival at a fitness plateau), the search algorithm arrives at the fitness plateau faster than the BPMiSearch.

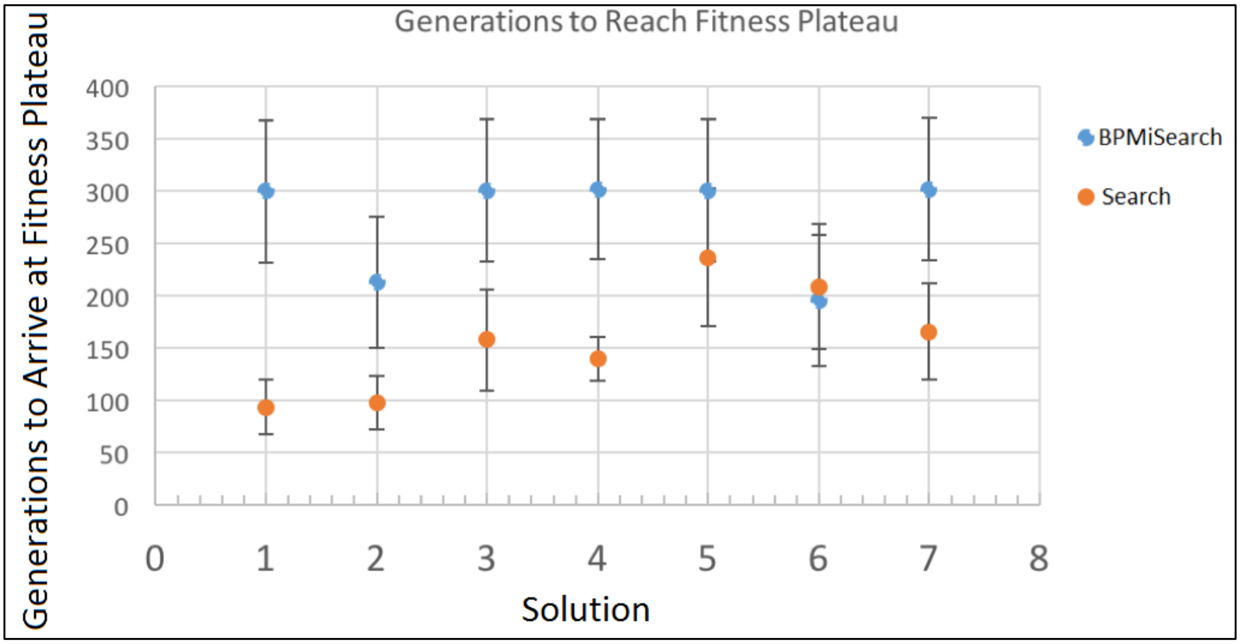

Figure 6.8: Average required generations to arrive at a fitness plateau with the standard deviation

6.3.5. Analysis

Following a discussion with domain experts after reviewing the results, it was observed that the BPMiSearch has supported many of their decisions. For example, in Service 1 presented in Table 6.4, locating functionalities for the 'Receptionist' and 'Medical Records' in one service makes sense because the receptionist uses the medical records to retrieve patient's details and book their appointments. In another example, Service 6 comprises a large set of SSEs. Two factors justify the decision of grouping these SSEs together; (i) most of the functionalities comprising the SSEs are manual, and (ii) grouping these SSEs together generates a core service of cancer care that gathers all the fundamental functions, which is important for any system, while other services add extra features to the core system.

In order to compare the means of the two populations produced by two independent methods (i.e., the search and BPMiSearch), a valid test for the distribution of the data should be conducted to confirm that the sample data have been drawn from a normally distributed population. Due to the small sample size, a Shapiro-Wilk test for normality is performed at an alpha level of 0.05. The null and alternative hypotheses for this test are formulated as follows:

H_0: the population of data samples is normally-distributed.

H_1: the population of data samples is not normally-distributed

If the p-value is less than or equal to the chosen alpha level (i.e., 0.05), then the null hypothesis is rejected. Failing the normality test indicates that the confidence interval is 95% which indicates that that the data are not normally distributed.

The results of applying the Shapiro-Wilk normality test are presented in Table 6.7. For both the search and BPMiSearch data samples, the p-values (denoted by Sig. column), are greater than the alpha level. In addition, the skewness and kurtosis are within acceptable limits of ±2 (Trochim and Donnelly, 2006; Field, 2000; Field, 2009; Gravetter and Wallnau, 2014). Thus, the null hypothesis is accepted, which implies that the data are normally-distributed and that indicates the possibility of performing a parametric test such as the t-test.

Table 6.7: Test of normality of fitness values of the Search and BPMiSearch methods

	Descriptives				**Shapiro-Wilk**		
Method	**Mean**	**SD**	**Skewness**	**Kurtosis**	**Statistic**	**df**	**Sig.**
$BPMiSearch_{Fitness}$	0.8016	0.0121	-0.431	-1.842	0.881	7	0.23
$Search_{Fitness}$	0.7605	0.0063	-0.225	0.697	0.980	7	0.961

The t-test is used to analyse the outcome data by testing the differences between the two groups (i.e., the search and BPMiSearch). The t-test indicates how likely the results of the search methods would be obtained by chance. Statistically significant values denote that there is approximately less than a 5% chance of obtaining the observed results by chance. In obtaining this level of significance, the null hypothesis (i.e., that states that there is no significant change between the two methods) can be rejected. The statistical analysis will reveal any significant improvements in the fitness values.

Table 6.8 shows the results of conducting an independent samples t-test to compare the fitness values of the BPMiSearch and search.

Table 6.8: Independent samples t-test to compare BPMiSearch to the Search population average fitness

Method	**Group Statistics**			**Independent Samples Test**				
	Mean	**Std. Deviation**	**Std. Error Mean**	**t-value**	**df**	**Sig (2-tailed)**	**Cohen's d**	**Effect-size r**
$BPMiSearch_{Fitness}$	0.802	0.012	0.005	7.948	12	0.000004 (< **0.05**)	4.322	0.908
$Search_{Fitness}$	0.761	0.006	0.002					

The table presents the group statistics such as the mean, standard deviation, and standard error mean. In addition, the independent samples t-test presents values that include the t-value, degree of freedom and significance level (i.e., p-value) and, the Cohen's d value is presented to show the effect size (i.e., the strength of the improvement).

Based on the test results, the BPMiSearch samples group (N = 7) is associated with a fitness value M = 0.802 (SD = 0.012). By comparison, the search samples group (N = 7) is associated with a numerically less fitness value M= 0.761 (SD = 0.006). To test the hypothesis that the BPMiSearch samples, in comparison with search, are associated with statistically significant different mean fitness values, an independent samples t-test is performed. The independent samples t-test is associated with a statistically significant effect in comparison with the two methods; the fitness $t(12) = 7.948$, $p < 0.05$. Thus, the BPMiSearch samples are associated with a statistically significant larger mean fitness value than the search samples. Cohen's d is estimated at 4.322 with effect-size r estimated at 0.908, which is a large effect based on Cohen's (1992) guidelines.

As this is a "maximisation problem", the results suggest that the BPMiSearch method performs better than the search method when the population converges to a fitness plateau. The results indicate that the engagement of an expert drives the search into finding more effective solutions. Therefore, it is concluded that the BPMiSearch method has significantly outperformed the search method in terms of effectiveness.

With regard to efficiency, the number of generations needed to reach a fitness plateau has been examined. Table 6.9 shows the results of a normality test for the number of generations required to arrive at a fitness plateau. The Shapiro-Wilk test shows a significance level less than 5% for the BPMiSearch sample which means that the data are not normally distributed. Therefore, the t-test is not a valid test to be used in this case. The alternative is to use a non-parametric test such as the Mann-Whitney U test (Field, 2000).

Table 6.9: Test of normality for the generations needed to converge a fitness plateau

Method	Descriptives				Shapiro-Wilk		
	Mean	**SD**	**Skewness**	**Kurtosis**	**Statistic**	**df**	**Sig.**
$BPMiSearch_{Generations}$	274	47.44	-1.274	-0.556	0.643	7	0.001
$Search_{Generations}$	157	53.11	0.267	-0.947	0.944	7	0.677

The $BPMiSearch_{Generations}$ samples (N=7) are associated with value M=274 (SD=47.44). By comparison, the $Search_{Generations}$ samples (N=7) are associated with a numerically fewer number of generations M=157 (SD=53.11). To test the hypothesis that the $BPMiSearch_{Generations}$ in comparison with $Search_{Generations}$, samples are associated with statistically significant different generations' numbers, a Mann-Whitney test is performed. Table 6.10 shows the results of performing the Mann-Whitney test.

In using the two-tailed Mann-Whitney test, the two methods, $BPMiSearch_{Generations}$ and $Search_{Generations}$, are associated with a statistically significant effect; the Mann-Whitney U=3, n1=n2=7, p =0.006 (<0.05). Thus, the $BPMiSearch_{Generations}$ samples are associated with a statistically significant larger mean value than the $Search_{Generations}$ samples. These results suggest that the search method arrives more quickly to the fitness plateau more quickly compared to the $BPMiSearch_{Generations}$, and thus the search is more efficient. As finding good solutions in a shorter time is a minimisation problem, the results suggest that the search method outperforms the BPMiSearch method in terms of efficiency.

Table 6.10: Generations to arrive at fitness plateau

Method	Group Statistics			Mann-Whitney Test	
	Mean	Std. Deviation	Std. Error Mean	U-value	Sig. (2-tailed)
$BPMiSearch_{Generations}$	274	47.44	17.93	3.00	0.006
$Search_{Generations}$	157	53.11	20.074		

Nevertheless, the BPMiSearch algorithm is slightly slower than the search to converge a fitness plateau. However, the unexpected behaviour of the participants is the main cause of the delay. The flexible behaviour of people that could change at any time plays a significant role in steering the direction of the search. This changeable behaviour of people that stems from the implicit experience and knowledge of the participants controls the evaluation of the resulting services and causes a delay before emerging to a fitness plateau.

This experiment reveals that the interactive preference has an enormous impact on the search outcomes. It is concluded that the combination of the interactive tool with a knowledgeable domain expert may result in producing more effective services of higher quality in comparison with the search outcomes. Nevertheless, the quantitative search method is still more efficient in arriving at a fitness plateau.

As the participants were invited to share their remarks about the interactive experiment, the following key points were raised:

- Generally, the interactive search experience is a successful practice that produced feasible service solutions of high quality and high-level of satisfaction.
- The technique of deriving the feedback is easy and flexible. Compared to the experiment in chapter five, using the medium-size Likert scale (i.e., 5-number scale) is easier and more flexible than the long-size scale (i.e., 10-number scale).
- The visualisation tool that shows the connections between services is very useful and helps the participants to build a strong understanding of the entire solution in a short time. There is a clear consensus that this tool is a great addition that enables the participants to understand the presentation solutions and thus produce more reliable outcomes. In addition, this tool shows the granularity level of the candidate services using different colours. These colours are useful to identify the core candidate services as well (i.e., core candidate services have a red colour, and a solution can have more than one core service).
- All the participants are in agreement about the granularity level of the search space elements. Encapsulating the detailed activities while maintaining the chronological sequence between them is very useful and helpful for the evaluation. Presenting the abstract elements rather than the detailed functions provides a wider view on the business process which leads to more accurate and reliable feedback.
- A self-adaptive weight based on the level of experience of the participants is a suggested feature that may provide participants with additional power to steer the search direction. However, it is agreed that some constraints must control the addition of such a feature. These

restrictions include the availability of a high-level of understanding of the business process as well as a good level of experience in SOA development.

- With regard to human fatigue, there is a learning curve at the beginning which requires considerable effort from the participants. However, understanding the basic elements and the feedback mechanism reduces the participant's effort in later evaluations. In addition, having a time limit for the experiment and finishing on time is a very useful strategy for reducing the anticipated human fatigue.
- Another suggested feature is to give the participant the capability to manually move an element from one service to another one within the same individual solution. This may help to customise the final solution. However, this feature should be managed carefully as the goal of the experiment is to examine the interactive search (i.e., priori and posteriori preference are out of the scope of this research). Thus, this feature can be useful if it is achieved after completing the interactive search session.

6.4. DSRM Evaluation Phase

To evaluate the results of this phase, reference will be made to SBSE empirical evaluation methods (Wohlin et al., 2012; Lazar et al., 2010). Two methods are used to evaluate the BPMiSearch: level of satisfaction and benchmarking. Assessing the level of satisfaction is beneficial in the context of an interactive search because of the existence of qualitative value (i.e., the participant's feedback). Thus, evaluating the outcomes should take place by quantifying the expert evaluation. On the other hand, the comparison between BPMiSearch and BPAOntoSOA continues but uses different aspects.

6.4.1. Level of Satisfaction

Showing the level of satisfaction of the experts is a highly important factor which needs to be measured when developing an IT system. It is used to validate the resulting solution and also to ensure that it adheres to the SOA principles. In the previous chapter, the domain experts evaluated a set of solutions produced using the search. The average evaluation is estimated at 79.9%. Although results show a good level of satisfaction (i.e., based on the Likert scale), the experts emphasised that better results can be produced. The level of satisfaction test in this section is anticipated to reveal the degree to which their level of satisfaction has been influenced by the interactive experiment.

The SOA solutions produced using both the BPMiSearch and search are mixed in a pool and presented to the experts for evaluation. Five domain experts were recruited to participate in the evaluation test by providing their feedback on the presentation solutions. The solutions are presented one by one to participants for evaluation such that the participant does not know the creator of the presentation solution (i.e., search or BPMiSearch). This constraint is important to guarantee that the participant's judgement is not influenced by the source of the solution. This reduces biases. The evaluation tool presented in Figure 6.1 chooses a random solution and publishes it in the main window for evaluation. It also gives the participant the capability to evaluate the services one by one using the slider bar, or the whole solution using the main slider bar at the bottom of the window. A Likert scale is used for the goal as presented in Table 6.1.

The evaluation pool contains 14 solutions (i.e., 7 BPMiSearch samples, and 7 search samples). The participants are invited to evaluate as many of these solutions within the limited time available. This experiment adopts a within-group design, which is more suitable when a small group of resources is available (i.e., experts with top managerial positions at KHCC). In addition, this design isolates the effect of individual differences and provides powerful tests (Lazar et al., 2010). To minimise the bias caused by human fatigue, the time of the experiment is set to a maximum of one hour; the experiment will be stopped even though some solutions are not evaluated. The evaluation tool is implemented in Java and installed on standard PC machines that use Windows OS.

Table 6.11 presents the results of the evaluation. These results show the level of satisfaction about the outcomes of the BPMiSearch and search methods. The participants have evaluated a total of nine solutions produced using the BPMiSearch, and twelve solutions produced using the search. The table shows the method, the solution number, and the evaluation value.

Table 6.11: Level of satisfaction scores

Method	Solution	Evaluation	Method	Solution	Evaluation
BPMiSearch	1	0.7556	Search	1	0.85
	2	0.775		2	0.7778
	3	0.9636		3	0.78
	4	0.9333		4	0.7556
	5	0.8667		5	0.8
	6	0.9333		6	0.68
	7	0.8667		7	0.7667
	8	0.9		8	0.9111
	9	0.9		9	0.8
	-	-		10	0.76
	-	-		11	0.8667
	-	-		12	0.7667
	Average	**0.877**		**Average**	**0.793**

The highest evaluation for BPMiSearch solutions is 0.9333, while the average evaluation for all the BPMiSearch solutions is 0.877. In contrast, the highest evaluation for search solutions is 0.9111 while the average evaluation is 0.793. In both cases, the results show that participants are satisfied with the results based on the Likert scale presented in Table 6.1.

Figure 6.9 shows the distribution of evaluation values on the BPMiSearch and search samples. It is observed that the majority of the BPMiSearch samples are superior to the search samples. This observation emphasises the values presented in Table 6.1 that experts are more satisfied with the BPMiSearch outcomes by 0.084. A statistical analysis test is needed to show if the difference is significant.

Figure 6.9: Level of satisfaction about BPMiSearch and search solutions

The assumption of normality is tested using the Shapiro-Wilk test. A review of the results is presented in Table 6.12 for the $\text{BPMiSearch}_{\text{Satisfaction}}$ (Shapiro-Wilk=0.898, df=9, p=0.242). This shows a normal distribution. In addition, the skewness (0.828) and kurtosis (0.304) are within the acceptable ranges. With regard to the $\text{Search}_{\text{Satisfaction}}$ (Shapiro-Wilk=0.945, df=9, p=0.64), skewness (0.301), and kurtosis (1.48) indicate a normal distribution for the data samples as well.

As the statistics suggest that normality is a reasonable assumption, a parametric test (e.g., t-test) can be performed on this sample.

Table 6.12: Normality test results for the level of satisfaction scores

Method	Descriptives				Shapiro-Wilk		
	Mean	SD	Skewness	Kurtosis	Statistic	df	Sig.
$\text{BPMiSearch}_{\text{Satisfaction}}$	0.877	0.071	0.828	0.304	0.898	9	0.242
$\text{Search}_{\text{Satisfaction}}$	0.791	0.064	0.301	1.48	0.945	9	0.64

An independent samples t-test is performed comparing the mean satisfaction scores of BPMiSearch with the search sample solutions. Table 6.13 presents the statistics and t-test results. The mean for the search sample evaluation is estimated at 0.793, which is consistent with the evaluation result conducted in chapter five is estimated at 0.799. This indicates high reliability.

The two-tailed independent samples t-test shows that the $BPMiSearch_{Satisfaction}$ samples (M=0.877, SD=0.071) are more agreeable than the $Search_{Satisfaction}$ samples (M=0.793, SD=0.0601), t(16)=2.872, p=0.011 (<0.05). The Cohen's d is estimated at 1.277 with an effect size r=0.538; this is a large effect according to Cohen's (1992) guidelines. Thus, it is confirmed with a confidence level of 95% that a significant improvement in the level of satisfaction has been obtained using the BPMiSearch.

Table 6.13: Level of satisfaction Statistics

Method	Group Statistics			Independent Samples Test				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	effect-size r
$BPMiSearch_{Satisfaction}$	0.877	0.071	0.024	2.872	16	0.011	1.277	0.538
$Search_{Satisfaction}$	0.793	0.0601	0.017					

Results show that the interactive experiment has improved the level of satisfaction, and developed the acceptance of the produced solutions. The interactive feedback has a high impact on the results as it reflects the knowledge and experience of the participants on the resulting solutions. The reason is that the subjective feedback values helped the search to produce candidate solutions that are closer to human expectations (Ramirez et al., 2018). This experiment provides a positive contribution to the field of interactive search by highlighting the impact of an interactive method on the service identification problem. It is concluded that the use of the BPMiSearch helps to enrich the quality, effectiveness and the acceptance rate of the resulting candidate SOA solutions.

6.4.2. Benchmarking

In the previous two chapters, a comparison between the interactive BPMiSearch framework and BPAOntoSOA framework has been conducted. The similarities between the two approaches stimulate the comparison as both are top-down approaches that aim to derive candidate software services from the business process. In the previous iterations, different aspects have been discussed such as the abstraction level of input data, automation, and the type of the resulting candidate service. The comparison continues in this iteration using different aspects. As the interactive context has been presented in this chapter, the comparison focuses on two aspects; (i) the utilisation of the interactive human preference to improve the quality of the resulting services, and (ii) satisfaction level of experts about the resulting services. Table 6.14 highlights these differences with reference to human preference.

Table 6.14: the BPMiSearch SIM vs. BPAOntoSOA

Perspective	BPMiSearch	BPAOntoSOA
Quality assessment	The interactive search uses the human preference to feed into the search process and steer the trajectory of the search.	A static algorithm performs the clustering with no human interaction.
Satisfaction level of domain experts	Core service connected to a set of smaller services.	No core service, functions are distributed on all services.

The BPAOntoSOA uses a static algorithm (i.e., RPA clustering) that extracts the required elements from the Riva-based business process architecture (BPA) and then groups them into candidate services based on the relationships between these elements. The human involvement is limited with regard to following the guidelines about how to prepare the data to be used by the algorithm (but not to interact with the algorithm itself) and consequently, the participant does not make any decisions. However, evaluating the resulting candidate services is subjective as the expert examines the resulting services at the end based on implicit knowledge, but this evaluation has no effect on the results. Human preference does not play any role in shaping or evaluating the candidate services in BPAOntoSOA. In contrast, in the BPMiSearch, the role of the user is important to steer the trajectory of the search by providing the appropriate evaluation. This evaluation helps the search algorithm to choose the strongest individuals to live and evolve.

With regard to statistical analysis, the fitness values of the candidate services using BPAOntoSOA are calculated. Table 6.15 shows the statistics of BPAOntoSOA using the fitness function adopted by the search method. Note that the fitness value can be calculated using the formula presented in Equation 6.2. The coupling and cohesion values are calculated based on counting the elements inside each service as well as the number of relationships between the services or within each service.

$$Fitness(x) = (Weight_{coupling} * \left(1 - CpF(x)\right) + (Weight_{cohesion} * ChS(x)) \quad (6.2)$$

Table 6.15: Fitness statistics for BPAOntoSOA

Service	Coupling	Cohesion	Fitness
1	0.5	2	0.75
2	0.67	4	0.5
3	0.6	2	0.7
4	0	2	1
5	0.33	2	0.835
6	0.88	1	0.31
7	0.67	1	0.415
8	0.5	0	0.25
9	0	0	0.5
10	0	1	0.75
11	0	2	1
12	0.33	1	0.585
13	0.5	0	0.75
14	0	2	1
15	0.57	0	0.215
16	0	0	0.5
17	0	2	1
18	0.5	0	0.75
Average	**0.336**	**0.648**	**0.656**

Conducting seven runs of BPAOntoSOA, the RPA clustering algorithm has produced the same fitness values, therefore, the mean fitness value will remain the same during all the runs. Table 6.16 presents the results of conducting an independent samples t-test to compare the BPMiSearch results presented in Table 6.5 with the BPAOntoSOA.

Table 6.16: Independent samples t-test to compare BPMiSearch samples with BPAOntoSOA samples

Method	Group Statistics			Independent Samples Test				
	Mean	Std. Deviation	Std. Error Mean	t-value	df	Sig (2-tailed)	Cohen's d	Effect-size r
BPMiSearch	0.802	0.012	0.005	31.82	12	0.000 (< **0.05**)	17.06	0.993
BPAOntoSOA	0.656	0.000	0.000					

Based on the test results, the BPMiSearch samples group (N = 7) is associated with a fitness value M = 0.802 (SD = 0.012). By comparison, the BPAOntoSOA samples group (N = 7) is associated with a numerically less fitness value M= 0.656 (SD = 0.00). To test the hypothesis that the BPMiSearch in comparison with the BPAOntoSOA samples are associated with statistically significantly different mean fitness values, an independent samples t-test is performed. The independent samples t-test is associated with a statistically significant effect in comparison with the two methods; the fitness $t(12) = 31.82$, $p < 0.05$. Thus, the BPMiSearch samples are associated with a statistically significant larger mean fitness value than the BPAOntoSOA samples. Cohen's d is estimated at 17.06 with effect-size r estimated at 0.993; this is a large effect based on Cohen's (1992) guidelines.

With reference to the second aspect of the comparison, i.e. the level of domain experts' satisfaction, the resulting services of BPAOntoSOA have been compared to the outcomes of BPMiSearch by a domain expert. Comparing the two solutions side by side, experts observed that some of the services produced by the two frameworks are identical, i.e. have the same set of activities. However, the expert observed that the major difference between the outcomes of the two frameworks is the granularity level of the resulting candidate services. It is observed that BPMiSearch produces a core service that allocates the key functionalities in one candidate service, i.e., core service, and distributes other functionalities on small services that can have connections with the core service or even perform as stand-alone services. In contrast, the BPAOntoSOA framework distributes the functionalities of all candidate services without creating a specific core service.

Figure 6.10 depicts the difference between the two frameworks. On the left-hand side, the BPMiSearch a set of SSEs are allocated inside the core service, and a set of smaller services support the core service. BPAOntoSOA solution on the right-hand side of the figure, functions are distributed on different candidate services with no core service. The figure highlights three types of elements; (i) a core service that performs multiple functionalities through a set of SSEs, (ii) a core SSE is allocated inside a core service, and (iii) a supporter service that usually has one clear purpose and is located outside core services.

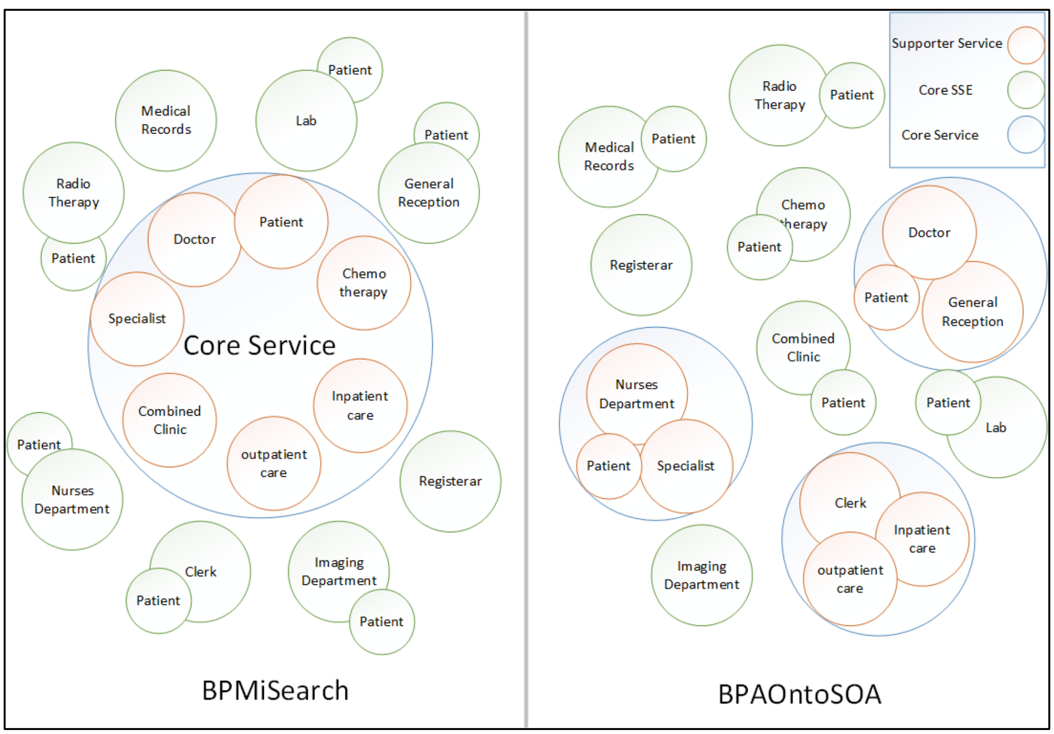

Figure 6.10: Granularity level of resulting candidate services using BPMiSearch and BPAOntoSOA

Although the domain expert prefers a solution that has a core service, along with a set of support services as this structure reflects the reality of the real workflow, the expert conflicts with some SSEs that are allocated inside/outside the core service. The expert explains that BPMiSearch allocates some SSEs inside the core service, while other SSEs that have the same level of importance are allocated outside the core service. For example, in a solution produced using BPMiSearch, ‘Radio Therapy’ SSE is allocated inside the core service, whereas ‘Chemo Therapy’ SSE is formulated as a stand-alone service. Experts see that chemotherapy and radiotherapy should be treated in a similar way. In BPAOntoSOA, radiotherapy and chemotherapy have been identified within two services at the same level. Appendix G presents two solutions, one is produced using BPAOntoSOA, and another one that is produced using BPMiSearch. The appendix shows the common services between the two solutions, and then the special services for each solution.

In light of this observation, and to investigate these results in more detail, the seven solutions produced using the interactive BPMiSearch framework, i.e. with the participation of domain experts, have been analysed. It is observed that these candidate service solutions produce a core service with the support of a set of stand-alone services. The allocation of SSEs differs from a solution to another such that the core service and support services do not have the same elements in different solutions. For example, in one solution, radiotherapy and chemotherapy are both allocated in the core service, whereas, in another solution, both of them are allocated in two stand-alone services. This difference refers mainly to (i) the relationships between the SSEs (i.e., SSEs with high connectivity are more likely to be allocated in the core), and (ii) the feedback from the domain expert who interacts with the search tool to construct these solutions. Human preference can lead to the allocation of a specific SSE inside or outside the core candidate service.

Tracing back to the original BPMN models and the relationships between the core functionalities in core service and other services, it is found that SSEs inside the core service have a high level of connectivity. These SSEs are connected strongly with each other so they act as one big service. Therefore, SSEs that participate with more activities, e.g. general reception, are anticipated to be in the core service in any solution, whereas, SSEs that have low connections with the core are anticipated to be identified in stand-alone services. It is concluded that, although the relationships between SSEs plays the most important role in creating a core service, the interactive preference of participants helps to identify the SSEs that will be allocated in the core service. In BPAOntoSOA, all candidate services have approximately the same level of granularity. The interactive BPMiSearch framework resolves this problem by (i) allowing the expert to steer the trajectory of the search, which allows the production of solutions that match the expert's preference, and (ii) provides the capability to produce multiple solutions, so a participant can select a better solution from a set of candidate feasible solutions.

Overall, the BPMiSearch method has outperformed the BPAOntoSOA in terms of effectiveness, i.e. better fitness values, and with regard to the successful utilisation of human preference to enrich the quality of outcomes. BPAOntoSOA outperforms the BPMiSearch by assigning the same level of granularity of services, which can satisfy domain experts. However, BPMiSearch has the capability of interactively producing multiple feasible solutions, which are subject to the domain expert's satisfaction.

6.5. Threats to Validity

There is no doubt that the experimental studies strive to achieve high reliability. An indicator of a reliable experiment is the ability to replicate the same experiment in other locations by other research teams and yield consistent, dependable and stable results (Lazar et al., 2010). The considerable challenge when having the human-in-the-loop and which causes the experiment to be different to others in the so-called "hard" sciences such as physics or chemistry, is that there is no control over human behaviour which makes measurements of the social interaction subject to high fluctuation and inconsistency; this makes them less replicable. The major causes of fluctuations are errors. There are two main types of errors when conducting an experimental study:

1) Random errors: these errors are not correlated with the real value and they push the observed values up and down around the accurate value. There is no control over these random errors and as a consequence there is no way to eliminate them. However, the impact of random errors can be reduced by enlarging the sample size. Random errors may have a significant influence on the observed values when the sample size is small. Random errors normally offset each other when the sample size is large, which leads the observed mean to be very close to the actual correct value (Lazar et al., 2010). Although it is not possible to claim that the impact of random errors has been eliminated by 100%, the large sample size raises confidence that the observed value is more accurate and closer to the actual value. To satisfy this issue, seven domain experts with high level of knowledge and experience at KHCC are recruited to participate in the interactive experiment.

2) Systematic errors: these errors are also called "biases" and have a different nature compared to random errors. In contrast to random errors that affect the observed values in both directions, the biases push the observed values in the same direction. Systematic errors do not offset each other, resulting in too low or too high mean values. To reduce the impact of biases, the cause should be controlled. This includes the biases in the experimental procedures, the bias caused by participants or experimenters, and the bias caused by the environment. In a well-designed experiment, the biases are under control and this improves the data and makes the observed values closer to the actual values. However, biases can never be fully eliminated, but keeping them under control improves the quality of the results. A set of potential biases and mechanisms to avoid them is presented as follows:

    a) The learning effect: if the participants work on the SOA solutions prepared in advance using the search, they will learn from these solutions and try to clone them. To avoid this bias, the participants are invited to create SOA solutions by interacting with the tool before seeing other sample solutions produced by the search. This step is expected to minimise the learning effect.

b) The influence of personal relationships: the evaluation could be affected if the participant knows the creator of the SOA solution (i.e., either a machine or another participant). Thus, the identity of the SOA solution creator is concealed creating a more unbiased evaluation.

c) Human fatigue: if the participants feel tired, the results will not be accurate. Therefore, to overcome this issue, a set of processes should be considered so as to minimise the number of interaction times when doing the interactive search in order to ensure the participants have enough breaks and feel comfortable about the physical environment in the lab.

d) Changes in KHCC business process models: many changes and updates on different parts of the BPMs have occurred since the data collection in 2006. These changes confused some domain experts not familiar with them. In addition, it is possible that some domain experts are not familiar with all the business processes in the organisation. To overcome these limitations, three procedures have been performed: (i) conduct a workshop to explain the major parts of the business process with the support of the senior domain experts, (ii) present abstract functionalities rather than the detailed activities in the business process, enabling the experts to be more confident when making decisions, and (iii) discuss the major changes in the business process with the aid of the senior domain experts to explain the new upgrades; this has helped other experts understand these changes.

6.6. Summary

The experiments in this chapter have investigated the impact of human interactive preference on the search outcomes. The design and implementation of a novel interactive framework (i.e., BPMiSearch) have been discussed in addition to the experiments that would satisfy the goals of this chapter. This BPMiSearch framework has been demonstrated with the collaboration of domain experts using a real-life case study. Statistical analysis results and the participant assessment reveal that the BPMiSearch service identification framework is highly effective (i.e., in terms of coupling, cohesion, and experts' level of satisfaction) and an efficient interactive search engine for the dynamic derivation of SOA candidate services from role-based business process models. The main contributions that are presented in this chapter can be summarised as follows:

(i) The BPMiSearch is an effective and efficient framework to derive candidate software services from role-based BPMs. The qualitative feedback of domain experts steers the direction of the search to find high-quality candidate SOA solutions.

(ii) The implicit knowledge and explicit experience of domain experts play a significant role in the interactive process. Therefore, high experience participants are anticipated to have a higher capability to produce more effective candidate services comparing to participants with less experience in business process and services. This is because the knowledge of business process models is required to allow participants to steer the trajectory of search and provide accurate feedback on the presentation solutions.

(iii) The interactive method has successfully improved the level of satisfaction of the domain experts. This contributes to the importance of a qualitative value to enrich the search outcomes.

(iv) The colourful visualisation of the SOA solutions is extremely accessible and this helps to raise the engagement of the participants in the interactive sessions. As a result, this may lead to producing more reliable solutions.

Results of the experiments show that the BPMiSearch method has outperformed the search (i.e., with no human interaction) technique and also the BPAOntoSOA. The only two exceptions are (i) the search method without human interaction arrives more quickly at a fitness plateau, and (ii) granularity level of candidate services produced by BPAOntoSOA can possibly be more acceptable by domain experts.

Chapter 7
Conclusions

Although a fundamental component of software SOA development, service identification is a difficult, non-trivial, and cognitively demanding task for software engineers to accomplish. Search-Based Software Engineering (SBSE) techniques have been widely utilised to automate development activities, but they raise many challenges such as reflecting the reality of a software engineer's activities with the representation and fitness function. Human involvement with software development can be beneficial in addressing these challenges which can help to evolve the SOA development and improve the quality of outcomes (Ramirez et al., 2018). Hence, interactive search and exploration techniques have been applied with promising results in a wide range of research fields (Simons et al., 2014). This research has investigated the possibility of using an interactive search-based software engineering (iSBSE) technique to derive candidate software services from role-based business process models (BPMs).

The key objectives of this research are to use interactive search to simplify, enhance, and semi-automate the service identification process, such that the services produced can contribute to high-quality SOA solutions. As a first step, a comprehensive layered framework has been developed which fulfils all the service identification activities (i.e., input BPMN preparation, service identification, and candidate service refinement). This novel framework has been validated by domain experts at the King Hussain Cancer Center (KHCC). In the next phase, a search-based method has been used to explore and exploit the search space to perform the service identification activities. The framework successfully arrived at feasible solutions with higher fitness values when compared to manual methods and fully-automated service identification processes. In the third phase, an interactive search-based method has been introduced to extend the previous search-based method. The main objectives of the interactive search method are to find a balanced combination of quantitative fitness values and qualitative feedback of experts. In addition, the interactive search can reduce the gap between the software engineer stakeholder and the optimisation search algorithm, thus helping to achieve the participant's acceptance and satisfaction. It has been found that striking a balance between exploration and exploitation of the search space of the service identification problem can help to find solutions that satisfy the business needs, obtain higher fitness values, and also achieve higher values of expert's level of satisfaction compared to the non-interactive search-based method. In the following sub-sections, section 7.1 presents a summary of the research outcomes that include bottom-up traceability to answer the research questions and satisfy the research hypothesis. Future directions are presented in section7.2.

7.1. Fulfilment of the Research Questions and the Research Hypothesis and Summary of Research Outcomes

This section presents a critical review of the outcomes of this research in order to fulfil the research hypothesis by answering all the research questions. Figure 7.1 provides bottom-up illustration of how the research hypothesis was answered using the findings and outcomes of Chapters 4, 5, and 6. The four research questions are answered while presenting each point of the research outcomes in order to address the research hypothesis by the end of this section. Figure 7.1 shows that RQ1 and RQ2 are principally addressed in Chapter 4. The outcomes of addressing these two questions feed into the next chapter to address RQ3. By satisfying RQ3 in Chapter 5 (non-interactive search framework for service identification), addressing the interactive search framework becomes possible. Therefore, Chapter 6 builds on the outcomes of all the previous chapters to investigate the influence of the interactive context on the outcomes of the search. The following points discuss the key contributions and outcomes of this research, and at the same time discuss how the research questions have been addressed. In light of fulfilling the research questions, the validity of the research hypothesis is presented in the overall section (7.1.6).

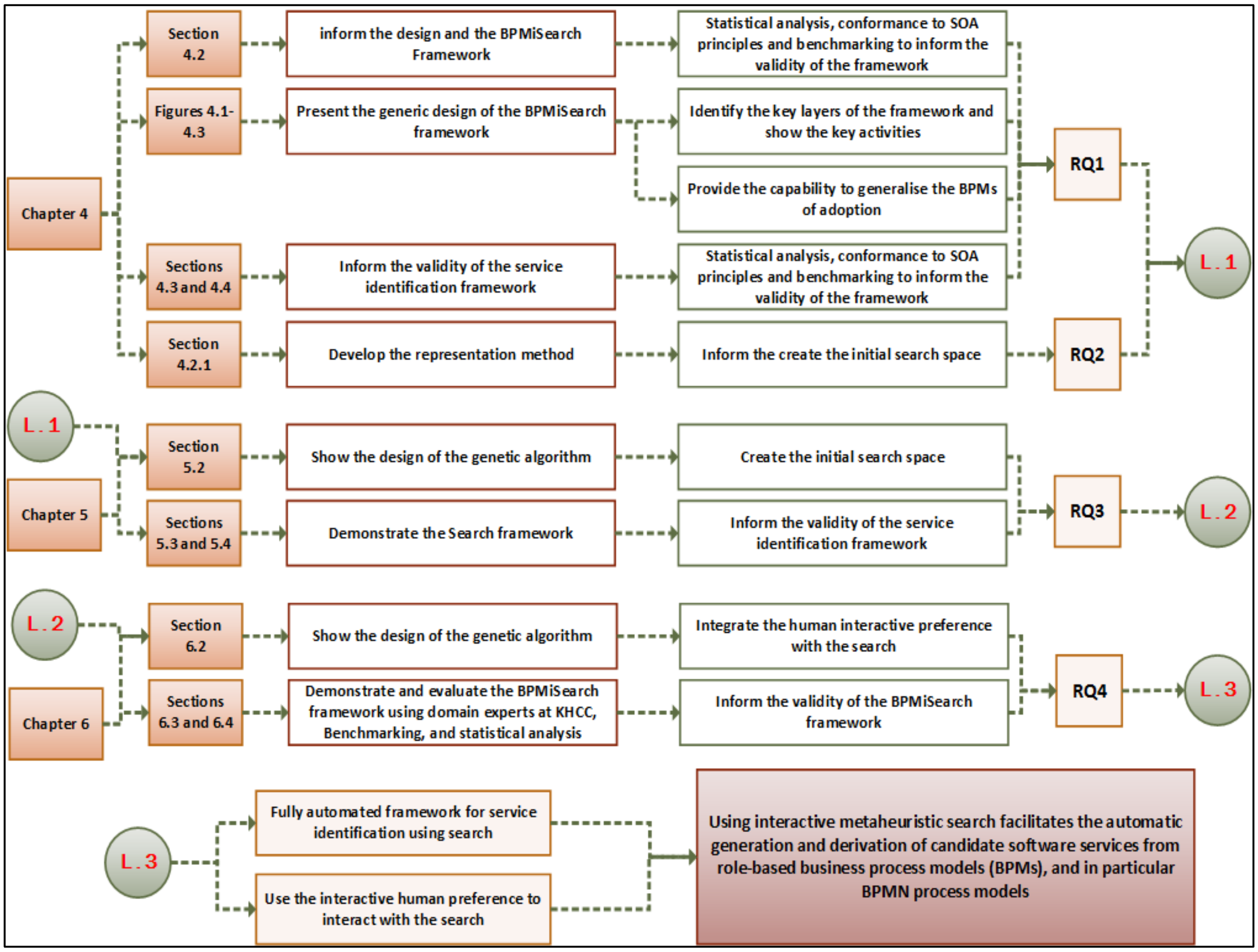

Figure 7.1: Bottom-up answering of research questions and research hypothesis

7.1.1. BPMiSearch, an Interactive Search-based Service Identification Framework

The main outcome of this research is the BPMiSearch layered framework, which derives candidate SOA solutions from the organisation’s BPMs. It can be concluded that creating a service identification framework with these layers, in which each layer underpins a specific functionality, is critical in fulfilling all the business requirements. The flexibility of this design enables the generalisation and extension of the framework. For example, the genetic algorithm can be replaced by another technique (i.e., simulated annealing) using the same input and produce services in the same format. Utilising standard input notation (i.e., BPMN), and a standard output format (i.e., WSDL files) supports the flexibility of the framework’s design. It is also concluded that managing a top-down input artefact requires a mapping approach to frame the input business process models at an appropriate level of granularity. This layer is significant to generalise the framework. For example, the iterative development started with a manual mapping, then this method has been replaced by a search-based technique, and finally, an interactive search-based technique has been used to perform the search process. The entire architecture of the service identification framework has not been affected by using different techniques. As a result, a comparison between different techniques (i.e., sometimes the technique represents the current state of the framework such as the manual mapping or search-based framework) has been conducted to investigate the impact of each technique on the quality of the outcomes. In addition, the design of this framework enables the adoption of different techniques in the future which supports making more comparisons. The output service refinement layer has high importance as it constructs the SOA solution from a set of candidate services. The construction of WSDL files frames the candidate services in a form that is easier for software developers to implement.

However, BPMiSearch framework has some limitations. Firstly, the business-IT alignment has been achieved by deriving the services from the business requirements (i.e., BPMN models) not the business goals. Using business goals may require the adoption of different design metrics to fulfil these goals which is anticipated to produce different solutions. Secondly, tracing each business process elements in the two directions (i.e., from the BPMN to the services and then from services to BPMN) is not possible for all different types of elements. In other words, it is not possible for the BPMiSearch framework to construct the BPMN from a set of services. The reason is that there is no one to one mapping in the two directions. For example, gateway elements, lanes, and events help to group elements in candidate services, but these elements do not appear in the resulting web services. In addition, though the connection between elements is preserved, the allocation of these elements is changed. Therefore, there is no way to derive the BPMN models from the services.

In response to **RQ1** that states: "**To what extent can role-based business process models such as BPMN 2.0 models be mapped to services following Service Oriented Architectures (SOAs) principles?**"

It is concluded that role-based business process models can be mapped to the SOA. This mapping can be useful in producing feasible candidate services that achieve business-IT alignment. However, this mapping has some limitations as it does not use the business goals as input, and also there is no full two-way traceability for all elements from the BPMN to services.

Overall, with BPMiSearch being a top-down approach, BPM-driven, interactive, domain-independent and semi-automated using an interactive search algorithm that adheres to SOA principles means it also contributes to the paradigm of SOA by using iSBSE to simplifying the service identification process and enrich the quality of resulting candidate services. Furthermore, it supports the emerging trend of business-IT alignment and adds a further contribution to the trend of bridging the gap between the business context and the software system-to-be (Odeh and Kamm, 2003).

7.1.2. Implementation of the Service Identification Problem

Responding to **RQ2** that states: "**In what way can SOA services be best implemented for metaheuristic search?**", a formulation of the service identification problem has been adopted to enable the utilisation of an integer-based bin-packing representation method. The contribution of this research with reference to the problem implementation is that the service identification problem has been formulated as a bin-packing optimisation problem to enable the allocation of business process elements (i.e., derived from the input BPMN models and prepared in the format of Search Space Elements) in feasible candidate services. Furthermore, corresponding genetic operators and fitness function have been formulated to enable the exploration and exploitation of the search space as presented in chapters four and five. Indeed, this representation method provides a useful abstraction of the essential characteristics of the service identification solution. The formulation of service identification problem to utilise this representation effectively facilitates the visualisation for the interacting human domain expert as presented in chapter six. Noticeably, this feature is useful to minimise the cognitive load on the domain expert, mainly with large-scale projects that may be one of the many causative factors behind the difficulties of constructing SOA solutions. Additionally, this integer-based representation enables efficient genetic operators and offers a good level of traceability from the service identification problem to the service identification solution. The representation method has supported implementing the key constraints of SOA services in order to ensure that resulting solutions are feasible and conform to SOA principles. However, the service refinement layer uses the candidate services that result after performing the search and maps the business process elements inside these candidate services to the corresponding service components based on the standard WSDL specifications. Since the WSDL file (i.e., XML-based) has a tree structure, using a tree-based representation can be a potential choice to construct WSDL files. Nevertheless, the tree-based representation is more complex and less efficient to perform the search process and the check for service constraints.

7.1.3. Exploration and Exploitation of Service Identification Search Space

In answering **RQ3** that states: "**Can the services solution space be effectively and efficiently explored and exploited in order to derive candidate SOA solutions from BPMN 2.0 models?**", the results of the initial experiments presented in chapter five show that striking a balance between exploration and exploitation of the search space is a critical factor in producing effective and efficient solutions. Results reveal that performing a balance between coupling and cohesion produces more effective and efficient solutions. However, it is concluded that adopting parameter sets that work efficiently for other problem contexts does not necessarily lead to the same effectiveness with the service identification problem. Therefore, manual parameter tuning using an empirical trial-and-error method means the fitness of the resulting solution is highly satisfactory. Furthermore, the computational speed of the localised search is acceptable. In addition, the visualisation tools support human comprehension, and thus more reliable evaluation can be conducted. It is concluded that the search method with the tool support represents the base for interactive evolutionary search in which the human steers the trajectory of the search to enrich the quality of the resulting solutions. The interactive experiments' results suggest that the interactive search can help to increase the satisfaction and acceptance of resulting solutions.

A single-objective GA has been adopted as it can be sensitive to the interactive preference of participants. However, using a single-objective GA with weighted-sum aggregation fitness function has some limitations (e.g., the use of a limited number of design metrics that do not cover all the SOA principles). In addition, using an empirical trial-and-error for genetic parameter tuning can be replaced by a dynamic parameter control in order to find a more effective combination of parameters.

7.1.4. Collaborative Expert/Computer Interaction

The natural interaction between software engineers and the search algorithm is necessary to develop the interactive BPMiSearch framework to derive candidate services from role-based BPMs. In response to **RQ4** that states: "**What is the impact of the human interactive preference on the search outcomes?**", a number of challenges have been addressed in chapter six in order to develop an interactive search experience in such a way that the domain experts interact jointly to steer the trajectory of the search. As described in chapter six, BPMiSearch (i.e., the presented framework for service identification) has been developed to provide a context for the domain experts to steer the direction of the search towards more promising solutions. The key findings of this chapter are (i) the interactive search has a significantly positive influence on the fitness values of the outcomes, and (ii) the stakeholders' level of satisfaction and acceptance has positively increased. However, the non-interactive search is faster to find solutions in comparison with the interactive search, but this limitation is caused because the subjective preference of the experts when providing feedback.

It is concluded that within a service identification experience, the interactive search can provide an effective, natural, and interactive mechanism to narrow and focus the search to find promising high-quality solutions. The feedback from domain experts greatly assisted by the quantitative fitness values presented to them supports the quality improvement of the resulting solutions. Another significant finding which has been concluded is that the colourful visualisation of the connections between services helps to hold expert engagement. As a result, visualisation minimises human fatigue and leads to more reliable evaluation. A natural, collaborative interaction between the human domain expert and the interactive search-based framework has been achieved. Such a collaborative interaction appears to be effective in encouraging the discovery of SOA candidate solutions. The derived solutions using BPMiSearch have achieved a high level of trust, acceptance and satisfaction.

However, the main limitation of this experience is the adoption of two quantitative design metrics only (i.e., coupling and cohesion) and one qualitative metric to assess the quality of the candidate services during the search process. Besides the importance of the coupling and cohesion metrics, using a combination of these metrics in addition to the human preference values has an advantage of making the search algorithm more sensitive to the human interaction. The advantage of using a simple combination of metrics is to give a clear example on how the interactive search can be powerful. In contrast, to ensure that the resulting candidate services adhere to different SOA principles (e.g., modularity and reusability of services), more design metrics should be adapted. Using multiple design metrics requires the adoption of a multi/many-objective GA (e.g., NSGA-II or NSGA-III) to manage the interactive search.

7.1.5. The Adoption of DSRM and the CCR Case Study

The development of this research has been framed using DSRM. The DSRM has many advantages that make it useful to conduct this research. The DSRM phases support the incremental development of the research artefacts, starting from the theoretical grounding of the problem until the later stages of evaluation and communication. It is concluded that the iterative nature of DSRM provides the researcher of a sense of the progress while developing the framework, which positively allows constructing the project in smaller and simpler parts of the framework. In addition, DSRM allows the adoption of both quantitative and qualitative evaluation methods to evaluate the research outcomes. This helps to evaluate the outcomes after each iteration. However, doing the development of the service identification framework on different iterations, then demonstrate and evaluate the new parts could be more complicated and a challenging task. This requires more effort to prepare the design from the beginning as well as to adopt appropriate evaluation techniques to examine the newly added part in each iteration before proceeding to the next iteration. In some cases, it could be difficult to examine the additional parts of the iteration as the resulting framework performs as one part.

With reference to the case study, BPMiSearch has been demonstrated using the CCR case study as revealed in chapters three, four, five and six. The selection of the CCR was significant, not only because it is considered as a representative case study, but also because it contributes to the "SOA and Healthcare" domain. In addition, by selecting a representative and sufficient (i.e., the main characteristics that make the CCR sufficient and representative are its comprehensiveness, the large scale, high complexity, and flexibility), it is anticipated that the service identification framework can be applied to derive services using case studies from different domains. It is concluded that the adoption of the CCR case study provides a reliable solution to these organisations as they can effectively improve their operational activities, manage their resources, and reduce the running costs. It is also concluded that the flexibility of the case study is a very important characteristic that should be available for the evaluation. However, since the initial development of CCR was in 2006, and then some changes have emerged to the workflow, it is difficult for some domain experts to understand the former processes presented in the CCR BPMs as they are out of date. The sufficiency and representativeness of the CCR have not been affected by this, but understanding the workflow became more challenging for domain experts. Therefore, it is recommended to work on an updated version of the CCR or select a new case study in the future to test the service identification framework.

Using the CCR case study during the iterations of DSRM provides a piece of evidence on how powerful the DSRM is. It has allowed the case study to be utilised in different ways during different iterations. In addition, it shows how sufficient and flexible the CCR case study is. It has been in different ways to evaluate different aspects of the framework during the iterations. This useful characteristic of the CCR has supported the incremental development and demonstration of the BPMiSearch framework over multiple DSRM iterations. For example, in Chapter 4, the SSEs have been derived from the CCR BPMN to be used for manual service identification. In Chapter 5, data segments from the case study have been used by domain experts to evaluate the outcomes on the non-interactive search, whereas, in Chapter 6, domain experts have evaluated different data segments of the business process during the interactive search process. Furthermore, the experts compare the interactive and non-interactive search outcomes based on the entire data segments in the outcomes. The flexibility of the CCR case study allows working on these distinct data segments to fulfil the evaluation objectives.

7.1.6. Overall

The above discussion reveals that the early-identified research questions (i.e., RQ1 - RQ4) have been answered and consequently addresses the following research hypothesis: "**Using interactive metaheuristic search facilitates the derivation of candidate software services from role-based business process models (BPMs), and in particular BPMN process models**". It is concluded that interactive evolutionary computation can provide significant opportunities for an effective combination of quantitative and qualitative search and exploration of the service identification solution space. Successful expert/machine collaboration within an interactive context is the most critical aspect contributing to that success. This effective integration has resulted in a smooth engagement of the domain experts to conduct

SOA service identification. Furthermore, it is concluded that automating or semi-automating the activities of the top-down service identification framework (i.e., data preparation and data refinement phases are fully automated, whereas the service identification phase is semi-automated as it requires human interaction) helps to produce feasible candidate services that fulfil the business requirements with less errors and reduced costs. In addition, the capability of constructing multiple solutions provides an opportunity to find good-enough (i.e., with high fitness values) optimisations among the resulting services. The services produced using the interactive framework have a higher level of experts' satisfaction and acceptance. Finally, it is concluded that such computationally intelligent tool can provide opportunities for the discovery of useful SOA candidate services which may, in turn, yield significant SOA development gains.

However, the BPMiSearch framework has some limitations. These limitations include that BPMiSearch framework does not support deriving services based on the business goals. To develop such an aspect, more design metrics should be used to satisfy different goals, which requires the adoption of a MOGA rather the SOGA used in this research. Using a MOGA with a large number of design metrics can make the interactive search less sensitive to the human preference, which is a new challenge. Finally, being unable to trace all the business process elements in both directions (i.e., from BPMs to SOA and then from SOA to BPMs) limits the verification process to some extent as there is no direct one-to-one mapping between the business process elements and service components.

This research has an impact on both academia and industry. Firstly, this research shows an application of the interactive search on a real problem in software engineering (i.e. the service identification), which helps to build a deeper understanding to the influence of human interactive preference on the search. In addition, by developing the BPMiSearch framework, this research stimulates researchers to do further attempts in service identification by improving the interactive search technique to make it more effective and sensitive to human feedback.

Regarding the impact on the industry, the use of automatic (or semi-automatic) derivation of services helps organisations to effectively create and upgrade their software systems more effectively (i.e., high quality) and efficiently (i.e., fewer errors, time, and money). Using this tool in the industry can help organisations to immediately reflect the changes on business process to the corresponding software services. This utilisation of the interactive search in service identification is anticipated to become an evolutionary approach to achieve the business-IT alignment.

7.2. Future Directions

Future work includes investigating techniques to derive the services from the enterprise business goals. This may require extending the BPMiSearch framework to define more design metrics that satisfy business goals. BPMiSearch framework can be further improved by including non-functional requirements such as the Quality of Services (QoS). Using more design metrics to evaluate the candidate services encourages the adoption of a multi/many objective search algorithm (e.g., NSGA-II and NSGA-III) to facilitate the adoption of multiple fitness measures such as elegance, modularisation and reusability.

Although BPMiSearch supports mapping the business process to candidate SOA services, investigating the effectiveness and efficiency achieved by using different representation techniques is also a future direction. For example, the structure of tree-based representation looks like the structure of the WSDL file, which makes it a potential candidate representation. This selection requires more investigation in the future.

With regard to the interactive experience, future work also includes investigation of the approaches in which the human-machine interaction might be enhanced to more sensitively reflect evaluation intentions, and so further reduce human fatigue. Adding more design metrics would affect the sensitivity of the interactive algorithm, but it is an important future direction to investigate.

With reference to the software services, the new trend of using Microservices Architecture has become an important option. The advantages of these services include easy deployment, easy maintenance, and good fault tolerance (Chen, 2018). Using Microservices systems as the output of the service identification framework supports different aspects such as agility and reusability. Therefore, using this architecture is considered a future work.

Regarding the genetic parameters, tuning the genetic parameters using dynamic parameter control is anticipated to provide a wider range of parameter probabilities that support finding a more effective search and exploration experience. Dynamic parameter tuning offers enormous potential to reduce the execution time as well as the error rate, therefore, it is highly recommended that a dynamic parameter control via self-adaptation is adapted to tune the parameters as presented by Simons and Parmee (2010). Finally, as suggested by participants, some capabilities may help to enhance the quality of the search outcomes. For example, one capability may be to manually move individual elements from one service to another within a SOA solution. This feature gives the participant more freedom to shape the services in a different way, which, it is anticipated, will accelerate the arrival of high-quality solutions. Furthermore, assigning more weight to the human value in the fitness function could be useful with highly-experienced domain experts and this may help to reach more effective and efficient solutions.

References

Abd Rahman, R., Ramli, R., Jamari, Z., & Ku-Mahamud, K. R. (2016). *Evolutionary algorithm with roulette-tournament selection for solving aquaculture diet formulation. Mathematical Problems in Engineering, 2016.*

Aburub, F., Odeh, M., Beeson, I., Pheby, D., & Codling, B. (2008). *Modelling healthcare processes using role activity diagramming. International Journal of Modelling and Simulation*, 28(2), 147-155.

Afzal, W., Torkar, R., & Feldt, R. (2009). A systematic review of search-based testing for non-functional system properties. *Information and Software Technology*, 51(6), 957-976.

Ahmad, M., & Odeh, M. (2012). Semantic Derivation of Enterprise Information Architecture from Business Process Architecture. *In Computer Theory and Applications (ICCTA), 2012 22nd International Conference on* (pp. 79-84). IEEE.

Ahmad, M. (2016). Semantic derivation of enterprise information architecture from riva-based business process architecture *(Doctoral dissertation, University of the West of England).*

Ahmed, F., & Deb, K. (2013). Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing, 17(7), 1283-1299.*

Alghamdi, B., Potter, L. E., & Drew, S. (2016). Identifying Best Practices in organisational SOA Governance Adoption: Case Study of Saudi Arabia's E-Government Programme. In PACIS (p. 365).

Alajmi, A., & Wright, J. (2014). Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem. *International Journal of Sustainable Built Environment*, 3(1), 18-26.

Aljawawdeh, H. J., Simons, C. L., & Odeh, M. (2015). Metaheuristic design pattern: Preference. *In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 1257-1260). ACM.

Alkkiomäki, V. (2016). The role of service-oriented architecture as a part of the business model. *International Journal of Business Information Systems*, 21(3), 368-387.

Allweyer, T. (2016). BPMN 2.0: introduction to the standard for business process modeling. *BoD–Books on Demand*.

Al-Thuhli, A., Al-Badawi, M., Baghdadi, Y., & Al-Hamdani, A. (2015). Migrating social business process to SOA. *In Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services* (p. 89). ACM.

Axelsson, S., Baca, D., Feldt, R., Sidlauskas, D., & Kacan, D. (2009). Detecting defects with an interactive code review tool based on visualisation and machine learning. *In the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE 2009).*

Atlas, (2018). Qualitative and Quantitative Research. Available at: http://atlasti.com/quantitative-vs-qualitative-research [Accessed 27 February 2018].

Azevedo, L. G., Santoro, F., Baião, F., Souza, J., Revoredo, K., Pereira, V., & Herlain, I. (2009). A method for service identification from business process models in a SOA approach. *In Enterprise, Business-Process and Information Systems Modeling (pp. 99-112). Springer, Berlin, Heidelberg.*

Azevedo, L. G., Santoro, F., Baião, F., Diirr, T., Souza, A., de Souza, J. F., & Sousa, H. P. (2014). A method for bridging the gap between business process models and services. *iSys-Revista Brasileira de Sistemas de Informação,* 6(1), 62-98.

Azevedo, L. G., Santoro, F., Baião, F., Souza, J., Revoredo, K., Pereira, V., & Herlain, I. (2009). A method for service identification from business process models in a SOA approach. *In Enterprise, Business-Process and Information Systems Modeling* (pp. 99-112). Springer, Berlin, Heidelberg.

Back, T. (1996). Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. *Oxford university press.*

Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (2000). Evolutionary computation 1: Basic algorithms and operators *(Vol. 1). CRC press.*

Badica, C., Badica, A., & Litoiu, V. (2003). Role activity diagrams as finite state processes. In null (p. 15). IEEE.

Baghdadi, Y. (2006). Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing. Information systems frontiers, 8(5), 395-410.

Bandyopadhyay, S., & Saha, S. (2012). Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications. *Springer Science & Business Media*.

Bartley, M. G., Galpin, D., & Blackmore, T. (2002). A comparison of three verification techniques: directed testing, pseudo-random testing and property checking. *In Proceedings of the 39th annual Design Automation Conference* (pp. 819-823). ACM.

Bavota, G., Di Penta, M. and Oliveto, R. (2014). Search based software maintenance: Methods and tools*. In Evolving Software Systems* (pp. 103-137). Springer, Berlin, Heidelberg.

Berger, B., & Leighton, T. (1998). Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1), 27-40.

Berkey, J. O., & Wang, P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the operational research society*, 38(5), 423-429.

Bianchini, D., Cappiello, C., De Antonellis, V., & Pernici, B. (2009). P2S: A methodology to enable inter-organizational process design through web services. *In International Conference on Advanced Information Systems Engineering* (pp. 334-348). Springer, Berlin, Heidelberg.

Bianchini, D., Cappiello, C., De Antonellis, V., & Pernici, B. (2014). Service identification in inter-organizational process design. *IEEE Transactions on Services Computing,* (1), 1.

Birkmeier, D. Q., Gehlert, A., Overhage, S., & Schlauderer, S. (2013). Alignment of business and IT architectures in the German federal government: A systematic method to identify services from business processes. *In System Sciences (HICSS), 2013 46th Hawaii International Conference* on (pp. 3848-3857). IEEE.

Blickle, T., & Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4), 361-394.

Boerner, R., & Goeken, M. (2009). Service identification in SOA governance literature review and implications for a new method. In Digital Ecosystems and Technologies, 2009. *DEST'09. 3rd IEEE International Conference* on (pp. 588-593). IEEE.

Börner, R., Goeken, M., & Rabhi, F. (2012). SOA development and service identification: A case study on method use, context and success factors (No. 189). *Working Paper Series, Frankfurt School of Finance & Management*.

Bowling, A. (1997). Research methods in health. *Buckingham. Open University Press.*

Bowman, M., Briand, L. C., & Labiche, Y. (2010). Solving the class responsibility assignment problem in object-oriented analysis with multi-objective genetic algorithms. *IEEE Transactions on Software Engineering*, 36(6), 817-837.

Bozkurt, M., Harman, M., & Hassoun, Y. (2013). Testing and verification in service-oriented architecture: a survey. *Software Testing, Verification and Reliability*, 23(4), 261-313.

Breitmayer, B. J., Ayres, L., & Knafl, K. A. (1993). Triangulation in qualitative research: Evaluation of completeness and confirmation purposes. *Image: The Journal of Nursing Scholarship,* 25(3), 237-243.

Buttle F (1996) SERVQUAL: review, critique, research agenda. *European Journal of Marketing*, 30(1), 8-32.

Burns, N., & Grove, S. K. The Practice of Nursing Research: Conduct, Critique, and Utilization.

Cabanillas, C., Baumgrass, A., Mendling, J., Rogetzer, P., & Bellovoda, B. (2013). Towards the enhancement of business process monitoring for complex logistics chains. *In International Conference on Business Process Management* (pp. 305-317). Springer, Cham.

Calcagno, C., Distefano, D., Dubreil, J., Gabi, D., Hooimeijer, P., Luca, M., ... & Rodriguez, D. (2015). Moving fast with software verification. *In NASA Formal Methods Symposium* (pp. 3-11). Springer, Cham.

Cardoso, J. (2005). How to measure the control-flow complexity of web processes and workflows. *Workflow handbook*, 2005, 199-212.

Chekuri, C., & Khanna, S. (1999, January). On multi-dimensional packing problems. *In Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 185-194). Society for Industrial and Applied Mathematics.

Chen, L. (2018). Microservices: Architecting for Continuous Delivery and DevOps. *In IEEE International Conference on Software Architecture (ICSA).*

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., & Rackham, G. (2005). Impact of service orientation at the business level. *IBM Systems Journal,* 44(4), 653-668.

Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1), 124-134.

Cho, Y. H., Kim, J. K., & Hie, K. S. (1998). Role-based approach to business process simulation modeling and analysis. Computers & industrial engineering, 35(1-2), 343-346.

Cohen, J. (1992). A power primer. *Psychological bulletin,* 112(1), 155.

Collis, J., & Hussey, R. (2013). Business research: A practical guide for undergraduate and postgraduate students. *Macmillan International Higher Education*.

Columbia University, (2017). Figure 1 – Activity diagram of a typical GA. [image] Available at: http://www.columbia.edu/~zj2170/schoolbus.html [Accessed 17 Nov 2017].

Christophe, B., Boussard, M., Lu, M., Pastor, A., & Toubiana, V. (2011). The web of things vision*: Things as a service and interaction patterns. Bell labs technical journal*, 16(1), 55-61.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287.

Davis, L. (1991). Handbook of genetic algorithms.

De Jong and Jayshree Sarma. (1992). Generation gap revisited. *Whitley*, 2419, 19-28.

De Jong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems.

De Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., ... & Weyns, D. (2013). Software engineering for self-adaptive systems: A second research roadmap. *In Software Engineering for Self-Adaptive Systems II (pp. 1-32).* Springer, Berlin, Heidelberg.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.

Dijkman, R. M., Dumas, M., & Ouyang, C. (2008). Semantics and analysis of business process models in BPMN. *Information and Software technology,* 50(12), 1281-1294.

Döhring, M., Zimmermann, B., & Karg, L. (2011). Flexible workflows at design-and runtime using BPMN2 adaptation patterns. *In International Conference on Business Information Systems* (pp. 25-36). Springer, Berlin, Heidelberg.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming,* 91(2), 201-213.

Dwivedi, V., & Kulkarni, N. (2008). A model driven service identification approach for process centric systems. *In Congress on Services Part II, 2008. SERVICES-2. IEEE* (pp. 65-72). IEEE.

Durán-Muñoz, I., & Bautista-Zambrana, M. R. Applying Ontologies to Terminology: Advantages and Disadvantages. *HERMES-Journal of Language and Communication in Business,* 26(51), 65-77.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. *In Guide to advanced empirical software engineering* (pp. 285-311). Springer, London.

Ebrahimifard, A., Amiri, M. J., Arani, M. K., & Parsa, S. (2016). Mapping BPMN 2.0 choreography to WS-CDL: a systematic method. *Journal of E-Technology*, 7(1), 1-23.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. *In Guide to advanced empirical software engineering* (pp. 285-311). Springer, London.

Eiben, A. E., & Smith, J. E. (2003). Introduction to evolutionary computing (Vol. 53). Berlin: springer.

El Hayek, J., Moukrim, A., & Negre, S. (2008). New resolution algorithm and pretreatments for the two-dimensional bin-packing problem. *Computers & Operations Research*, 35(10), 3184-3201.

Erl. T. (2007). SOA principles of service design. Boston*: Prentice Hall*, 37, 71-75.

Erl, T. (2008). SOA design patterns. *Pearson Education*.

Farrell, J., & Lausen, H. (2006). Semantic Annotations for WSDL, W3C Working Draft.

Feldt, R., & Magazinius, A. (2010). Validity Threats in Empirical Software Engineering Research-An Initial Survey. In *SEKE* (pp. 374-379).

Fernandez, A. (2013). Camunda BPM platform loan assessment process lab. Brisbane, Australia: Queensland University of Technology.

Field, A. (2000). Discovering statistics using spss for windows. London-Thousand Oaks- New Delhi: *Sage publications*.

Field, A. (2009). Discovering statistics using SPSS. London: *SAGE*.

Frey, F. J., Hentrich, C., & Zdun, U. (2015). Capability-based service identification in service-oriented legacy modernization. *In Proceedings of the 18th European Conference on Pattern Languages of Program* (p. 10). ACM.

Geambasu, C.V. (2012). BPMN vs. UML activity diagram for business process modeling. *Accounting and Management Information Systems,* 11(4), p.637.

Gerring, J., & Cojocaru, L. (2016). Case-selection: A diversity of methods and criteria. *Sociological Methods & Research*, 45, 392-423.

Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, 13(2), 209-228.

Gligor, V. D., Gavrila, S. I., & Ferraiolo, D. (1998). On the formal definition of separation-of-duty policies and their composition. *In Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on* (pp. 172-183). IEEE.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *In Foundations of genetic algorithms* (Vol. 1, pp. 69-93). Elsevier.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning, 3(2), 95-99.

Gravetter, F., & Wallnau, L. (2014). *Essentials of statistics for the behavioral sciences (8th ed.). Belmont, CA: Wadsworth*.

Greene, J., & McClintock, C. (1985). Triangulation in evaluation: Design and analysis issues. Evaluation review, 9(5), 523-545.

Greene, J. C., Caracelli, V. J., & Graham, W. F. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational evaluation and policy analysis, 11(3), 255-274.*

Greenwood, D. (2008). Goal-oriented autonomic business process modeling and execution: Engineering change management demonstration. *In International Conference on Business Process Management (pp. 390-393). Springer, Berlin, Heidelberg*.

Gruhn, V., & Laue, R. (2006). Complexity metrics for business process models. *In 9th international conference on business information systems (BIS 2006)* (Vol. 85, pp. 1-12).

Gu, Q., & Lago, P. (2010). Service identification methods: a systematic literature review. *In European Conference on a Service-Based Internet (pp. 37-50). Springer*, Berlin, Heidelberg.

Guan, Q., Feng, S., & Ma, Y. (2012). A network topology clustering algorithm for service identification. *In Computer Science & Service System (CSSS), 2012 International Conference on* (pp. 1583-1586). IEEE.

Guindon, R. (1990). Designing the design process: Exploiting opportunistic thoughts. *Human-Computer Interaction*, 5(2), 305-344.

Guo, P., Wang, X., & Han, Y. (2010, October). The enhanced genetic algorithms for the optimization design. In *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference* on (Vol. 7, pp. 2990-2994). IEEE.

Harman, M., & Jones, B. F. (2001). Search-based software engineering. *Information and software Technology*, 43(14), 833-839.

Harman, M., & McMinn, P. (2010). A theoretical and empirical study of search-based testing: Local, global, and hybrid search. *IEEE Transactions on Software Engineering,* 36(2), 226-247.

Harman, M. (2011). Software engineering meets evolutionary computation. *Computer*, (10), 31-39.

Harman, M., Hassoun, Y., Lakhotia, K., McMinn, P., & Wegener, J. (2007). The impact of input domain reduction on search-based test data generation. *In Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 155-164). ACM.

Harman, M., Mansouri, S. A., & Zhang, Y. (2012). Search-based software engineering: Trends, techniques and applications. *ACM Computing Surveys (CSUR)*, 45(1), 11.

Harmon, P. (2003). Business process change: a manager's guide to improving, redesigning, and automating processes. *Morgan Kaufmann.*

Havey, M. (2005). Essential business process modeling. *" O'Reilly Media, Inc.".*

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems,* 19(2), 4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). *Design science in information systems research*. MIS Q 28 (1): 75–105.

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. *MIT press.*

Huergo, R. S., Pires, P. F., Delicato, F. C., Costa, B., Cavalcante, E., & Batista, T. (2014). A systematic survey of service identification methods. Service Oriented Computing and Applications, 8(3), 199-219.

Inaganti, S., & Behara, G. K. (2007). Service identification: BPM and SOA handshake. BPTrends, 3, 1-12.

Jafarov, N., & Lewis, E. (2015). Reinterpreting the principles of SOA through the cybernetic concepts of VSM to design the ESB as iPaaS in the cloud. *In Science and Information Conference (SAI),* 2015 (pp. 850-858). IEEE.

Jamshidi, P., Mansour, S., Sedighiani, K., Jamshidi, S., & Shams, F. (2012). An automated service identification method. *Technical Report, TR-ASER-2012-01. Automated Software Engineering Research Group, Shahid Beheshti University.*

Jamshidi, P., Khoshnevis, S., Teimourzadegan, R., Nikravesh, A., & Shams, F. (2009). An automated method for service specification*. In Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010* (pp. 25-28). ACM.

Jamshidi, P., Sharifi, M., & Mansour, S. (2008). To establish enterprise service model from enterprise business model. *In Services Computing, 2008. SCC'08. IEEE International Conference on* (Vol. 1, pp. 93-100). IEEE.

Jones, D. (2018). Programming using genetic algorithms: isn't that what humans already do. Available at: http://shape-of-code.coding-guidelines.com/2013/10/18/programming-using-genetic-algorithms-isnt-that-what-humans-already-do/ [Accessed 8 January 2018].

Jureta, I., Kolp, M., Faulkner, S., & Wautelet, Y. (2005). A goal-oriented framework for business modeling *(No. UCL-Université Catholique de Louvain). Working Paper IAG, 05/126.*

Kaabi, R. S., Souveyet, C., & Rolland, C. (2004). Eliciting service composition in a goal driven manner. *In Proceedings of the 2nd international conference on Service oriented computing* (pp. 308-315). ACM.

Kamienski, C., Dantas, R., Fildago, J., Sadok, D., & Ohlman, B. (2010). Service creation and execution with the Service Refinement Cycle. *In Network Operations and Management Symposium (NOMS*), 2010 IEEE (pp. 829-832). IEEE.

Kazemi, A., Rostampour, A., Jamshidi, P., Nazemi, E., Shams, F., & Azizkandi, A. N. (2011). A genetic algorithm based approach to service identification. *In Services (SERVICES), 2011 IEEE World Congress* on (pp. 339-346). IEEE.

Kephart, J. O., & Chess, D. M. (2003*). The vision of autonomic computing. Computer,* (1), 41-50.

Khan, Z., Odeh, M., & McClatchey, R. (2011). Bridging the gap between business process models and service-oriented architectures with reference to the grid environment. International Journal of Grid and Utility Computing, 2(4), 253-283.

Khlif, W., Makni, L., Zaaboub, N., & Ben-Abdallah, H. (2009). Quality metrics for business process modeling. *In Proceedings of the 9th WSEAS international conference on Applied computer science* (pp. 195-200). World Scientific and Engineering Academy and Society (WSEAS).

Khlif, W., Zaaboub, N., & Ben-Abdallah, H. (2010). Coupling metrics for business process modeling. *WSEAS Transactions on Computers,* 9(1), 31-41.

Khoshnevis, S., & Shams, F. (2017). Automating identification of services and their variability for product lines using NSGA-II. *Frontiers of Computer Science,* 11(3), 444-464.

Kim, C. H., Weston, R. H., Hodgson, A., & Lee, K. H. (2003). The complementary use of IDEF and UML modelling approaches. *Computers in Industry*, 50(1), 35-56.

Kim, T., Lee, K., & Baik, J. (2015). An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm. *Journal of Systems and Software*, 102, 134-144.

Kim, Y., & Doh, K. G. (2009). Formal identification of right-grained services for service-oriented modeling. *In International Conference on Web Information Systems Engineering* (pp. 261-273). Springer, Berlin, Heidelberg.

Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). *Preliminary guidelines for empirical research in software engineering. IEEE Transactions on software engineering,* 28(8), 721-734.

Klien, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R., & Feltovich, P. J. (2004). Ten challenges for making automation a" team player" *in joint human-agent activity. IEEE Intelligent Systems*, 19(6), 91-95.

Klose, K., Knackstedt, R., & Beverungen, D. (2007, June). Identification of Services-A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning. *In ECIS* (Vol. 7, pp. 1802-1814).

Kohlborn, T., Korthaus, A., Chan, T., & Rosemann, M. (2009). Identification and analysis of business and software services—a consolidated approach. *IEEE Transactions on Services Computing*, 2(1), 50-64.

Kotonya, G., & Sommerville, I. (1998). Requirements engineering: processes and techniques. *Wiley Publishing*.

Kueng, P., & Kawalek, P. (1997). Goal-based business process models: creation and evaluation. *Business process management journal*, 3(1), 17-38.

Lazar, J., Feng, J. H., & Hochheiser, H. (2017). Research methods in human-computer interaction. *Morgan Kaufmann*.

Leopold, H., & Mendling, J. (2012). Automatic derivation of service candidates from business process model repositories. In International Conference on Business Information Systems (pp. 84-95). *Springer, Berlin, Heidelberg*.

Li, K., Liu, H., Wu, Y., & Xu, X. (2014). A two-dimensional bin-packing problem with conflict penalties. *International Journal of Production Research*, 52(24), 7223-7238.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lima, S. R. & Huacarpuma, R. C. (2015). A Methodology for Identifying Candidate Services and Compositions. Service Technology Magazine.

Ma, Q., Zhou, N., Zhu, Y., & Wang, H. (2009). Evaluating service identification with design metrics on business process decomposition. *In Services Computing, 2009. SCC'09. IEEE International Conference on* (pp. 160-167). IEEE.

Malhotra, R., Singh, N., & Singh, Y. (2011). Genetic algorithms: Concepts, design for optimization of process controllers. *Computer and Information Science*, 4(2), 39.

Mani, S., Sinha, V. S., Sukaviriya, N., & Ramachandra, T. (2008). Using user interface design to enhance service identification. *In Web Services, 2008. ICWS'08. IEEE International Conference on* (pp. 78-87). IEEE.

Marculescu, B., Feldt, R., Torkar, R., & Poulding, S. (2015). An initial industrial evaluation of interactive search-based testing for embedded software. *Applied Soft Computing*, 29, 26-39.

Marculescu, B., Poulding, S., Feldt, R., Petersen, K., & Torkar, R. (2016). Tester interactivity makes a difference in search-based software testing: A controlled experiment. *Information and Software Technology*, 78, 66-82.

Mariani, T., & Vergilio, S. R. (2017). A systematic review on search-based refactoring. Information and Software Technology, 83, 14-34.

Meignan, D., Knust, S., Frayret, J. M., Pesant, G., & Gaud, N. (2015). A review and taxonomy of interactive optimization methods in operations research. *ACM Transactions on Interactive Intelligent Systems (TiiS*), 5(3), 17.

Munir, K., Odeh, M., & McClatchey, R. (2012). Ontology-driven relational query formulation using the semantic and assertional capabilities of OWL-DL. *Knowledge-Based Systems*, 35, 144-159.

Nielsen, J. (2005). Time budgets for usability sessions. Useit. com: Jakob Nielsen's web site, 12.

Noever, D., & Baskaran, S. (1992). Steady state vs. generational genetic algorithms: A comparison of time complexity and convergence properties. *Preprint series*, 92-07.

Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., & Rolland, C. (2005). A strategy driven business process modelling approach. *Business Process Management Journal,* 11(6), 628-649.

Object Management Group, (2011). Business Process Model and Notation. Available at: https://www.omg.org/bpmn/ [Accessed 2 Mar 2015].

Object Management Group, (2013). BPMN2 Modeler. Available at: https://www.eclipse.org/bpmn2-modeler/ [Accessed 22 September 2018].

Odeh, M., & Kamm, R. (2003). Bridging the gap between business models and system models. *Information and Software Technology,* 45(15), 1053-1060.

Odeh, Y. (2015). GQ-BPAOntoSOA: A goal-and object-based semantic framework for deriving software services from an organisation's goals and riva business process architecture *(Doctoral dissertation, University of the West of England).*

Offermann, P., Hoffmann, M., & Bub, U. (2009, September). Benefits of SOA: Evaluation of an implemented scenario against alternative architectures. *In Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009*. 13th (pp. 352-359). IEEE.

Orlikowski, W. J., & Baroudi, J. J. (1991) Studying information technology in organizations: Research approaches and assumptions. *Information systems research*, 2(1), 1-28.

OULD, M.A. and OULD, M. (1995). Business Processes: Modelling and analysis for re-engineering and improvement. *Wiley Chichester.*

Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. *In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on* (pp. 3-12). IEEE.

Papazoglou, M. (2008). Web services: principles and technology. *Pearson Education*.

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, August). Design methodology for web services and business processes. *In International Workshop on Technologies for E-Services* (pp. 54-64). Springer, Berlin, Heidelberg.

Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. *In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on* (pp. 3-12). IEEE.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2008). Service-oriented computing: a research roadmap. *International Journal of Cooperative Information Systems*, 17(02), 223-255.

Papazoglou, M. P., & Van Den Heuvel, W. J. (2007). Business process development life cycle methodology. *Communications of the ACM,* 50(10), 79-85.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: a model for producing and presenting information systems research. *In Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006)* (pp. 83-106). sn.

Pimentel, V., & Nickerson, B. G. (2012). Communicating and displaying real-time data with WebSocket. *IEEE Internet Computing*, 16(4), 45-53.

Pitangueira, A. M., Maciel, R. S. P., & Barros, M. (2015). Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. *Journal of Systems and Software*, 103, 267-280.

Pooley, R., & King, P. (1999). The unified modelling language and performance engineering. *IEEE Proceedings-Software*, 146(1), 2-10.

Poorazizi, M. E., Steiniger, S., & Hunter, A. J. (2015). A service-oriented architecture to enable participatory planning: an e-planning platform. *International Journal of Geographical Information Science*, 29(7), 1081-1110.

Prat, N., Comyn-Wattiau, I., & Akoka, J. (2014). Artifact Evaluation in Information Systems Design-Science Research-a Holistic View. *In PACIS* (p. 23).

Pufahl, L., & Weske, M. (2016). Enabling Batch Processing in BPMN Processes. In *BPM* (Demos) (pp. 28-33).

Qian, M., Zhou, N., Zhu, Y. Wang, H (2009). Evaluating Service Identification with Design Metrics on Business Process Decomposition. *IEEE International Conference on Services computing*.160-167.

Rai, M. M. (2006). Single-and multiple-objective optimization with differential evolution and neural networks. *VKI lecture series: introduction to optimization and multidisciplinary design, 58.*

Ramírez, A., Romero, J. R., & Ventura, S. (2015). An approach for the evolutionary discovery of software architectures. *Information Sciences,* 305, 234-255.

Ramirez, A., Romero, J. R., & Simons, C. (2018). A systematic review of interaction in search-based software engineering. *IEEE Transactions on Software Engineering.*

Ramírez, A., Romero, J. R., & Ventura, S. (2016). A comparative study of many-objective evolutionary algorithms for the discovery of software architectures. *Empirical Software Engineering*, 21(6), 2546-2600.

Reijers, H. A., & Vanderfeesten, I. T. (2004). Cohesion and coupling metrics for workflow process design. *In International Conference on Business Process management* (pp. 290-305). Springer, Berlin, Heidelberg.

Reijers, H. A. (2003). Design and control of workflow processes: business process management for the service industry. Springer-Verlag.

Rocha, M., & Neves, J. (1999). Preventing premature convergence to local optima in genetic algorithms via random offspring generation. *In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 127-136).* Springer, Berlin, Heidelberg.

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, (4), 557-572.

Rodrigues, R. W. (2000). Formalising UML activity diagrams using finite state processes. *In Proc. of the 3rd intl. conf. on the unified modeling language*, york, uk.

Rowlands, B. H. (2005). Grounded in practice: Using interpretive research to build theory. *The Electronic Journal of Business Research Methodology*, 3(1), 81-92.

Rub, A., & Farid, F. A. (2006). A business process improvement methodology based on process modelling, applied to the healthcare sector (*Doctoral dissertation, University of the West of England, Bristol*).

Ryan, A. B. (2006). Post-positivist approaches to research. Researching and Writing your Thesis: a guide for postgraduate students, 12-26.

Saidani, O., & Nurcan, S. (2006). A Role-Based Approach for Modeling Flexible Business Processes. *BPMDS*, 6, 111-120.

Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3), 263-278.

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. IEEE Transactions on software engineering, (4), 557-572.

Seawright, J., & Gerring, J. (2008). Case selection techniques in case study research: A menu of qualitative and quantitative options. *Political Research Quarterly*, 61(2), 294-308.

Seng, O., Stammel, J., & Burkhart, D. (2006, July). Search-based determination of refactorings for improving the class structure of object-oriented systems. *In Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 1909-1916). ACM.

Shackelford, M., & Simons, C. L. (2014). Metaheuristic design pattern: Interactive solution presentation. *In Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation* (pp. 1431-1434). ACM.

Shahrbanoo, M., Ali, M., & Mehran, M. (2012). An Approach for agile SOA development using agile principals*. arXiv preprint arXiv:1204.0368.*

Sharma, C., Sabharwal, S., & Sibal, R. (2014). A survey on software testing techniques using genetic algorithm. *arXiv preprint arXiv:1411.1154.*

Shashwat, A., & Kumar, D. (2017, February). A service identification model for service oriented architecture. *In Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference* on (pp. 1-5). IEEE.

Shim, B., Choue, S., Kim, S., & Park, S. (2008). A design quality model for service-oriented architecture. *In 2008 15th Asia-Pacific Software Engineering Conference* (pp. 403-410). IEEE.

Shirazi, H. M., Fareghzadeh, N., & Seyyedi, A. (2009). A combinational approach to service identification in SOA. *Journal of Applied Sciences Research*, 5(10), 1390-1397.

Simons, C. L., & Parmee, I. C. (2009, October). An empirical investigation of search-based computational support for conceptual software engineering design. *In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference* on (pp. 2503-2508). IEEE.

Simons, C. (2011). Interactive evolutionary computing in early lifecycle software engineering design (*Doctoral dissertation, University of the West of England*).

Simons, C. L., & Parmee, I. C. (2012). Elegant object-oriented software design via interactive, evolutionary computation. *arXiv preprint arXiv:*1210.1184.

Simons, C. L., Parmee, I. C., & Gwynllyw, R. (2010). Interactive, evolutionary search in upstream object-oriented class design. *IEEE Transactions on Software Engineering,* 36(6), 798-816.

Simons, C. L., & Parmee, I. C. (2010, October). Dynamic parameter control of interactive local search in UML software design. In *SMC* (pp. 3397-3404).

Simons, C. L., Smith, J., & White, P. (2014). Interactive ant colony optimization (iACO) for early lifecycle software design. *Swarm Intelligence*, 8(2), 139-157.

Singh, L., & Singh, S. (2014, March). A genetic algorithm for scheduling workflow applications in unreliable cloud environment. *In International Conference on Security in Computer Networks and Distributed Systems* (pp. 139-150). Springer, Berlin, Heidelberg.

Sjøberg, D. I., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N. K., & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE transactions on software engineering*, 31(9), 733-753.

Smith, J., & Vavak, F. (1999). Replacement strategies in steady state genetic algorithms: Static environments. *Foundations of genetic algorithms*, 5, 219-233.

Smith, J., & Simons, C. (2015). The influence of search components and problem characteristics in early life cycle class modelling. *Journal of Systems and Software*, 103, 440-451.

Snelick, R., Uludag, U., Mink, A., Indovina, M., & Jain, A. (2005). Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE transactions on pattern analysis and machine intelligence*, 27(3), 450-455.

Snelick, R., Uludag, U., Mink, A., Indovina, M., & Jain, A. (2005). Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE transactions on pattern analysis and machine intelligence*, 27(3), 450-455.

Souza, E., Moreira, A., & De Faveri, C. (2017). An approach to align business and IT perspectives during the SOA services identification. *In Computational Science and Its Applications (ICCSA), 2017 17th International Conference* on(pp. 1-7). IEEE.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9), 1275-1296.

Teka, A. Y., Condori-Fernandez, N., & Sapkota, B. (2012). A systematic literature review on service description methods. *In International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 239-255). Springer, Berlin, Heidelberg.

Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, 17(3), 411-436.

Tonella, P., Susi, A., & Palma, F. (2013). Interactive requirements prioritization using a genetic algorithm. *Information and software technology*, 55(1), 173-187.

Trochim, W. M., & Donnelly, J. P. (2006). The research methods knowledge base (3rd ed.). *Cincinnati, OH:Atomic Dog*.

Troiano, L., Birtolo, C., & Armenise, R. (2016). A validation study regarding a generative approach in choosing appropriate colors for impaired users. *SpringerPlus*, 5(1), 1090.

Tsai, W. T., Paul, R., Wang, Y., Fan, C., & Wang, D. (2002). Extending WSDL to facilitate web services testing. *In High Assurance Systems Engineering, 2002. Proceedings. 7th IEEE International Symposium* on (pp. 171-172). IEEE.

Upadhyaya, B., Khomh, F., & Zou, Y. (2012). Extracting RESTful services from Web applications. *In Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference* on (pp. 1-4). IEEE.

Vaishnavi, V., & Kuechler, W. (2004) Design Science research in information systems. Available from: http://www.desrist.org/design-research-in-information-systems/ [Accessed 22 February 2018].

Van Herk, H., Poortinga, Y. H., & Verhallen, T. M. (*2004). Response styles in rating scales: Evidence of method bias in data from six EU countries. Journal* of Cross-Cultural Psychology, 35(3), 346-360.

Vathsavayi, S., & Koskimies, K. (2013). Interleaving human and search-based software architecture design. *Proceedings of the Estonian Academy of Sciences,* 62(1).

Vavak, F., & Fogarty, T. C. (1996, April). A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. *In AISB workshop on Evolutionary Computing* (pp. 297-304). Springer, Berlin, Heidelberg.

Vomann, H (2006) Web Services. https://commons.wikimedia.org/wiki/File:Webservices.png Available from: [Accessed 15 February 2018].

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of information systems,* 4(2), 74-81.

Wang, H., Olhofer, M., & Jin, Y. (2017). A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3(4), 233-245.

Wang, Z., Xu, X., & Zhan, D. (2005, October). Normal forms and normalized design method for business service. *In e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on* (pp. 79-86). IEEE.

W3schools (2015). WSDL Documents. [online] Available at: http://www.w3schools.com/xml/xml_wsdl.asp [Accessed 9 Feb. 2016].

W3schools (2016). XML Web Services. [online] Available at: http://www.w3schools.com/xml/xml_services.asp [Accessed 9 Feb. 2016].

Weile, D. S., & Michielssen, E. (1997). Genetic algorithm optimization applied to electromagnetics: A review. *IEEE Transactions on Antennas and Propagation*, 45(3), 343-353.

Weissman, C. D., & Bobrowski, S. (2009, June). The design of the force. com multitenant internet application development platform. *In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (pp. 889-896). ACM.

Weller, J., Esswein, W., Stark, J., & Juhrisch, M. (2009). Meet the challenge in service identification: A ratio-based approach. *PACIS* 2009 Proceedings, 52.

Whitley, D. (1993). An executable model of a simple genetic algorithm*. In Foundations of genetic algorithms* (Vol. 2, pp. 45-62). Elsevier.

Whitner, R. B., & Balci, O. (1989). Guidelines for selecting and using simulation model verification techniques. *In Proceedings of the 21st conference on Winter simulation* (pp. 559-568). ACM.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering. Springer Science & Business Media.*

Yamany, E., Eid, A., Shaheen, M., & Sayyad, A. S. (2014). OPTI-SELECT: an interactive tool for user-in-the-loop feature selection in software product lines. *In Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools-Volume 2* (pp. 126-129). ACM.

Yousef, R., Adwan, O., & Abushariah, M. A. (2014). Extracting SOA Candidate Software Services from an Organization's Object Oriented Models. *Journal of Software Engineering and Applications*, 7(09), 770.

Yousef, R., Odeh, M., Coward, D., & Sharieh, A. (2011). Translating RAD Business Process Models into BPMN Models: A Semi-Formal Approach. *IJWA*, 3(4), 187-196.

Yousef, R. M. (2010). BPAOntoSOA: A Semantically Enriched Framework for Deriving SOA Candidate Software Services from Riva-based Business Process Architecture (*Doctoral dissertation, University of the West of England, Bristol*).

Yusoff, Y., Ngadiman, M. S., & Zain, A. M. (2011). Overview of NSGA-II for optimizing machining process parameters. *Procedia Engineering*, 15, 3978-3983.

Zadeh, A. T., Mukhtar, M., Sahran, S., & Lotfi, Z. (2016). AUTOMATED SERVICE IDENTIFICATION FRAMEWORK (ASIF). *Journal of Theoretical & Applied Information Technology*, 83(3).

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2013). Collaborative web service qos prediction via neighbourhood integrated matrix factorization. *IEEE Transactions on Services Computing*, 6(3), 289-299.

Zheng, Z., Zhang, Y., & Lyu, M. R. (2014). Investigating QoS of real-world web services. *IEEE transactions on services computing,* 7(1), 32-39.

Zowghi, D., & Gervasi, V. (2003). On the interplay between consistency, completeness, and correctness in requirements evolution. *Information and Software technology*, 45(14), 993-1009.

Appendix A
BPMN Models of KHCC

The business process models for the CCR case-study were developed by (Yousef 2010) in their research. These models have been replicated and upgraded from BPMN 1.0 to BPMN 2.0 by (Ahmad, 2016) using Camunda BPMN 2.0 Modeler utility. This tool produces the XML-based format of business process models with ".bpmn" extension using specification of BPMN 2.0 (OMG 2011). Java APIs such as Eclipse BPMN 2.0 Modeler and Camunda support reading and updating these models.

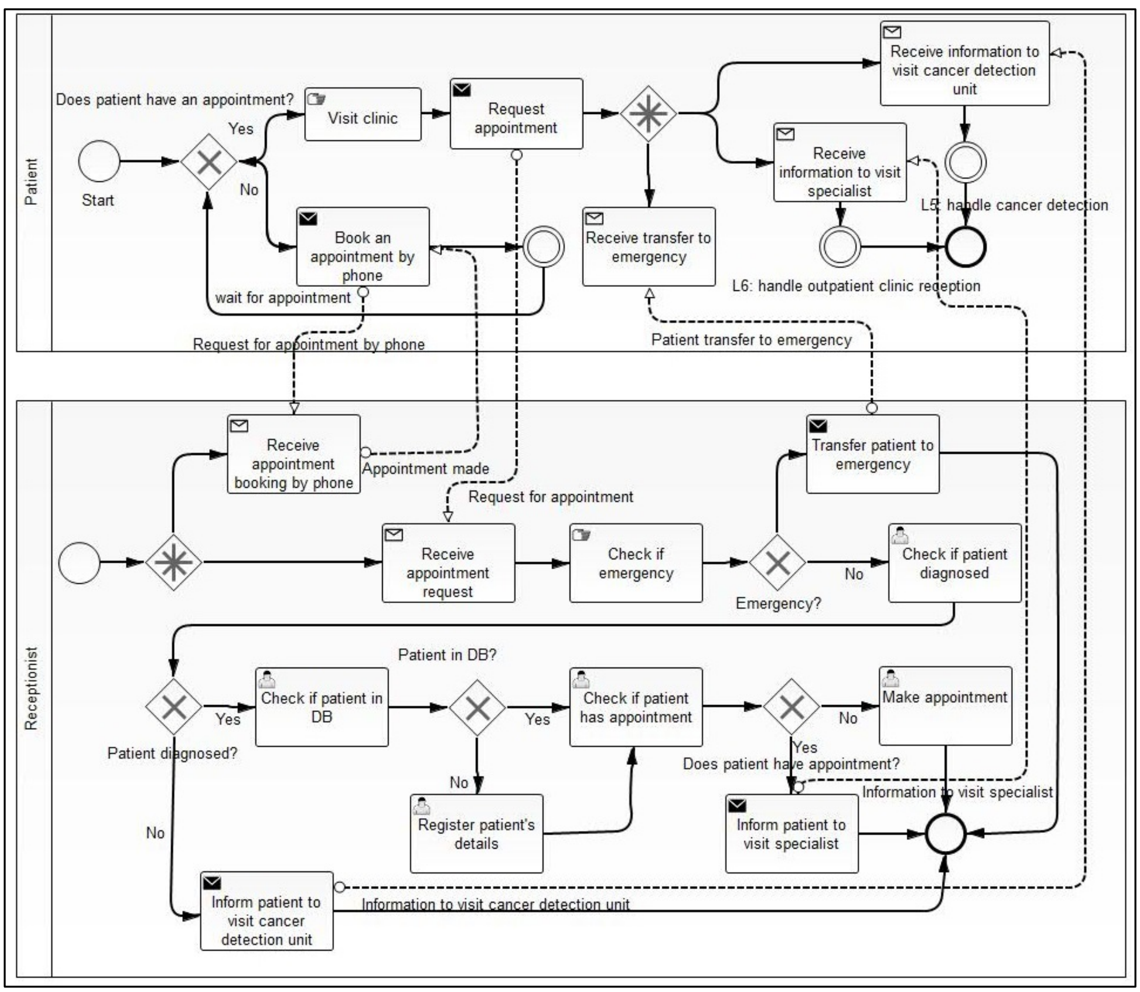

Figure A.1 BPMN model CP1: Handle Patient General Reception, adapted by (Ahmad, 2015). Used with author's permission.

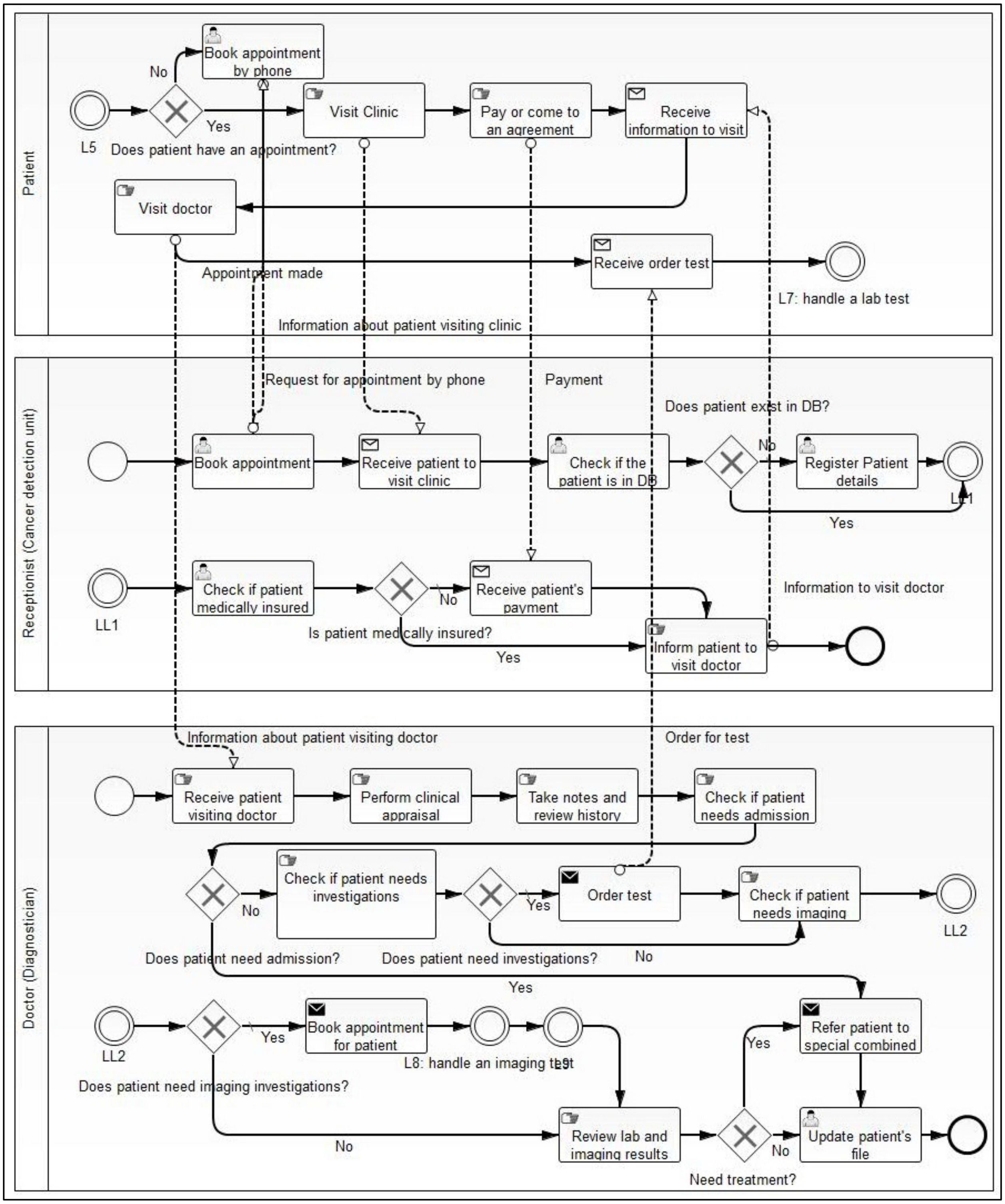

Figure A.2: BPMN CP2: Handle Cancer detection, adapted by (Ahmad, 2015). Used with author's permission.

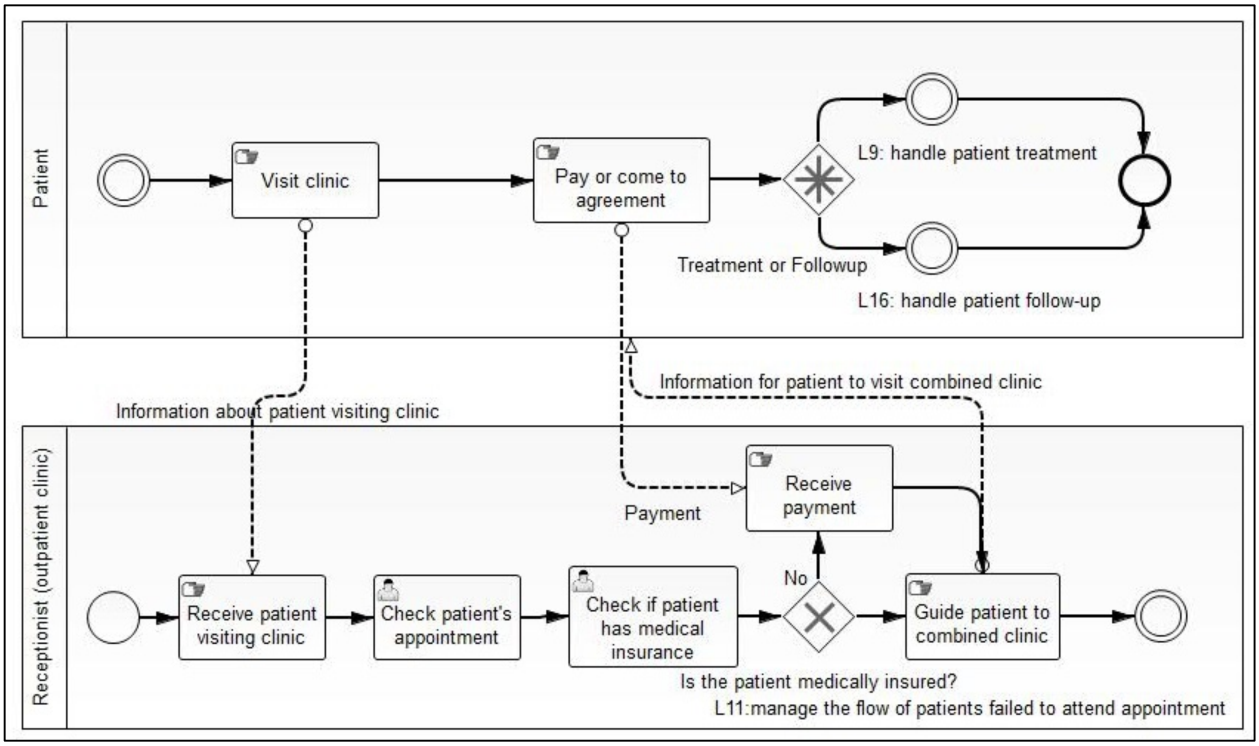

Figure A.3 : BPMN CP3: Handle Outpatient clinic reception, adapted by (Ahmad, 2015). Used with author's permission.

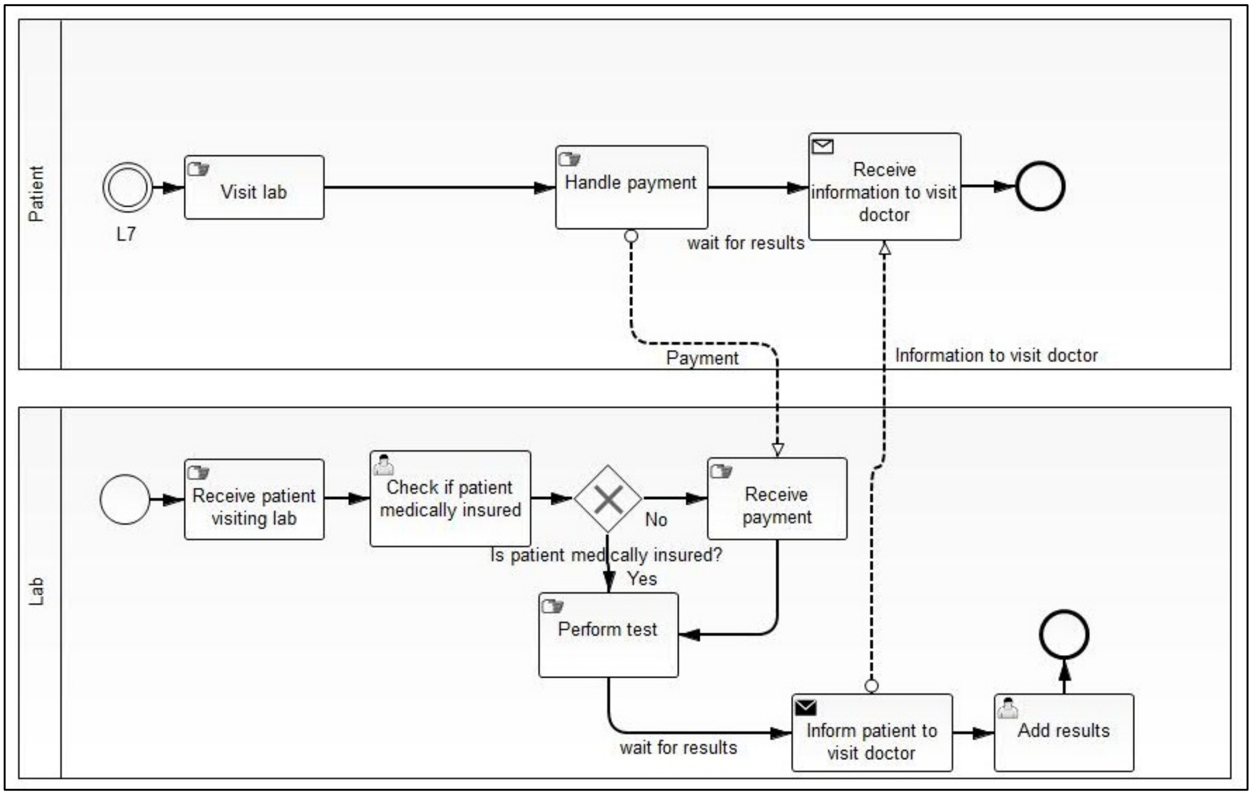

Figure A.4: BPMN CP4: Handle Lab test, adapted by (Ahmad, 2015). Used with author's permission.

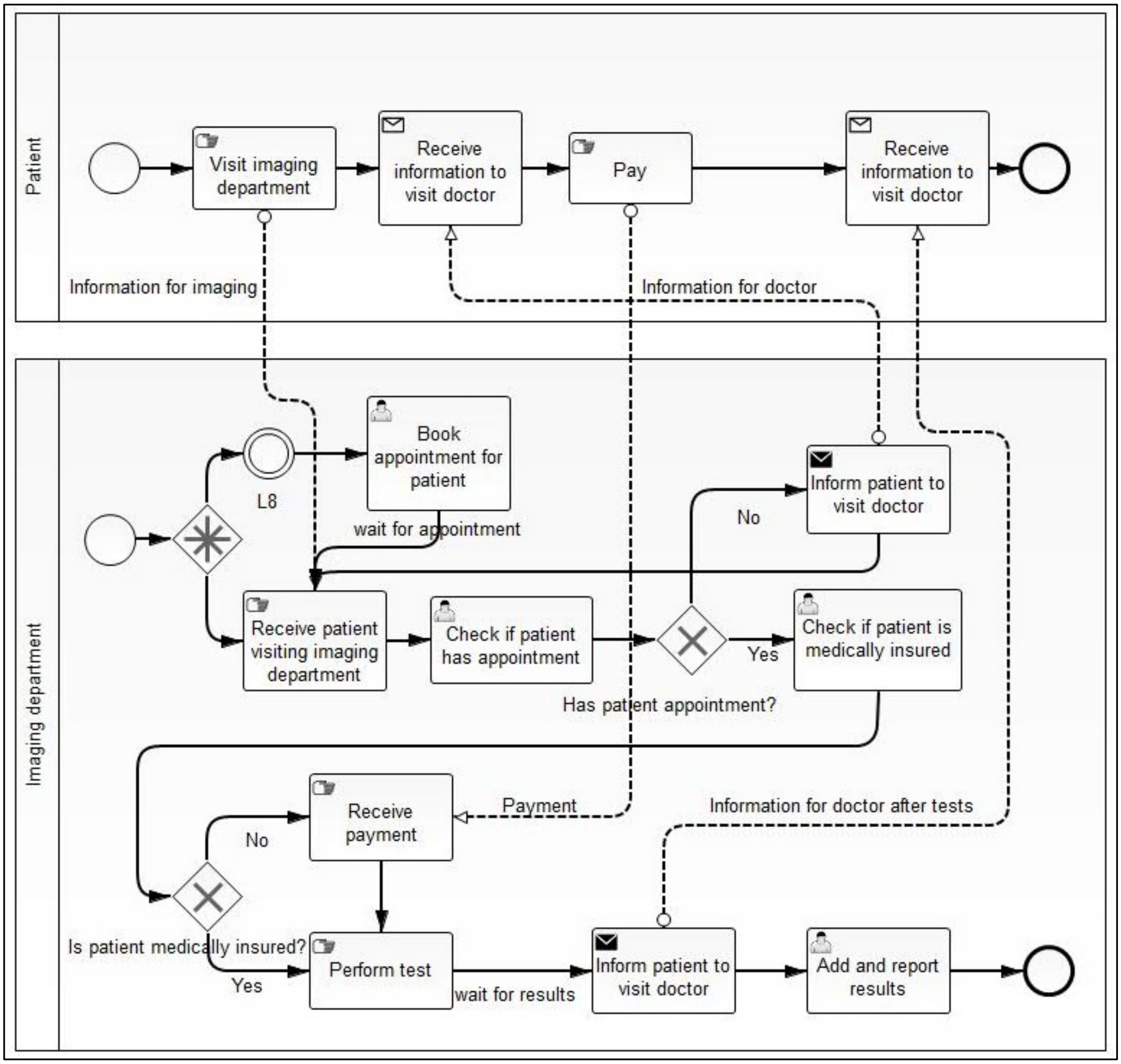

Figure A.5: BPMN CP5: Handle Imaging test, adapted by (Ahmad, 2015). Used with author's permission.

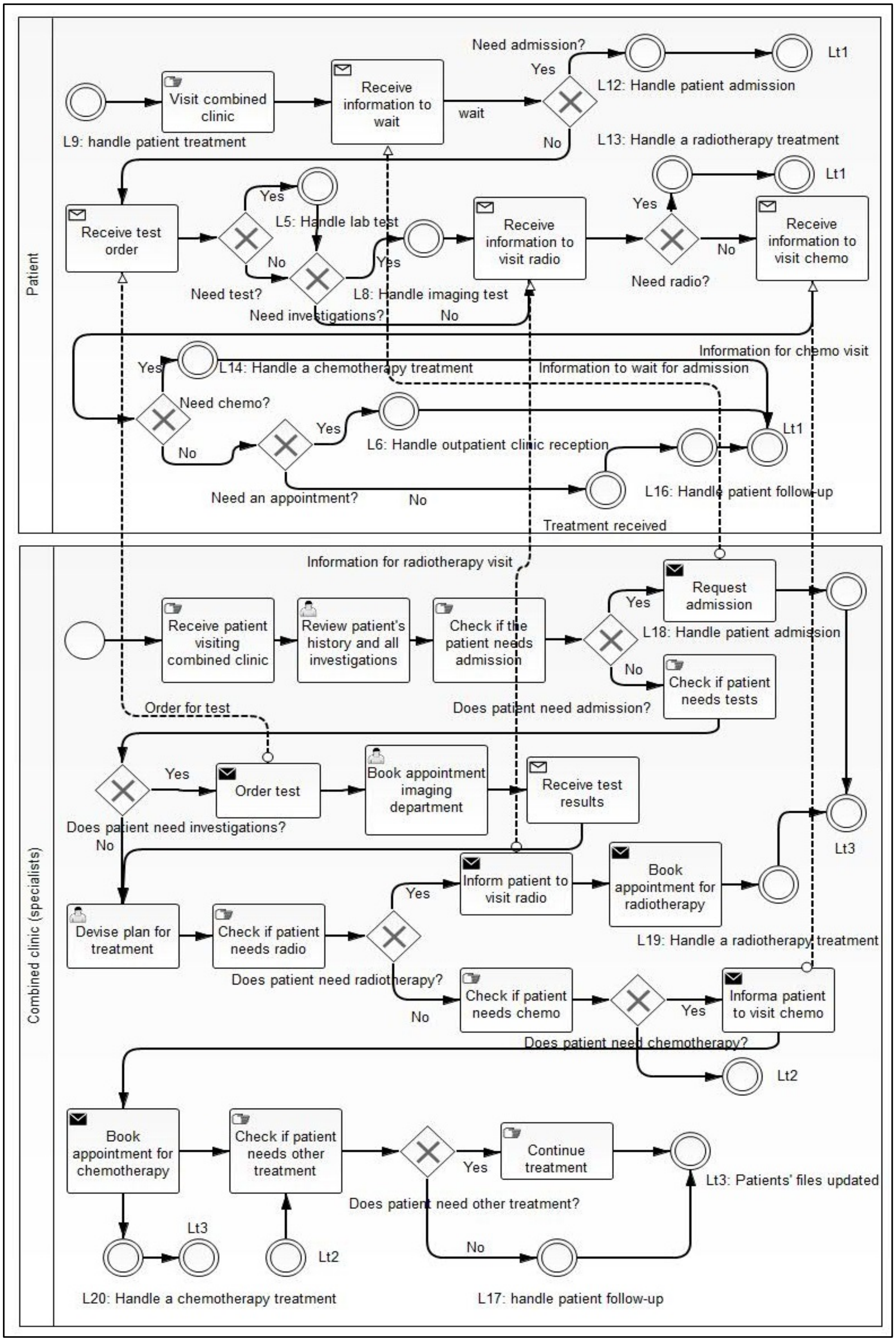

Figure A.6: BPMN CP6: Handle Patient treatment, adapted by (Ahmad, 2015). Used with author's permission.

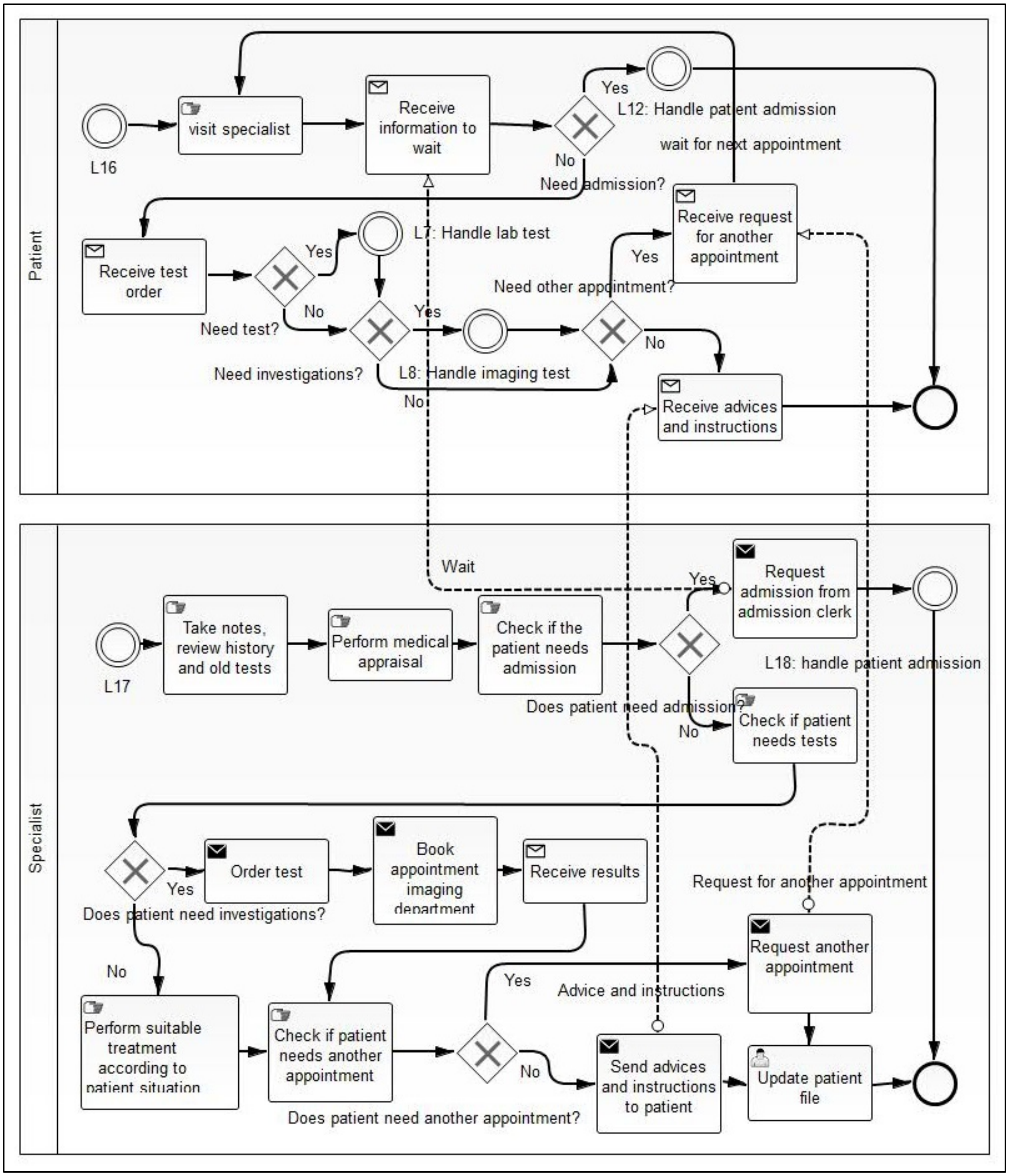

Figure A.7: BPMN CP7: Handle Patient follow-up, adapted by (Ahmad, 2015). Used with author's permission.

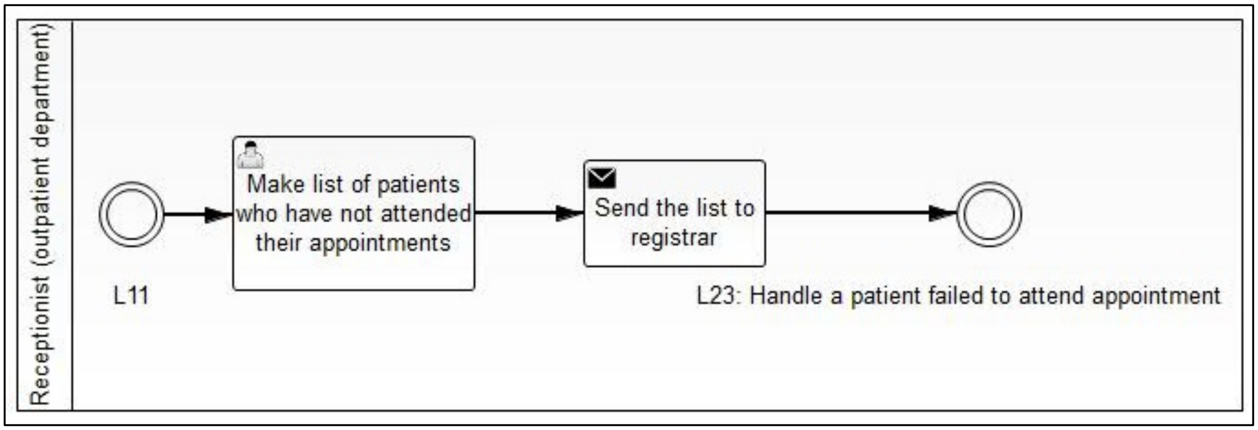

Figure A.8: BPMN CMP1: Manage the Flow of Patients fail to attend appointment, adapted by (Ahmad, 2015). Used with author's permission.

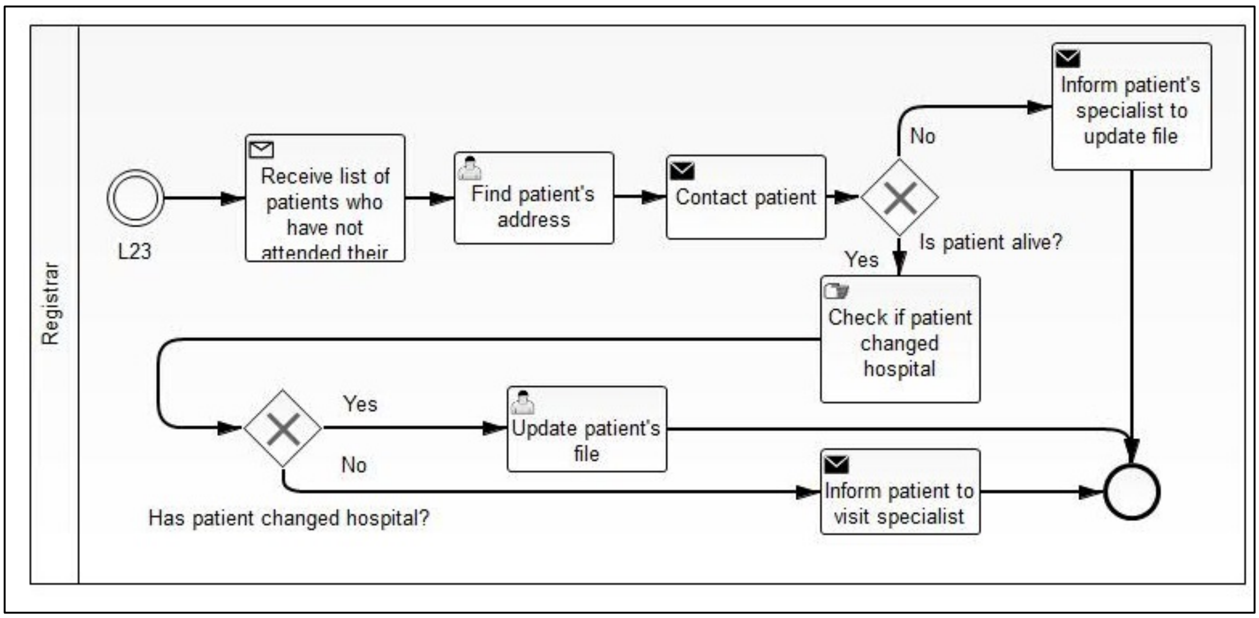

Figure A.9: BPMN CP8: Handle Patient fail to attend the appointment, adapted by (Ahmad, 2015). Used with author's permission.

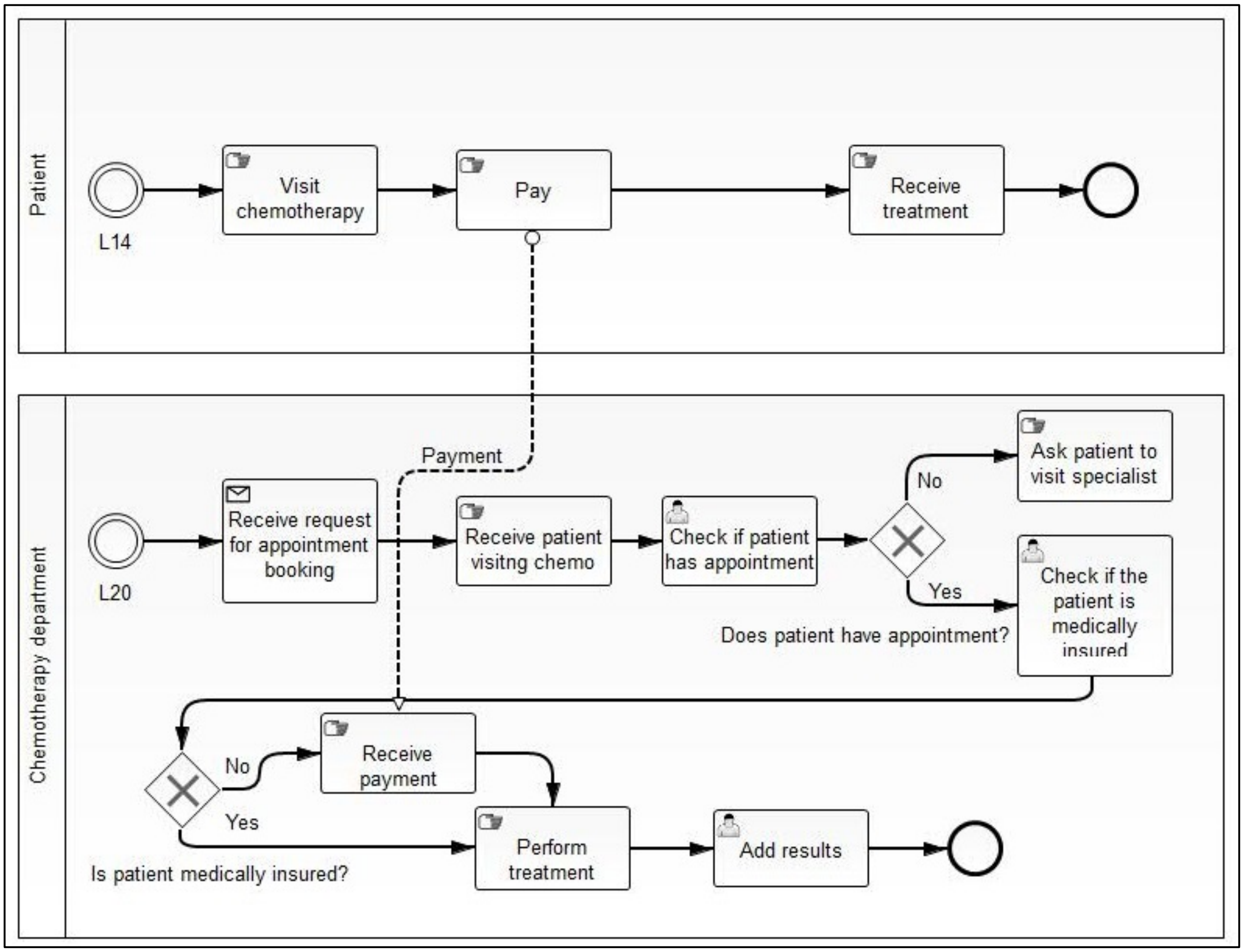

Figure A.10: BPMN CP9: Handle Chemotherapy treatment, adapted by (Ahmad, 2015). Used with author's permission.

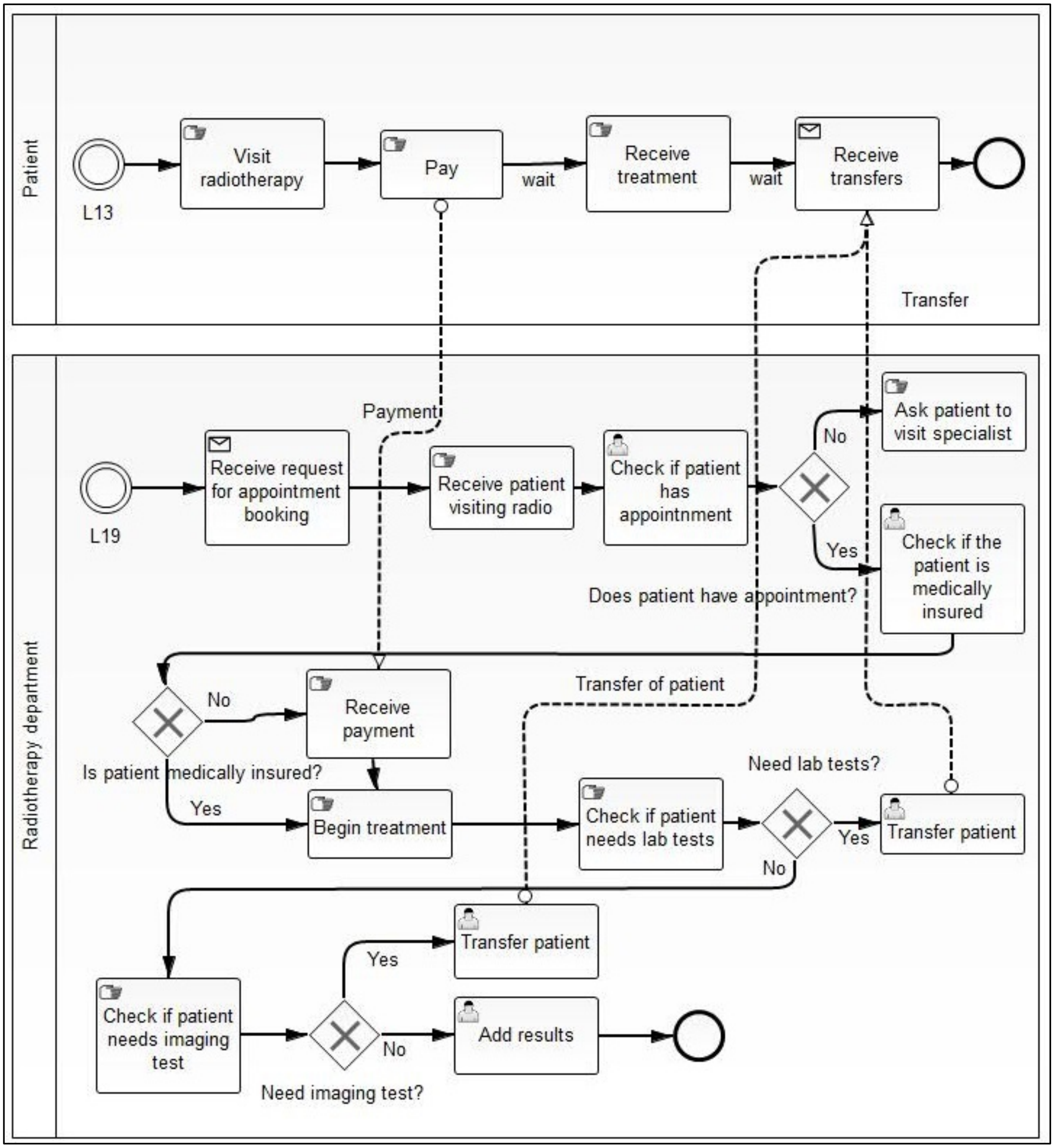

Figure A.11: BPMN CP10: Handle Radiotherapy treatment, adapted by (Ahmad, 2015). Used with author's permission.

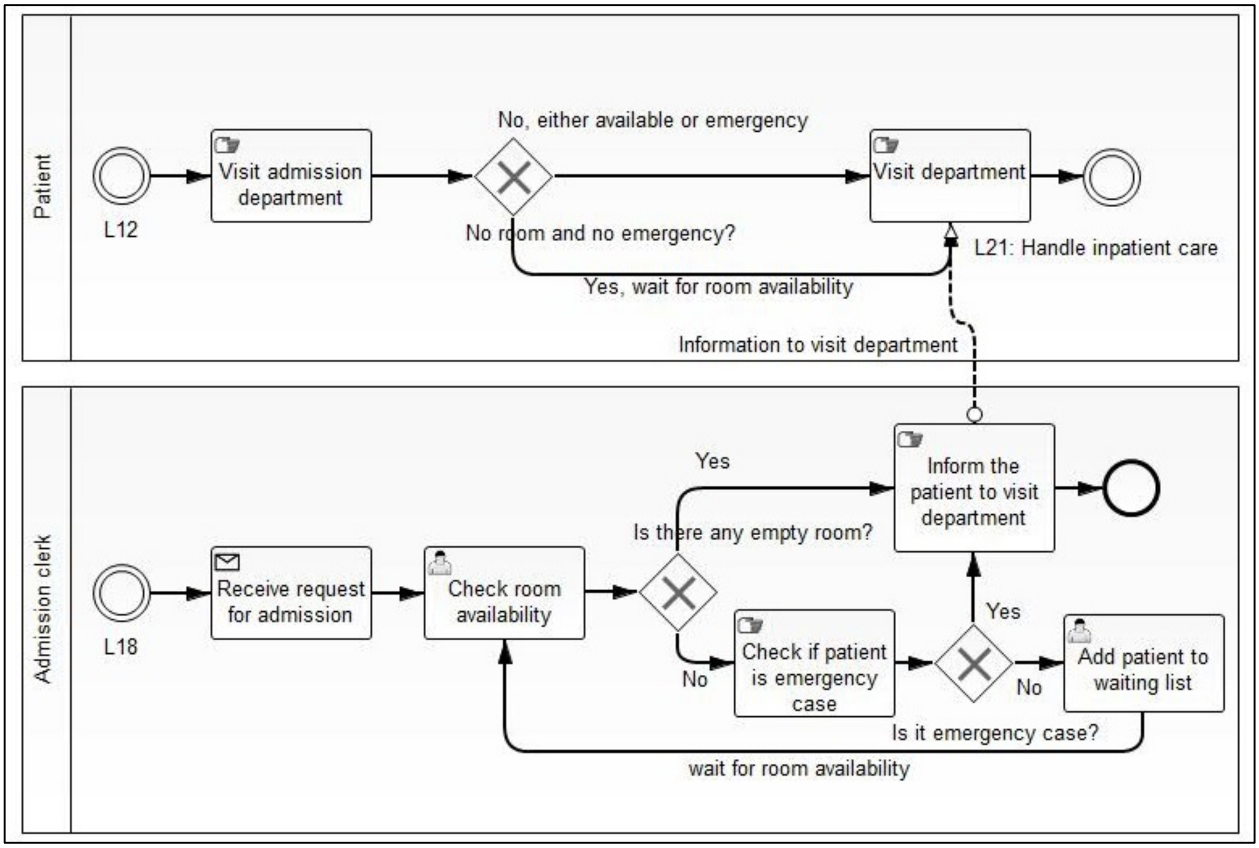

Figure A.12: BPMN CP11: Handle Patient admission, adapted by (Ahmad, 2015). Used with author's permission.

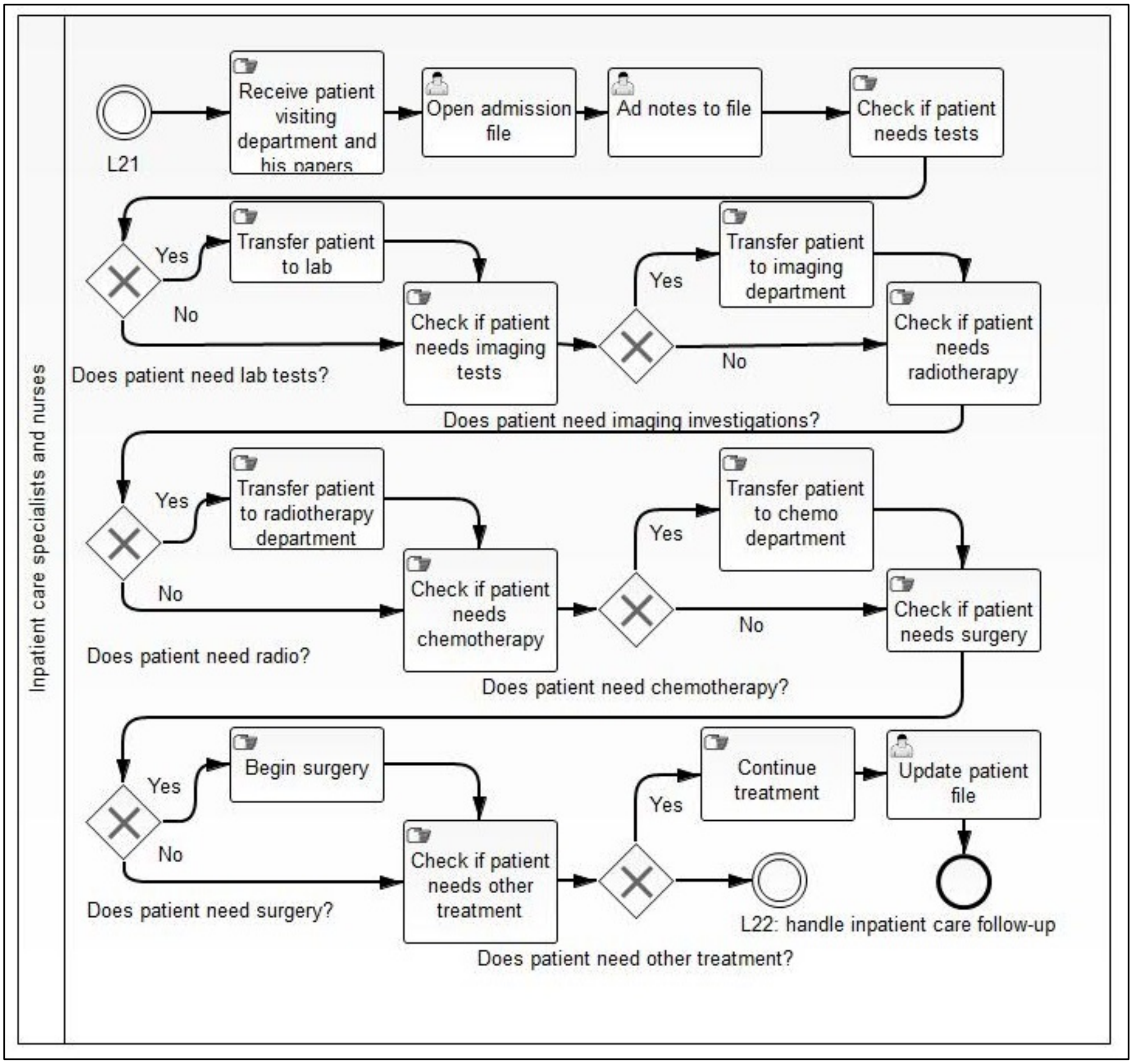

Figure A.13: BPMN CP12: Handle Inpatient care, adapted by (Ahmad, 2015). Used with author's permission.

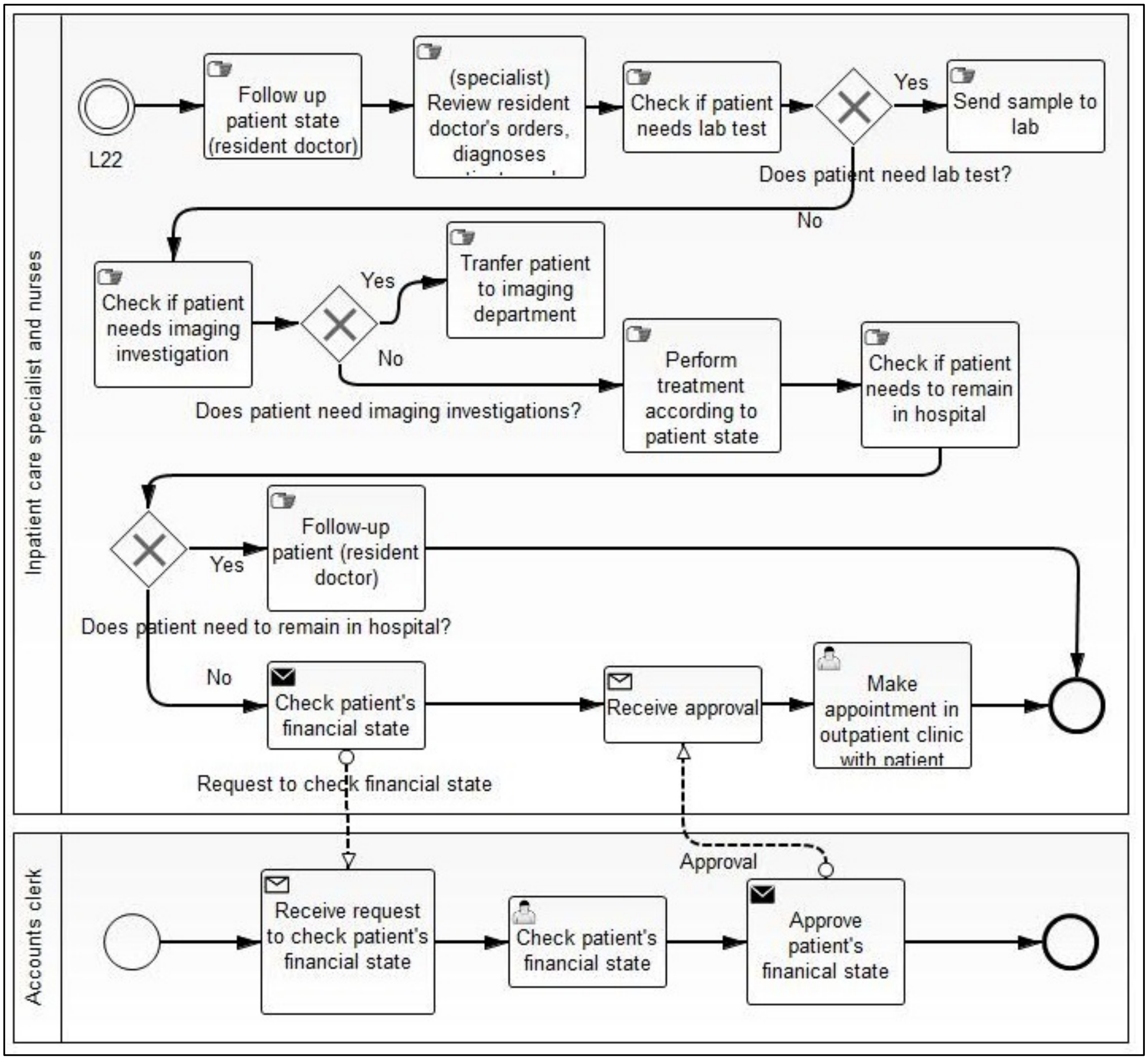

Figure A.14: BPMN CP13: Handle Inpatient follow-up, adapted by (Ahmad, 2015). Used with author's permission.

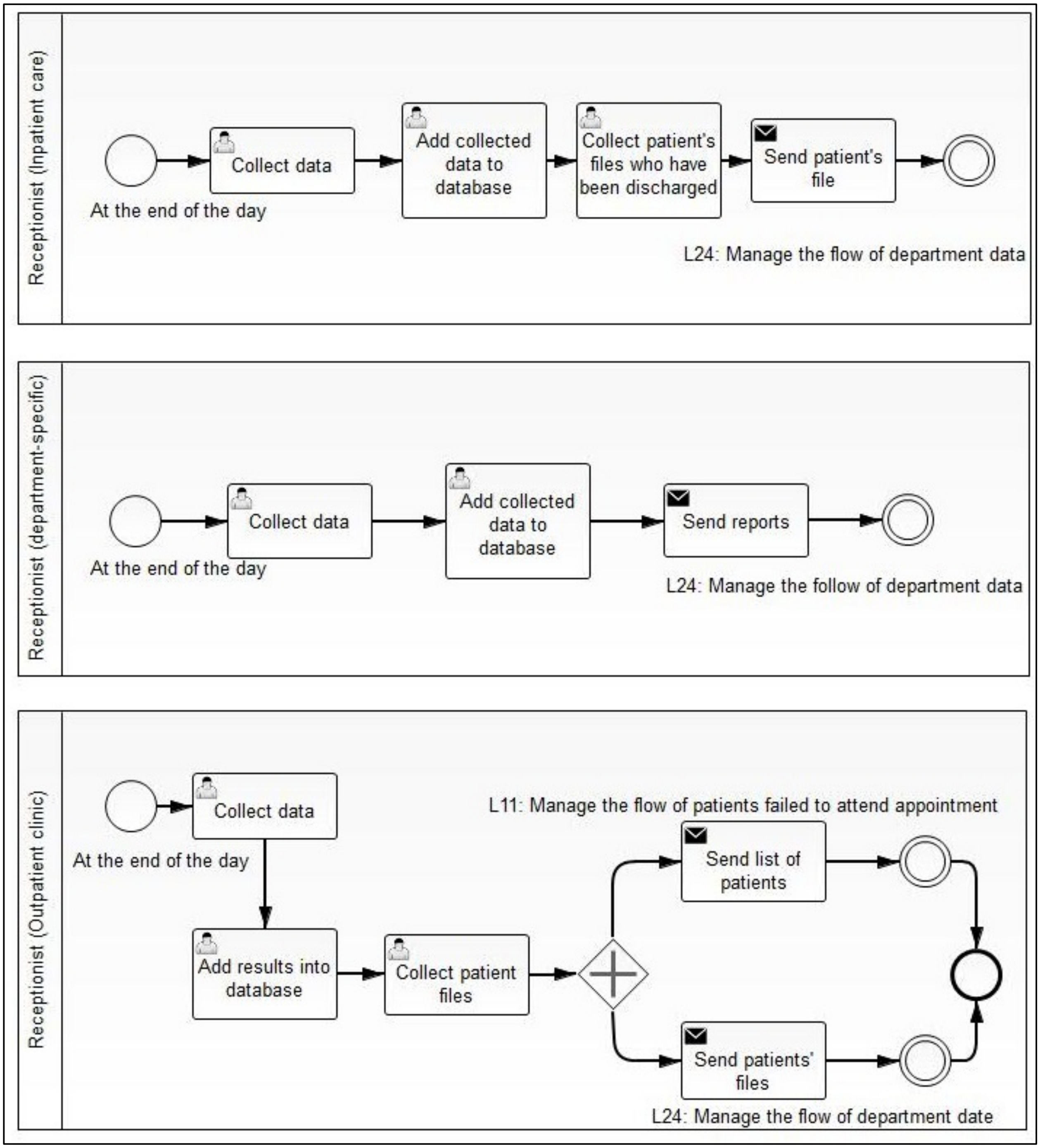

Figure A.15: BPMN CP14: Handle End of day data, adapted by (Ahmad, 2015). Used with author's permission.

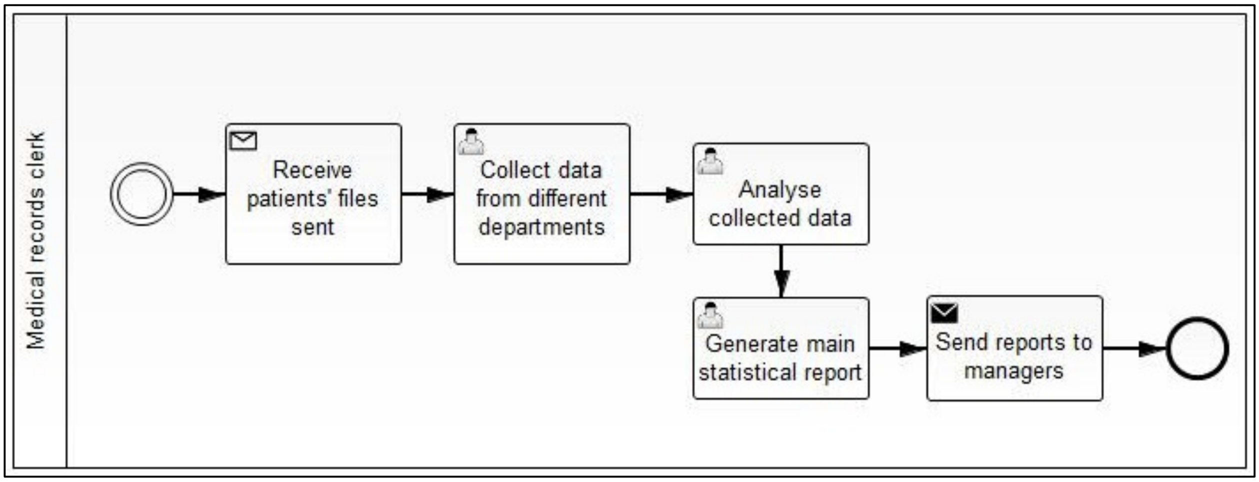

Figure A.16: BPMN CMP2: Manage the Flow of End of day data, adapted by (Ahmad, 2015). Used with author's permission.

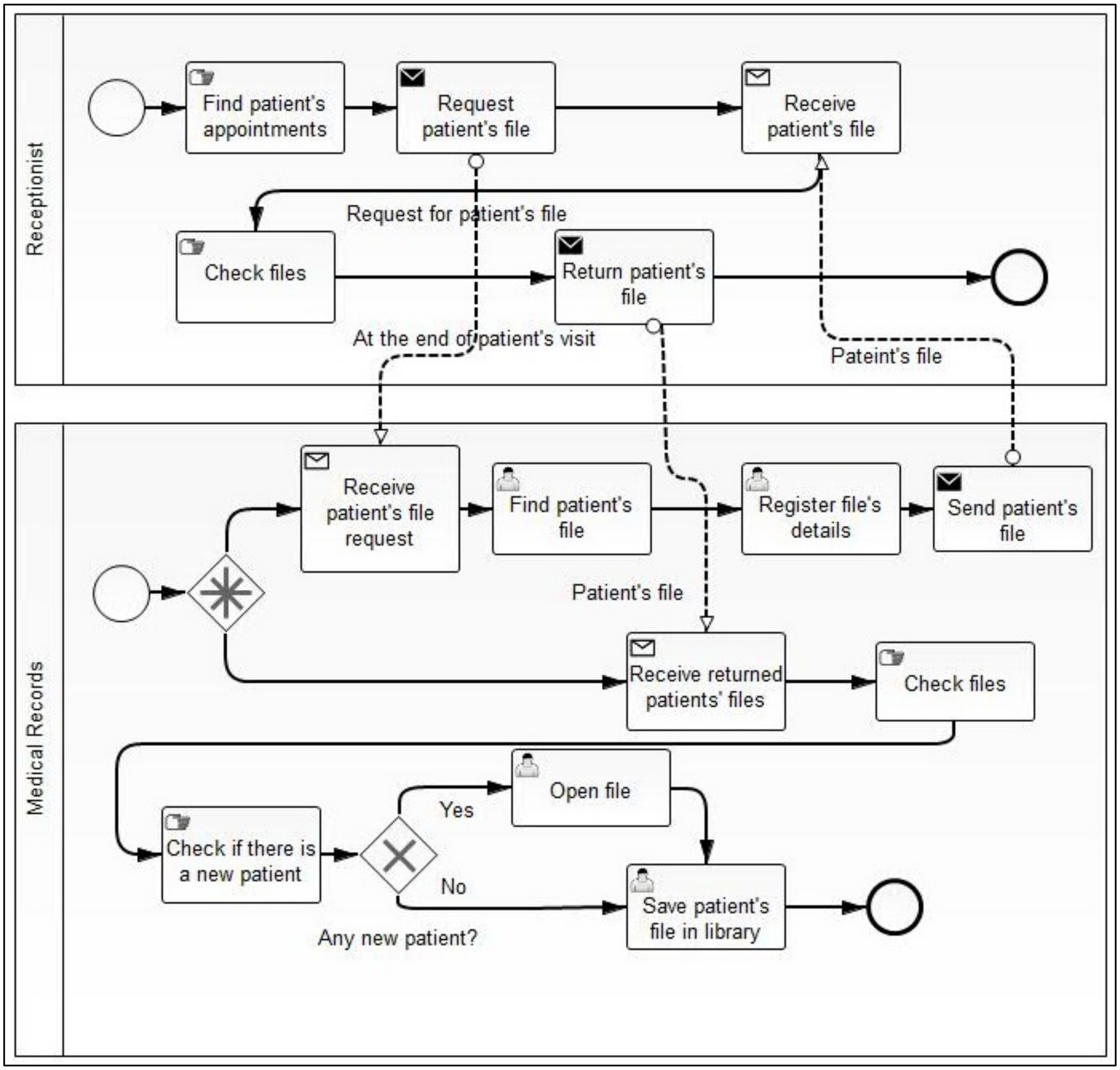

Figure A.17: BPMN CP15: Handle Medical records, adapted by (Ahmad, 2015). Used with author's permission.

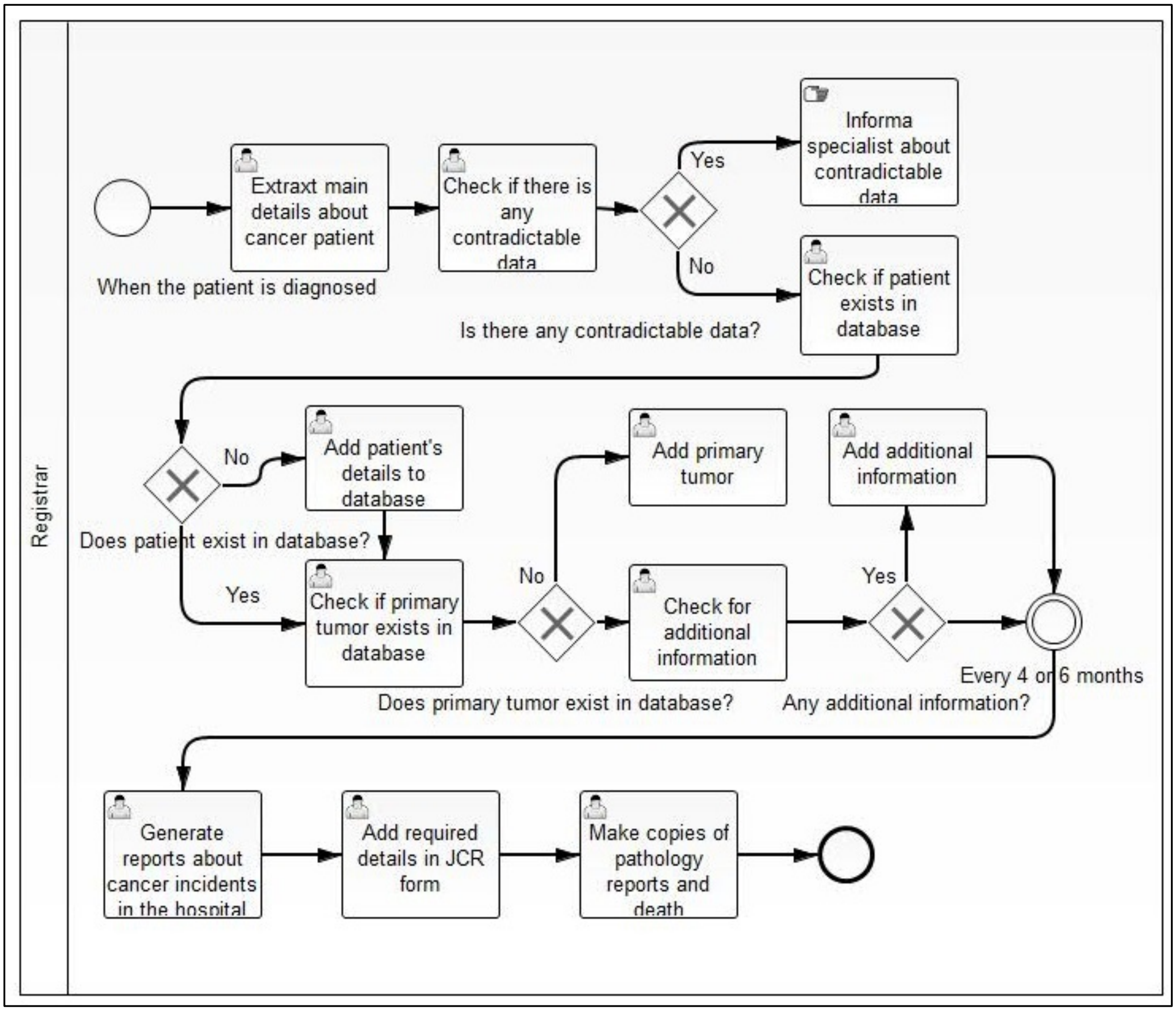

Figure A.18: BPMN CP16: Handle Hospital registration, adapted by (Ahmad, 2015). Used with author's permission.

Appendix B

Derived Search Space Elements CCR

Table B.1: Search Space Elements (SSEs) Derived from CCR Case Study

SSE ID	SSE Name/Role	Tasks and Activities	Remarks
SSE 1	$Receptionist_5$ [Handle a Patient General Reception]	• Request appointment	
SSE 2	$Receptionist_6$ [Handle a Patient General Reception]	• Book an appointment by phone • Visit clinic	
SSE 3	$Patient_9$	• Pay or come to an agreement • Visit doctor • Book appointment by phone • Visit Clinic	Interact with doctor
SSE 4	$Receptionist_9$ (Cancer detection unit)	• Inform patient to visit doctor • Check if patient medically insured • Register Patient details • Book appointment • Check if the patient is in DB	
SSE 5	Doctor (Diagnostician)	• Check if patient needs admission • Receive patient visiting doctor • Refer patient to special combined clinic • Review lab and imaging results • Perform clinical appraisal • Check if patient needs imaging investigations • Book appointment for patient • Take notes and review history • Check if patient needs investigations • Order test • Update patient's file	
SSE 6	$Patient_4$	• Visit clinic • Pay or come to agreement	Interact with clinic
SSE 7	$Receptionist_3$ (outpatient clinic)	• Guide patient to combined clinic • Check if patient has medical insurance • Receive payment • Receive patient visiting clinic • Check patient's appointment	
SSE 8	$Patient_8$	• Handle payment • Visit lab	Interact with lab
SSE 9	Lab	• Receive patient visiting lab	

		• Inform patient to visit doctor • Perform test • Receive payment • Check if patient medically insured • Add results	
SSE 10	Patient_1	• Pay • Visit imaging department	Interact with imaging department
SSE 11	Imaging department	• Inform patient to visit doctor • Inform patient to visit doctor • Add and report results • Perform test • Receive payment • Receive patient visiting imaging department • Check if patient is medically insured • Check if patient has appointment • Book appointment for patient	
SSE 12	Patient_6	• Visit combined clinic	Interact with combined clinic
SSE 13	Combined clinic (specialists)	• Check if patient needs tests • Order test • Check if the patient needs admission • Inform patient to visit radio • Receive patient visiting combined clinic • Book appointment for radiotherapy • Inform a patient to visit chemo • Book appointment imaging department • Continue treatment • Review patient's history and all investigations • Check if patient needs other treatment • Check if patient needs chemo • Request admission • Check if patient needs radio • Book appointment for chemotherapy • Devise plan for treatment	
SSE 14	Patient_5	• Receive information to wait • Visit specialist	Interact with specialist
SSE 15	Specialist	• Perform suitable treatment according to patient situation • Check if patient needs tests • Check if the patient needs admission • Perform medical appraisal	

		• Book appointment imaging department • Request another appointment • Send advices and instructions to patient • Check if patient needs another appointment • Take notes, review history and old tests • Update patient file • Request admission from admission clerk • Order test	
SSE 16	$Registrar_2$	• Inform patient's specialist to update file • Inform patient to visit specialist • Check if patient changed hospital • Contact patient • Update patient's file • Find patient's address •	Interact with patient and specialist
SSE 17	$Patient_7$	• Pay • Visit chemotherapy • Receive treatment •	Interact with chemotherapy
SSE 18	Chemotherapy department	• Perform treatment • Receive payment • Receive patient visiting chemo • Ask patient to visit specialist • Add results • Check if the patient is medically insured • Check if patient has appointment	
SSE 19	$Patient_3$	• Pay • Visit radiotherapy • Receive treatment	Interact with radiotherapy
SSE 20	Radiotherapy department	• Receive payment • Transfer patient • Ask patient to visit specialist • Transfer patient • Receive patient visiting radio • Check if the patient is medically insured • Check if patient has appointment • Begin treatment • Add results • Check if patient needs imaging test • Check if patient needs lab tests	

SSE 21	$Patient_2$	• Visit department • Visit admission department	Interact with admission department
SSE 22	$Clerk_1$ [Admission clerk]	• Check if patient is emergency case • Inform the patient to visit department • Add patient to waiting list • Check room availability	
SSE 23	Inpatient care specialists and nurses	• Begin surgery • Check if patient needs surgery • Transfer patient to chemo department • Check if patient needs chemotherapy • Check if patient needs imaging tests • Transfer patient to lab • Check if patient needs tests • Continue treatment • Receive patient visiting department and his papers • Check if patient needs other treatment • Open admission file • Transfer patient to radiotherapy department • Check if patient needs radiotherapy • Transfer patient to imaging department • Update patient file • Add notes to file	
SSE 24	Inpatient care specialist and nurses	• Follow-up patient (resident doctor) • Make appointment in outpatient clinic with patient • Check if patient needs to remain in hospital • Perform treatment according to patient state • Check patient's financial state • Transfer patient to imaging department • Follow up patient state (resident doctor) • Check if patient needs imaging investigation • Send sample to lab • Check if patient needs lab test • (specialist) Review resident doctor's orders, diagnoses patients and review old tests	

SSE 25	$Clerk_2$ [Accounts clerk]	• Approve patient's financial state • Check patient's financial state •	
SSE 26	$Receptionist_2$ (Inpatient care)	• Collect patient's files who have been discharged • Add collected data to database • Collect data • Send patient's file	
SSE 27	$Receptionist_1$ (department-specific)	• Add collected data to database • Collect data • Send reports	
SSE 28	$Receptionist_8$ (Outpatient clinic)	• Send patients' files • Add results into database • Collect data • Send list of patients • Collect patient files	
SSE 29	$Receptionist_4$	• Request patient's file • Check files • Return patient's file • Find patient's appointments	
SSE 30	Medical Records	• Register file's details • Find patient's file • Save patient's file in library • Send patient's file • Open file • Check files • Check if there is a new patient	
SSE 31	$Registrar_1$	• Add additional information • Check for additional information • Add primary tumor • Extract main details about cancer patient • Make copies of pathology reports and death certificates • Add required details in JCR form • Generate reports about cancer incidents in the hospital • Check if primary tumor exists in database • Add patient's details to database • Check if patient exists in database • Check if there is any contradictable data • Inform a specialist about contradictable data	Handle updates on Jordan Cancer Registry (JCR) database.
SSE 32	$Receptionist_7$ (outpatient department)	• Transfer patient to emergency • Inform patient to visit cancer detection unit	

		• Check if patient diagnosed • Inform patient to visit specialist • Check if patient has appointment • Check if patient in DB • Register patient's details • Make appointment • Check if emergency • Send the list to registrar • Make list of patients who have not attended their appointments	
SSE 33	$Clerk_3$ [Medical records clerk]	• Send reports to managers • Generate main statistical report • Analyse collected data • Collect data from different departments	

Appendix C
Results of the First Experiment

Solutions by domain experts at the King Hussein Cancer Centre (KHCC)

Table C.1: Domain expert solution 1

Manual Solution 1				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	1 2 6 7 16 32	.25	.083	.4165
Service 2	12 13 14 15 17 18 19 20 33	.25	0	.625
Service 3	21 22 23 24 25	.25	.111	.4305
Service 4	3 4 5 26 27 28 29 30 31	.25	.167	.4585
Service 5	8 9 10 11	0	.333	.6665
Total		**.15**	**.189**	**.5195**

Table C.2: Domain expert solution 2

Manual Solution 2				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	1 2 3 4 5 6 7 16 32	0	0	.5
Service 2	21 22 23 24 25 26 27 28 29 30	0	.125	.5625
Service 3	8 9 10 11 17 18 19 20	.167	.078	.4555
Service 4	31 33	.25	.125	.4375
Service 5	12 13 14 15	.556	.25	.347
Total		**.195**	**.116**	**.4605**

Table C.3: Domain expert solution 3

Manual Solution 3				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	1 2 3 4 5 29 30	0	0	0.5

	32			
Service 2	6 7 16 21 22	0	0.25	0.625
Service 3	17 18 19 20	0.444	0.1	0.328
Service 4	31 33	0.2	0.143	0.472
Service 5	12 13 14 15	0.25	0.1	0.425
Service 6	8 9	0	1	1
Service 7	10 11	0	1	1
Service 8	23 24 25 26 27 28	0.556	0.25	0.347
Total		**0.181**	**0.355**	**0.587**

Table C.4: Domain expert solution 4

Solution 4				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	29 30 32	0	0	0.5
Service 2	6 7 10 11	0.125	0.119	0.497
Service 3	17 18 19 20 21 22	0.429	0.25	0.411

	23			
Service 4	31 33	0.25	0.333	0.542
Service 5	12 13	0.167	0.3	0.567
Service 6	8 9	0.286	0.1	0.407
Service 7	14 15 16	0.4	0.167	0.384
Service 8	24 25 26 27	0	1	1
Service 9	1 2 3 4 5	0.667	0.5	0.417
Total		**0.258**	**0.308**	**0.525**

Table C.5: Domain expert solution 5

Solution 5				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	17 18 19 20 29 30 32	0.2	0.083	0.446
Service 2	6 7 10 11 14 15 16	0364	0.119	0.378
Service 3	21 22 31 33	0.119	0.125	0.497
Service 4	3 4 5 12 13 23 24	0	0.333	0.667

	25 26 27 28			
Service 5	1 2 8 9	0.353	0.082	0.365
Total		**0.208**	**0.147**	**0.47**

Table C.6: Domain expert solution 6

Solution 6				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	6 7 10 11 19 20 32	0.25	0.167	0.459
Service 2	3 4 5 14 15 16 17 18 29 30	0.231	0.089	0.429
Service 3	21 22 33	0.3	0.143	0.422
Service 4	12 13 24 25 26 27 28	0	0.333	0.667
Service 5	23 31	0333	0	0.334
Service 6	1 2 8 9	0.462	0.071	0.305
Total		**0.263**	**0.134**	**0.436**

Table C.7: Domain expert solution 7

Solution 7				
Service	SSE ID	Coupling	Cohesion	Fitness
Service 1	6 7 10 11 19 20 32	0.125	0.119	0.497
Service 2	14 15 21 22 23 33	0.111	0.143	0.516
Service 3	3 4 5 24 25 29 30 31	0.143	0.167	0.512
Service 4	1 2 8 9 26 27 28	0.3	0.143	0.422
Service 5	16 17 18	0.667	0.5	0.417
Service 6	12 13	0	0.167	0.584
Total		**0.224**	**0.207**	**0.492**

Appendix D
Material for the First Experiment

This appendix presents the material used to present the project and case study to domain experts at King Hussein Cancer Center such that they can participate in deriving software services from the business process models of KHCC.

Table D.1: Exercise objectives and estimated time

Exercise	Objectives	Time
1. Warming up	1. **Understand** the business process 2. **Understand** the goal of automation	5 minutes of discussion. 2 minutes/group to talk about their findings.
2. Prepare search space	1. Classify data elements that can be automated or not. 2. Compare the outputs when using **inter-role and intra-role** elements 3. Prepare the participants to build more understanding for the BPMs of KHCC (CCR case study) **(Priori interaction)**	15 minutes (discussion) + 2-3 minutes/group
3. Manual services	1. Produce **candidate services manually** (in order to **compare** them to the automated services).	15-20 mins
4. Evaluation	1. Evaluate services produced by search engine without interaction. **(posteriori interaction)** 3. Interact with the search engine itself to steer the trajectory of the search. **(Interactive search)**	30-40 mins

Exercise 1

In groups: Write a process that can be automated to help patients mainly to do a specific procedure faster or easier.

Time:

- 5 minutes (Discussion)
- 2 minutes/group (Communication)

Process to be automated	Example

Exercise 2:

- According to the list of processes, show the following:

1. Show what processes can be automated (How?). What processes cannot be automated at all.
2. Link the activities that **must** work together. (Splitting them makes no sense). Put them in brackets.
    - Time: 15 minutes (discussion) + 2 minutes/group (communication)

	Number	Notes
Can be automated	e.g., (1+2), 5,	
Cannot be automated		

Exercise 3: Create components out the functions' list.

Component Name	Functions

Exercise 4: Evaluate the following candidate services

Table D.2: Solution 1

Service	**Functions**	**Evaluation**
Service 1: Medical Records[Process_30] **+**	(1)Register file's details (2)Find patient's file (3)Save patient's file in library (4)Send patient's file (5)Open file (6)Check files (7)Check if there is a new patient	

Receptionist[Process_29] **+** Receptionist (department-specific)[Process_27]	(8)Request patient's file (9)Check files (10)Return patient's file (11)Find patient's appointments (12)Add collected data to database (13)Collect data (14)Send reports	
Service 2 Registrar[Process_31]	(1)Add additional information (2)Check for additional information (3)Add primary tumor (4)Extract main details about cancer patient (5)Make copies of pathology reports and death certificates (6)Add required details in JCR form (7)Generate reports about cancer incidents in the hospital (8)Check if primary tumor exists in database (9)Add patient's details to database (10)Check if patient exists in database (11)Check if there is any contradictable data (12)Informal specialist about contradictable data	
Service 3 Patient[Process_21] **+** Combined clinic (specialists)[Process_13] **+** Admission clerk[Process_22] **+**	(1)Visit department (2)Visit admission department Patient[Process_14] (3)Receive information to wait (4)visit specialist (5)Check if patient needs tests (6)Order test (7)Check if the patient needs admission (8)Inform patient to visit radio (9)Receive patient visiting combined clinic (10)Book appointment for radiotherapy (11)Informal patient to visit chemo (12)Book appointment imaging department (13)Continue treatment (14)Review patient's history and all investigations (15)Check if patient needs other treatment (16)Check if patient needs chemo (17)Request admission (18)Check if patient needs radio (19)Book appointment for chemotherapy (20)Devise plan for treatment (21)Check if patient is emergency case (22)Inform the patient to visit department (23)Add patient to waiting list (24)Check room availability	

Specialist[Process_15]	(25)Perform suitable treatment according to patient situation (26)Check if patient needs tests (27)Check if the patient needs admission (28)Perform medical appraisal (29)Book appointment imaging department (30)Request another appointment (31)Send advices and instructions to patient (32)Check if patient needs another appointment (33)Take notes, review history and old tests (34)Update patient file (35)Request admission from admission clerk (36)Order test	
Service 4 Lab[Process_9] **+** Patient[Process_8]	 (1)Receive patient visiting lab (2)Inform patient to visit doctor (3)Perform test (4)Receive payment (5)Check if patient medically insured (6)Add results (7)Handle payment (8)Visit lab	
Service 5 Patient[Process_19] **+** Radiotherapy department[Process_20]	 (1)Pay (2)Visit radiotherapy (3)Receive treatment (4)Receive payment (5)Transfer patient (6)Ask patient to visit specialist (7)Transfer patient (8)Receive patient visiting radio (9)Check if the patient is medically insured (10)Check if patient has appointment (11)Begin treatment (12)Add results (13)Check if patient needs imaging test (14)Check if patient needs lab tests	
Service 6 Receptionist (outpatient clinic)[Process_7] **+** Patient[Process_6] **+** Patient[Process_12]	 (1)Guide patient to combined clinic (2)Check if patient has medical insurance (3)Receive payment (4)Receive patient visiting clinic (5)Check patient's appointment (6)Visit clinic (7)Pay or come to agreement (8)Visit combined clinic	

+ Receptionist (Inpatient care)[Process_26]	(9)Collect patient's files who have been discharged (10)Add collected data to database (11)Collect data (12)Send patient's file	
Service 7 Patient[Process_3] **+** Doctor (Diagnostician)[Process_5] **+** Receptionist (Cancer detection unit)[Process_4] **+** Medical records clerk[Process_33] **+** Chemotherapy department[Process_18] **+** Patient[Process_17]	(1)Pay or come to an agreement (2)Visit doctor (3)Book appointment by phone (4)Visit Clinic (5)Check if patient needs admission (6)Receive patient visiting doctor (7)Refer patient to special combined clinic (8)Review lab and imaging results (9)Perform clinical appraisal (10)Check if patient needs imaging investigations (11)Book appointment for patient (12)Take notes and review history (13)Check if patient needs investigations (14)Order test (15)Update patient's file (16)Inform patient to visit doctor (17)Check if patient medically insured (18)Register Patient details (19)Book appointment (20)Check if the patient is in DB (21)Send reports to managers (22)Generate main statistical report (23)Analyse collected data (24)Collect data from different departments (25)Perform treatment (26)Receive payment (27)Receive patient visiting chemo (28)Ask patient to visit specialist (29)Add results (30)Check if the patient is medically insured (31)Check if patient has appointment (32)Pay (33)Visit chemotherapy (34)Receive treatment	
Service 8 Handle a Patient General Reception[Process_1] **+**	(1)Request appointment	

Handle a Patient General Reception[Process_2]	(2)Book an appointment by phone (3)Visit clinic (4)Transfer patient to emergency (5)Inform patient to visit cancer detection unit (6)Check if patient diagnosed (7)Inform patient to visit specialist (8)Check if patient has appointment (9)Check if patient in DB (10)Register patient's details (11)Make appointment (12)Check if emergency	
Service 9 Receptionist (outpatient department)[Process_32] **+** Receptionist (Outpatient clinic)[Process_28] **+** Registrar[Process_16]	(1)Send the list to registrar (2)Make list of patients who have not attended their appointments (3)Send patients' files (4)Add results into database (5)Collect data (6)Send list of patients (7)Collect patient files (8)Inform patient's specialist to update file (9)Inform patient to visit specialist (10)Check if patient changed hospital (11)Contact patient (12)Update patient's file (13)Find patient's address	
Service 10 Accounts clerk[Process_25] **+** Inpatient care specialist and nurses[Process_24] **+** Inpatient care specialists and nurses[Process_23]	(1)Approve patient's financial state (2)Check patient's financial state (3)Follow-up patient (resident doctor) (4)Make appointment in outpatient clinic with patient (5)Check if patient needs to remain in hospital (6)Perform treatment according to patient state (7)Check patient's financial state (8)Transfer patient to imaging department (9)Follow up patient state (resident doctor) (10)Check if patient needs imaging investigation (11)Send sample to lab (12)Check if patient needs lab test (13)(specialist) Review resident doctor's orders, diagnoses patients and review old tests (14)Begin surgery (15)Check if patient needs surgery (16)Transfer patient to chemo department (17)Check if patient needs chemotherapy (18)Check if patient needs imaging tests (19)Transfer patient to lab	

	(20)Check if patient needs tests (21)Continue treatment (22)Receive patient visiting department and his papers (23)Check if patient needs other treatment (24)Open admission file (25)Transfer patient to radiotherapy department (26)Check if patient needs radiotherapy (27)Transfer patient to imaging department (28)Update patient file (29)Add notes to file	
Service 11 Patient[Process_10] **+** Imaging department[Process_11]	(1)Pay (2)Visit imaging department (3)Inform patient to visit doctor (4)Inform patient to visit doctor (5)Add and report results (6)Perform test (7)Receive payment (8)Receive patient visiting imaging department (9)Check if patient is medically insured (10)Check if patient has appointment (11)Book appointment for patient	

Table D.3: Solution 2

Service	**Functions**	**Evaluation**
Service 1: Patient[Process_10] + Registrar[Process_31] + Imaging department[Process_11]	(1)Pay (2)Visit imaging department (3)Add additional information (4)Check for additional information (5)Add primary tumor (6)Extract main details about cancer patient (7)Make copies of pathology reports and death certificates (8)Add required details in JCR form (9)Generate reports about cancer incidents in the hospital (10)Check if primary tumor exists in database (11)Add patient's details to database (12)Check if patient exists in database (13)Check if there is any contradictable data (14)Informal specialist about contradictable data (15)Inform patient to visit doctor (16)Inform patient to visit doctor (17)Add and report results (18)Perform test (19)Receive payment	

	(20)Receive patient visiting imaging department (21)Check if patient is medically insured (22)Check if patient has appointment (23)Book appointment for patient	
Service 2 Patient[Process_19] + Radiotherapy department[Process_20] +	(1)Pay (2)Visit radiotherapy (3)Receive treatment (4)Receive payment (5)Transfer patient (6)Ask patient to visit specialist (7)Transfer patient (8)Receive patient visiting radio (9)Check if the patient is medically insured (10)Check if patient has appointment (11)Begin treatment (12)Add results (13)Check if patient needs imaging test (14)Check if patient needs lab tests	
Service 3 Patient[Process_21] + Patient[Process_6] + Patient[Process_14] + Combined clinic (specialists)[Process_13] + Patient[Process_12]	(1)Visit department (2)Visit admission department Receptionist (outpatient clinic)[Process_7] (3)Guide patient to combined clinic (4)Check if patient has medical insurance (5)Receive payment (6)Receive patient visiting clinic (7)Check patient's appointment (8)Visit clinic (9)Pay or come to agreement (10)Receive information to wait (11)visit specialist (12)Check if patient needs tests (13)Order test (14)Check if the patient needs admission (15)Inform patient to visit radio (16)Receive patient visiting combined clinic (17)Book appointment for radiotherapy (18)Informal patient to visit chemo (19)Book appointment imaging department (20)Continue treatment (21)Review patient's history and all investigations (22)Check if patient needs other treatment (23)Check if patient needs chemo (24)Request admission (25)Check if patient needs radio (26)Book appointment for chemotherapy (27)Devise plan for treatment (28)Visit combined clinic	

+ Admission clerk[Process_22] + Chemotherapy department[Process_18] + Patient[Process_17] + Specialist[Process_15]	(29)Check if patient is emergency case (30)Inform the patient to visit department (31)Add patient to waiting list (32)Check room availability (33)Perform treatment (34)Receive payment (35)Receive patient visiting chemo (36)Ask patient to visit specialist (37)Add results (38)Check if the patient is medically insured (39)Check if patient has appointment (40)Pay (41)Visit chemotherapy (42)Receive treatment (43)Perform suitable treatment according to patient situation (44)Check if patient needs tests (45)Check if the patient needs admission (46)Perform medical appraisal (47)Book appointment imaging department (48)Request another appointment (49)Send advices and instructions to patient (50)Check if patient needs another appointment (51)Take notes, review history and old tests (52)Update patient file (53)Request admission from admission clerk (54)Order test	
Service 4 Handle a Patient General Reception[Process_1] + Handle a Patient General Reception[Process_2] + Receptionist (outpatient department)[Process_32] + Receptionist (Outpatient clinic)[Process_28]	(1)Request appointment (2)Book an appointment by phone (3)Visit clinic (4)Transfer patient to emergency (5)Inform patient to visit cancer detection unit (6)Check if patient diagnosed (7)Inform patient to visit specialist (8)Check if patient has appointment (9)Check if patient in DB (10)Register patient's details (11)Make appointment (12)Check if emergency (13)Send the list to registrar (14)Make list of patients who have not attended their appointments (15)Send patients' files (16)Add results into database (17)Collect data	

+ Registrar[Process_16]	(18)Send list of patients (19)Collect patient files (20)Inform patient's specialist to update file (21)Inform patient to visit specialist (22)Check if patient changed hospital (23)Contact patient (24)Update patient's file (25)Find patient's address	
Service 5 Medical Records[Process_30] + Receptionist[Process_29]	 (1)Register file's details (2)Find patient's file (3)Save patient's file in library (4)Send patient's file (5)Open file (6)Check files (7)Check if there is a new patient (8)Request patient's file (9)Check files (10)Return patient's file (11)Find patient's appointments	
Service 6 Lab[Process_9] + Patient[Process_8]	 (1)Receive patient visiting lab (2)Inform patient to visit doctor (3)Perform test (4)Receive payment (5)Check if patient medically insured (6)Add results (7)Handle payment (8)Visit lab	
Service 7 Accounts clerk[Process_25] + Inpatient care specialist and nurses[Process_24] + Inpatient care specialists and nurses[Process_23] +	 (1)Approve patient's financial state (2)Check patient's financial state (3)Follow-up patient (resident doctor) (4)Make appointment in outpatient clinic with patient (5)Check if patient needs to remain in hospital (6)Perform treatment according to patient state (7)Check patient's financial state (8)Transfer patient to imaging department (9)Follow up patient state (resident doctor) (10)Check if patient needs imaging investigation (11)Send sample to lab (12)Check if patient needs lab test (13)(specialist) Review resident doctor's orders, diagnoses patients and review old tests (14)Begin surgery (15)Check if patient needs surgery	

              Medical records clerk[Process_33]	(16)Transfer patient to chemo department (17)Check if patient needs chemotherapy (18)Check if patient needs imaging tests (19)Transfer patient to lab (20)Check if patient needs tests (21)Continue treatment (22)Receive patient visiting department and his papers (23)Check if patient needs other treatment (24)Open admission file (25)Transfer patient to radiotherapy department (26)Check if patient needs radiotherapy (27)Transfer patient to imaging department (28)Update patient file (29)Add notes to file (30)Send reports to managers (31)Generate main statistical report (32)Analyse collected data (33)Collect data from different departments	
Service 8 Receptionist (department-specific)[Process_27] + Receptionist (Inpatient care)[Process_26]	(1)Add collected data to database (2)Collect data (3)Send reports (4)Collect patient's files who have been discharged (5)Add collected data to database (6)Collect data (7)Send patient's file	
Service 9 Patient[Process_3] + Doctor (Diagnostician)[Process_5] + Receptionist (Cancer detection unit)[Process_4]	(1)Pay or come to an agreement (2)Visit doctor (3)Book appointment by phone (4)Visit Clinic (5)Check if patient needs admission (6)Receive patient visiting doctor (7)Refer patient to special combined clinic (8)Review lab and imaging results (9)Perform clinical appraisal (10)Check if patient needs imaging investigations (11)Book appointment for patient (12)Take notes and review history (13)Check if patient needs investigations (14)Order test (15)Update patient's file (16)Inform patient to visit doctor (17)Check if patient medically insured (18)Register Patient details (19)Book appointment (20)Check if the patient is in DB	

Appendix E
Plan for the Interactive Search Experiment

<u>Plan for the CCR case study at KHCC August/September 2017</u>

Table of Contents

1. Introduction

This research is an attempt to bridge the gap between the business context and the systems in terms of services. In this research, we have adopted a set of methods to derive candidate services that conform to the Service Oriented Architecture (SOA) model of computing from the enterprise Business Process Models (BPM). Then, critically evaluate the quality of the resulting SOA service solutions, thus, we have adopted a case study to evaluate the outcomes of our research. This report aims to present the plan for the conducting the case study at King Hussein Cancer Centre (KHCC) in Jordan. This report shows clearly the objectives, tools, experimental design, and action plan for the case study. Furthermore, it describes some details about the participants, the procedure of collecting the data, the statistical analysis, and threats to validity. By providing this plan, we are trying to reach the highest level of reliability, and make the best use of the available resources. This plan helps to understand the goals of our visit to KHCC, the data to be collected, and the nature of experiment we are conducting at the KHCC. This is a very important step in order to achieve the best possible results.

2. Objective

By visiting KHCC and meeting the management and the domain experts we are targeting two main goals to be achieved as follows:

a. Assess the impact of human preference on the search, when mapping the BPMs to the SOA solutions. In other words, we aim to bridge the gap between BPM and SOA using interactive search, and assess the advantage of using the human preference to steer the trajectory of the search.
b. Assess the search outcomes using feedback of the domain experts at KHCC. And then compare the SOA solutions produced using search to the SOA solutions produced using interactive search in terms of quality and level of satisfaction.

3. The Tool Setup and Environment

The setup includes the software tool and the environment. In the tool, we describe the software application that we have prepared for this experiment. Whereas, in the environment part we discuss the needed machines and the required operating system.

a. The Search Engine Tool

The tool (i.e., also called the "search engine") comprises three main components; the BPM parser, the genetic algorithm (GA), and a decoder. All the three parts are integrated in one project in the form of an executable file, therefore, this file runs once it has been placed on the targeted machine. The BPM parser tool traverses the BPM models that placed on a local directory (i.e., the input files have ".bpmn" extension) and collects the required elements. The parser then prepares these elements in the right granularity level and builds the search space. The GA search engine searches for good-enough solutions, and assess these solutions using a fitness function. Moreover, the search engine allows the participant to interact with the search during the search process through a special GUI that provides the needed information and then asks the participant to evaluate some services. Finally, the decoder refines the resulting solutions from the search engine. It maps the BPM elements to the standard Web Service Description Language (WSDL) file components in order to produce the WSDL files. The search engine tool (i.e., the three parts) produces a set of files to document the results. These files contain the values of the design metrics (i.e., coupling, cohesion, human preference), the average fitness values for the population, the average fitness values for the best individuals, etc.

b. The Environment

The search engine tool has to be installed on a number of machines based on the number of participants (i.e., ideally 6-8 participants), such that each participant will work individually on a machine. The operating system can be Windows (i.e., 7 or 10), Linux, or Mac OS. Installing this tool requires the existence of the Java JRE library on the targeted machine (i.e., mostly available by default on all OS types).

4. Experimental Design

A within-group design is adopted. Expected number of participants is 6-8 domain experts. Each of them should firstly interact with the search in order to produce at least one SOA solution. Secondly, each expert should evaluate an existing set of SOA solutions (i.e., these solutions are produced by either search, or interactive search). The solutions produced by search will be prepared in advance in the first experiment. The independent and dependent set of parameters for the two experiments are presented in Table 5.1. In out the visit to KHCC we will conduct the second experiment as it requires the interaction with domain experts.

Table E.1: Parameters of the Search Experiments

#	Experiment	Parameter Type	Values
1	GA Search Vs Baseline	Independent	Crossover probability Mutation probability Population size Evolve times Breading pool size Tournament selection size Elitism
		Dependent	Effectiveness (i.e., in terms of Coupling, cohesion, and fitness value) Efficiency (i.e., in terms of time, and number of executions)
2	GA Search Vs Interactive GA	Independent	GA, Interactive GA
		Dependent	Quantitative values: Coupling, cohesion, and fitness. Qualitative values: level of satisfaction

Note that in the first experiment, we evaluate the search-based technique in comparison with the baseline search (i.e., using a random method). This experiment aims to find the combination of genetic parameters that will produce SOA models with good-enough fitness values. Samples out of this experiment should be ready in advance to be used in the experiment at KHCC. A set of selected solutions (i.e., randomly selected among a pool of good solutions) will be evaluated by domain experts to measure their level of satisfaction in comparison to the solutions they will produce interactively.

In the second experiment, the genetic parameters are fixed for both the search and the interactive search. The independent parameter is the existence or absence of human preference during the search. Therefore, the results focus on two major outcomes; the fitness values, and the participant's level of satisfaction.

5. Data Collection

Data will be collected from users before and during the experiment. No personal information about the participant will. Then the quantitative and qualitative values of design metrics will be collected while running the search engine. For each experiment, the search will be performed 10 times, and the average values will be observed. The data to be collected are described as follows:

a. Participant information: no personal details will be collected.

b. Interactive search results: quantitative values that include coupling, cohesion, fitness value of the entire population, and fitness value of the best individual. And qualitative value that represent participant's preference value. Note that these values will be collected during the search process.

c. Non-interactive search results: quantitative values that include coupling, cohesion, average fitness values of the entire population, and fitness value of the best individual.

d. Level of satisfaction results: qualitative values that are collected from participants when they evaluate different SOA solutions that are created by either the search or the interactive search. Therefore, Likert scale will be used here to collect a value from 1-5 for each service. The values will be scaled to be from 0-100, and the average will be calculated to find the human level of satisfaction.

6. Participants

Domain experts are the people who have a good understanding to the work process and workflows in the organisation. Therefore, doing an experiment with large number of domain experts could be one of the challenges, simply because these experts are mostly busy. However, conducting this experiment using a small sample size causes random errors. To solve this problem, we may suggest a moderate number of participants to do the experiments. Ideally, this number is six to eight participants, who are domain experts, technical or IT experts. As we have already adopted a within-group design that means that each participant would evaluate SOA solutions that were produced using either search or interactive search.

One of the challenges that may appear here, is that a domain expert could be familiar with one part of the workflow but not the other parts. Normally, the participants have experience in their domains but not in the entire workflow of KHCC. To overcome this challenge, we can hire knowledgeable people with high level of experience if available, or we can allocate more than one domain expert in a group to work as a team.

Another challenge, is that domain experts are typically not familiar with SOA services or the software in general. To overcome this challenge, we will provide a training session to explain the basics of the SOA services, and offer the participants the needed knowledge that is sufficient to work with the search engine tool. In addition, the experimenter should provide any needed support to help the participants understand the tool.

7. Statistical Analysis

Based on the normal distribution of the resulting data, statistical tests will be applied as follows:

a. Normal data distribution:
    i. Repeated measures ANOVA
    ii. T-test (i.e., student test).
b. Non-normal Data distribution:
    i. The Wilcoxon signed-rank test (If the data is non-normally distributed): a non-parametric statistical hypothesis test used when comparing two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ (i.e. it is a paired difference test).
    ii. Kolmogorov–Smirnov (K-S) test.

8. The Case Study

The Cancer Care and Registration (CCR) case study comprises 18 models that comprise 33 processes. This case study presents the full workflow of KHCC, and describes in details the activities that lead to fulfil the cancer care functionalities. Data of the workflow was elicited by Faisal Aburub (2006), and then modelled using Business Process Model and Notation (BPMN) by Yousef (2010). More details about the case study can be found on the following links:

http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.429689
http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.524727

During the experiment, the participants will not interact directly with the BPMN graphs or workflows, instead, they will interact with the data derived from the BPMs (i.e., in the first phase of data preparation phase). The data preparation and the friendly Graphical User Interface (GUI) play a significant role in helping the participant to understand the needed tasks and interact with the search in a creative way.

Representative data segments: the size of the case study is large (i.e., 18 models with 33 processes) which makes it challenging to find experts who are aware of each step in the entire workflow, in addition to an experience in the SOA services. To overcome this challenge, the participants should not see all the details of the case study, instead, they see some representative segments that give sufficient details to assess the information. However, the mapping from the business process to the services is not only a subjective issue, rather, it is built on criteria that guarantees that the resulting services reflect the actual business functionalities. This criteria comprises a set of key constraints as follows:

a. All the business elements in the BPMs should appear clearly in the resulting services. A traceability criteria should be available to check for the existence of each business functionality in the services. This includes the activities that are connected together to fulfil one functionality, such as booking an appointment. When a patient books an appointment to visit a doctor, he/she first calls the reception, then the receptionist checks for the patient record, if there is a record then a payment should be done first to book an appointment. All these activities are connected in a special order, so in the service they should appear as operations in one service that fulfils the entire functionality.
b. The connection between functions: should be the same, the connected functionalities in the BPM should be reflected as connected services. For example, visiting a doctor cannot take place before booking an appointment. Therefore, the two processes are connected sequentially by booking an appointment successfully, and then proceed to the doctor visit subsequently.
c. Data elements: should be assigned to parameters. If a data object exists in the BPMN elements, then it should appear as a parameter in the resulting service. For example, booking an appointment for the patient requires the patient to send his/her information to the receptionist, which appears as a data object in the BPM elements, and as a parameter in the corresponding service.

The participants should have an access to all the data segments in the case study as a record to be used when comparing the resulting services to the original source in the BPM. However, during the sessions, a technique on how to trace the source of each operation in the services will be presented to the participants, which simplifies the traceability. One of the key methods to trace the source BPMs is the naming convention that is adopted when creating the services. Each resulting service will show the name of the original process and model it came from. The names of these tasks and functions are unique to avoid any conflicts.

Interaction with the search: In this experiment, the participants interact with the search engine to steer the trajectory of the search. In this phase, the genetic algorithm starts to evolve the existing population in the search space, and then select some individuals from the population to be presented and evaluated by the participant. A Likert scale of five numbers will be used to evaluate each service. Moreover, a screen that shows the connections between services helps the participant to make more accurate decisions about the SOA model before making his/her assessment.

Evaluation of SOA Solutions: after producing SOA solutions in the previous step, the SOA resulting models should be evaluated, and compared to other solutions produced by the search engine only. The participant in this case studies a set of selected SOA solutions, some of these solutions are produced by search, and others are produced by interactive search. It is important that the participant study the selected solutions carefully, and then evaluate them (i.e., either by evaluating each service individually or by evaluating the full solution). The results of this test are useful to analyse the data statistically, in order to evaluate the human impact on the resulting SOA solutions.

9. Action Plan

The details of our action plan for our visit to the KHCC are described in Table. Note that the first step is presenting the project for the stakeholders. Next, the lab should be prepared for the experiments by installing the search engine tool in the lab. Four sessions are proposed to do the data collection and applying the experiments. It is important that the same participants should attend the experiment's sessions.

Table E.2: Action Plan of the case study at KHCC

#	Action	Step Description	Responsible	Date	Remarks
1	**Meet the Management**				
	Presentation	Meet the stakeholders of KHCC and provide the presentation	Hamzeh	7/9	Stakeholders have different background, so the presentation is not for researchers, PhD students, or technical audience
2	**Lab Preparation**				
		Install the tool on the lab machines, test the tool, and make sure that the logger records all the needed data	Hamzeh	10/9	The expected number of participants is 6-8. Prepare the software on the available machines to be ready for the practical sessions
3	**Goal #1**: Interact with the search				
	Session's expected outcomes	**Solutions produced using interactive search**			
	Introduction	Introduce the problem, solution, and benefits for KHCC	Hamzeh	11/9	**Time**: 10 minutes
	Get started with the tool	Present the tool and show participants how to use it	Hamzeh		**Time**: 10 minutes
	Interactive search session	Let the participants to use the interactive tool and steer the search. Record all the data and results	All participants **(Domain Experts)**		**Time**: 60 minutes. - The target for each participant to produce 3-5 solutions that satisfy the needs of the experts (targeted total is 15-25 solutions)
	Wrap up	Conclude the session, collect data, and brief introduction to the next evaluation day	Hamzeh		**Time**: 10 minutes
4	**Goal #2**: Compare solutions that were produced using the interactive or non-interactive search				
	Session's expected outcomes	**A table that presents the level of satisfaction about the solutions produced using the GA and the IGA**			
	Introduction	Introduce the needed activity and present the tool	Hamzeh	13/9	**Time**: 10 minutes
	Evaluate the selected candidate services	The tool will present a set of service solutions that were generated with/without the human interaction. The experts will be asked to evaluate each service solution using a 5-number scale	All participants **(Domain Experts)**		**Time**: 70 minutes - Note that the participants do not know whether the service solutions were produced automatically or with human interaction
	Wrap up	Conclude the session, collect data, and introduce the final day	Hamzeh		

5	**Goal #2**: Produce manual solutions and show the level of satisfaction about them				
	Session's expected outcomes	Manual solutions that can be quantitatively evaluated and compared to the search outcomes			
	Introduction	Introduce the session and objective, and show the tool	Hamzeh	17/9	Time: 10 minutes
	Produce manual solutions	Ask experts to produce manual solutions to be compared to other solutions quantitatively	**(Domain Experts)** Each expert individually		**Time**: 70 minutes Each expert will produce a manual solution. Then each expert should evaluate other expert's solution to show the level of satisfaction
	Evaluate full solutions that were produced manually by other experts and get the level of satisfaction	Evaluate candidate solutions that were produced with human engagement	**(Domain Experts)** Each expert individually		
	Wrap up	Conclude the session, thank the participants, and collect the data	Hamzeh		**Time**: 10 minutes
6	**Extra Details: extract more details about the services and systems available at KHCC**				
	Meet with dr. Emad Traish*	Interview dr. Emad, and get more details about the available systems and services. furthermore, the goal of this meeting is to find out the system that provides each of the services we are working on.	Hamzeh And **Dr Emad Traish Or Mr Nawras Libzo**	17/9	**Time:** 30-45 minutes *This meeting will be arranged upon request as the needed details may benefit other projects as well
7	**Statistical Analysis and Writing the Results**				
	Statistical analysis	Perform the statistical analysis tests, produce charts and tables	Hamzeh	18/9	
	Writing up	Writing Up the Results of Chapter 6	Hamzeh	20/9	

10. Threats to Validity

All research studies strive for high reliability. In order to conduct a reliable study, we should avoid errors.

a. Random errors: this type of error pushes the observed value in all directions. If the sample size is large enough, the random errors in different directions set each other, so in result, the mean value is very close to the actual value (Lazar et al., 2010). We can avoid random errors by enlarging the sample size. In other words, have more experts to participate in the case study and collect more data samples.

b. Systematic errors (or biases): in order to avoid bias, we should allow the participants to create their SOA solutions by interacting with the tool before working on the SOA solutions that were produced by the search engine, which minimizes the learning effect. Another issue, is the influence of personal relations between participants when they evaluate the services. Probably, the evaluation could be affected if the participant knows the source or the creator of the SOA solution. To overcome this issue, we should hide the identity of the person or machine who created the SOA solution. This also gives the results more accuracy and reliability.

   Another cause of bias is the human fatigue. If the participants feel tired, then the results will not be accurate. Therefore, to overcome this issue, we should consider a set of procedures such as minimise the number of interaction times when doing the interactive search, make sure to give the participants enough breaks, and make sure that the participants feel comfortable about the physical environment in the lab.

References

Ahmad, M., 2016. Semantic derivation of enterprise information architecture from riva-based business process architecture (Doctoral dissertation, University of the West of England).

Lazar, J., Feng, J.H. and Hochheiser, H., 2010. Research Methods in Human-Computer Interaction. John Wiley & Sons.

Rub, A. and Farid, F.A., 2006. A business process improvement methodology based on process modelling, applied to the healthcare sector (Doctoral dissertation, University of the West of England, Bristol,).

Yousef, R.M., 2010. BPAOntoSOA: A Semantically Enriched Framework for Deriving SOA Candidate Software Services from Riva-based Business Process Architecture (Doctoral dissertation, University of the West of England, Bristol).

Appendix F
Participant Material for the Interactive Experiment

This appendix presents the information sheet that was sent to participants before participation in the interactive sessions to derive cancer care services. This sheet includes details about the project, participants, confidentiality, and action plan.

PARTICIPANT INFORMATION SHEET

Background, Purpose and Goals of Research

This research is an attempt to bridge the gap between the business context and the systems in terms of services. In this research, we have adopted a set of methods to derive candidate services that conform to the Service Oriented Architecture (SOA) model of computing from the enterprise Business Process Models (BPM). Then, critically evaluate the quality of the resulting SOA service solutions, thus, we have adopted a case study to evaluate the outcomes of our research. This report aims to present the plan for the conducting the case study at King Hussein Cancer Centre (KHCC) in Jordan. This report shows clearly the objectives, tools, experimental design, and action plan for the case study. Furthermore, it describes some details about the participants, the procedure of collecting the data, the statistical analysis, and threats to validity. By providing this plan, we are trying to reach the highest level of reliability, and make the best use of the available resources. This plan helps to understand the goals of our visit to KHCC, the data to be collected, and the nature of experiment we are conducting at the KHCC. This is a very important step in order to achieve the best possible results.

By visiting KHCC and meeting the management and the domain experts we are targeting two main goals to be achieved as follows:

Assess the impact of human preference on the search, when mapping the BPMs to the SOA solutions. In other words, we aim to bridge the gap between BPM and SOA using interactive search, and assess the advantage of using the human preference to steer the trajectory of the search.

Assess the search outcomes using feedback of the domain experts at KHCC. And then compare the SOA solutions produced using search to the SOA solutions produced using interactive search in terms of quality and level of satisfaction.

Selection of Subjects and Conduct of Research

One of the most important steps in the evaluation process is choosing the appropriate subjects; as this ensures quality and validity (Okoli and Pawlowski, 2004). Since the interactive search experiment focuses on eliciting domain expert opinions, the criteria relied on must ensure the selection of the most knowledgeable and experienced in the subject area. The proposed selection criteria include competencies relating to the qualification, position, professional experience (Gupta and Clarke, 1996). Other factors include willingness of participants to be part of process. Best practice requires that a set of qualifying criteria is used to prequalify a list of possible subjects, who can then be officially invited stating the requirements for participation (Rowe and Wright, 1999).

Selection Criteria

Selection of experts is based on the following guidelines:
Having extensive working experience in one of the KHCC departments, with the full understanding of all the processes inside that department.
Having a general understanding of the entire business processes and workflow of KHCC.
Having a sound knowledge and understanding of the business process or business modelling.
Knowledge of business requirements, or software services is an extra advantage, but it is an optional requirement for this study.

Sample Size and Participants' Recruitment
The sample size for the study will consist of 6-8 participants. The recruiting of these participants will be done through:
The chair of the research council with the cooperation of the IT manager at KHCC will select a group of potential participants to do the experiments. The selection of these potential participants is based on the defined criteria (i.e., primarily because of their experience in the business process of the KHCC).
Invitations to the potential participants will be sent by the chair of the research council and IT manager through the KHCC email following their own internal procedure, and then an invitation to participate in this study will be sent to the participants in addition to a summary of the study.
Participants have the right to accept the invitation as volunteers, or reject the invitation without making any further clarifications. There is no penalty for rejecting the invitation.

Participants for this project will be selected primarily because of their knowledge and experience of the business process of KHCC. With regard to your place of employment, where appropriate, your manager / supervisor will be contacted to inform and explain the background of the research study.

We will ask you to perform a number of service identification, using the interactive research software to assist you. Before you use the interactive research software, we will take time to explain how the research software works. We will not ourselves direct you in your interactive design sessions, but the interactive research software will record some of your actions and decisions service derivation progresses. You are welcome to ask for clarification at any point in the research service assisted identification sessions, but in general we will try to be silent observers. When using the interactive research software, you may halt the service identification session at any point without the need to explain why. However, we anticipate that the interactive service identification session will be quite enjoyable and run for about 30 minutes, although the session will be terminated by the researchers at 60 minutes to prevent fatigue.

After the service identification sessions have been conducted, we would like you to give us your professional opinions as to the strengths and weaknesses of the research software tool, and the interactive service identification experience overall.

Subject Participation is Voluntary

Participation in this research project is voluntary. Participants can withdraw their consent and discontinue participation at any time without any consequences up to two weeks after the interactive session with the research software. There is no penalty for withdrawal of consent. Your Manager / Supervisor will not know whether or not you have withdrawn participation. If you withdraw your consent, any recorded data will be discarded and not be used in the study.

Confidentiality

The research is not expected to contain personal data. The collected data is strictly confidential and will not be made available to anyone except as anonymous data. Individual participants may be identified by only reference ID numbers i.e. participant 1, participant 2. Names of participants will not be identified. Participants are not required to provide names, and other personal details.
None of the research tools will require individual name to be provided.
Collected data will only be used by members of this research project for analysis in an attempt to understand the effectiveness of the research tool/solution.
The participant's reference ID number will be used to allow the participant to withdraw his/her data within a specific period of time using this reference ID. This ID remains confidential.
Needed data were collected after the experiments and stored on an encrypted drive. The data will be accessed by the supervisory team and the research student only. The data will not be used for any other purpose. All digital data will be stored in on researcher's allocated PC in UWE which is password protected and has a high level of security. UWE official email will be used for correspondence to ensure some level of formality and confidentiality.

On completion of the research programme; the digital media will be destroyed after a period of 12 months following publication of the study results. Moreover, the findings of the research will be made available online and the KHCC participants will be informed through sending them a link.

Action Plan

#	Action	Step Description	Responsible	Date	Remarks
1	**Meet the Management**				
	Presentation	Meet the stakeholders of KHCC and provide the presentation	Hamzeh	7/9	Stakeholders have different background, so the presentation is not for researchers, PhD students, or technical audience
2	**Lab Preparation**				
		Install the tool on the lab machines, test the tool, and make sure that the logger records all the needed data	Hamzeh	10/9	The expected number of participants is 6-8. Prepare the software on the available machines to be ready for the practical sessions
3	**Goal #1**: Interact with the search				
	Session's expected outcomes	**Solutions produced using interactive search**			
	Introduction	Introduce the problem, solution, and benefits for KHCC	Hamzeh	11/9	**Time**: 10 minutes
	Get started with the tool	Present the tool and show participants how to use it	Hamzeh		**Time**: 10 minutes
	Interactive search session	Let the participants to use the interactive tool and steer the search. Record all the data and results	All participants **(Domain Experts)**		**Time**: 60 minutes. - The target for each participant to produce 3-5 solutions that satisfy the needs of the experts (targeted total is 15-25 solutions)
	Wrap up	Conclude the session, collect data, and brief introduction to the next evaluation day	Hamzeh		**Time**: 10 minutes
4	**Goal #2**: Compare solutions that were produced using the interactive or non-interactive search				
	Session's expected outcomes	**A table that presents the level of satisfaction about the solutions produced using the GA and the IGA**			
	Introduction	Introduce the needed activity and present the tool	Hamzeh	13/9	**Time**: 10 minutes
	Evaluate the selected candidate services	The tool will present a set of service solutions that were generated with/without the human interaction.	All participants		**Time**: 70 minutes

		The experts will be asked to evaluate each service solution using a 5-number scale	**(Domain Experts)**		- Note that the participants do not know whether the service solutions were produced automatically or with human interaction
	Wrap up	Conclude the session, collect data, and introduce the final day	Hamzeh		
5	**Goal #2**: Produce manual solutions and show the level of satisfaction about them				
	Session's expected outcomes	Manual solutions that can be quantitatively evaluated and compared to the search outcomes			
	Introduction	Introduce the session and objective, and show the tool	Hamzeh	17/9	Time: 10 minutes
	Produce manual solutions	Ask experts to produce manual solutions to be compared to other solutions quantitatively	**(Domain Experts)** Each expert individually		**Time**: 70 minutes Each expert will produce a manual solution. Then each expert should evaluate other expert's solution to show the level of satisfaction
	Evaluate full solutions that were produced manually by other experts and get the level of satisfaction	Evaluate candidate solutions that were produced with human engagement	**(Domain Experts)** Each expert individually		
	Wrap up	Conclude the session, thank the participants, and collect the data	Hamzeh		**Time**: 10 minutes
6	**Extra Details: extract more details about the services and systems available at KHCC**				
	Meet with dr. Emad Traish*	Interview dr. Emad, and get more details about the available systems and services. furthermore, the goal of this meeting is to find out the system that provides each of the services we are working on.	Hamzeh And **Dr Emad Traish Or Mr Nawras Libzo**	17/9	**Time:** 30-45 minutes *This meeting will be arranged upon request as the needed details may benefit other projects as well
7	**Statistical Analysis and Writing the Results**				
	Statistical analysis	Perform the statistical analysis tests, produce charts and tables	Hamzeh	18/9	
	Writing up	Writing Up the Results of Chapter 6	Hamzeh	20/9	

During the sessions, the tools and software applications that are used to derive services and evaluate different solutions will be introduced. Knowledge in the search technique, algorithm configuration, parameter tuning, or WSDL standard is not required. The only requirement is an experience and knowledge in the workflow of the KHCC (fully or partially). In addition, a basic knowledge in business process modelling is an advantage, however, a session will be conducted to explain these models and data segments to work with.

Appendix G
Comparison between BPMiSearch and BPAOntoSOA

Common and different services between BPMiSearch and BPAOntoSOA frameworks

Table F.1: Common services between BPMiSearch and BPAOntoSOA

Service ID	Search Space Elements (SSE)	Detailed functions
Service 1	Patient	(1) Pay or come to an agreement (2) Visit doctor (3) Book appointment by phone (4) Visit Clinic
	Receptionist (Cancer detection unit)	(5) Inform patient to visit doctor (6) Check if patient medically insured (7) Register Patient details (8) Book appointment (9) Check if the patient is in DB
Service 2	Patient	(1) Pay (2) Visit imaging department
	Imaging department	(3) Inform patient to visit doctor (4) Inform patient to visit doctor (5) Add and report results (6) Perform test (7) Receive payment (8) Receive patient visiting imaging department (9) Check if patient is medically insured (10) Check if patient has appointment (11) Book appointment for patient
Service 3	Patient	(1) Pay (2) Visit radiotherapy (3) Receive treatment

	Radiotherapy department	(4) Receive payment (5) Transfer patient (6) Ask patient to visit specialist (7) Transfer patient (8) Receive patient visiting radio (9) Check if the patient is medically insured (10) Check if patient has appointment (11) Begin treatment (12) Add results (13) Check if patient needs imaging test (14) Check if patient needs lab tests
Service 4	Patient	(1) Pay (101) Visit chemotherapy (102) Receive treatment
	Chemotherapy department	(103) Perform treatment (104) Receive payment (105) Receive patient visiting chemo (106) Ask patient to visit specialist (107) Add results (108) Check if the patient is medically insured (109) Check if patient has appointment
Service 5	Patient	(1) Handle payment (2) Visit lab
	Lab	(3) Receive patient visiting lab (4) Inform patient to visit doctor (5) Perform test (6) Receive payment (7) Check if patient medically insured (8) Add results

Table F.2: BPMiSearch Services

Service ID	Search Space Elements (SSE)	Detailed functions
Service 1	Receptionist	(1) Request patient's file (2) Check files (3) Return patient's file (4) Find patient's appointments
	Medical Records	(5) Register file's details (6) Find patient's file (7) Save patient's file in library (8) Send patient's file (9) Open file (10) Check files (11) Check if there is a new patient

Service 2	Inpatient care specialist and nurses	(1) Follow-up patient (resident doctor) (2) Make appointment in outpatient clinic with patient (3) Check if patient needs to remain in hospital (4) Perform treatment according to patient state (5) Check patient's financial state (6) Transfer patient to imaging department (7) Follow up patient state (resident doctor) (8) Check if patient needs imaging investigation (9) Send sample to lab (10) Check if patient needs lab test (11) (specialist) Review resident doctor's orders, diagnoses patients and review old tests
	Accounts clerk	(12) Approve patient's financial state (13) Check patient's financial state
Service 3	Medical records clerk	(1) Send reports to managers (2) Generate main statistical report (3) Analyse collected data (4) Collect data from different departments
	Patient	(5) Visit department (6) Visit admission department
	Admission clerk	(7) Check if patient is emergency case (8) Inform the patient to visit department (9) Add patient to waiting list (10) Check room availability
	Inpatient care specialists and nurses	(11) Begin surgery (12) Check if patient needs surgery (13) Transfer patient to chemo department (14) Check if patient needs chemotherapy (15) Check if patient needs imaging tests (16) Transfer patient to lab (17) Check if patient needs tests (18) Continue treatment (19) Receive patient visiting department and his papers (20) Check if patient needs other treatment (21) Open admission file (22) Transfer patient to radiotherapy department (23) Check if patient needs radiotherapy (24) Transfer patient to imaging department (25) Update patient file (26) Add notes to file
	Receptionist (Inpatient care)	(27) Collect patient's files who have been discharged (28) Add collected data to database (29) Collect data (30) Send patient's file
	Receptionist (department-specific)	(31) Add collected data to database (32) Collect data (33) Send reports

	Receptionist (Outpatient clinic)	(34) Send patients' files (35) Add results into database (36) Collect data (37) Send list of patients (38) Collect patient files
	Registrar	(39) Add additional information (40) Check for additional information (41) Add primary tumour (42) Extract main details about cancer patient (43) Make copies of pathology reports and death certificates (44) Add required details in JCR form (45) Generate reports about cancer incidents in the hospital (46) Check if primary tumour exists in database (47) Add patient's details to database (48) Check if patient exists in database (49) Check if there is any contradictable data (50) Inform specialist about contradictable data
	Doctor (Diagnostician)	(51) Check if patient needs admission (52) Receive patient visiting doctor (53) Refer patient to special combined clinic (54) Review lab and imaging results (55) Perform clinical appraisal (56) Check if patient needs imaging investigations (57) Book appointment for patient (58) Take notes and review history (59) Check if patient needs investigations (60) Order test (61) Update patient's file
	Patient	(62) Visit clinic (63) Pay or come to agreement
	Receptionist (outpatient clinic)	(64) Guide patient to combined clinic (65) Check if patient has medical insurance (66) Receive payment (67) Receive patient visiting clinic (68) Check patient's appointment
	Patient	(69) Visit combined clinic

	Combined clinic (specialists)	(70) Check if patient needs tests (71) Order test (72) Check if the patient needs admission (73) Inform patient to visit radio (74) Receive patient visiting combined clinic (75) Book appointment for radiotherapy (76) Inform patient to visit chemo (77) Book appointment imaging department (78) Continue treatment (79) Review patient's history and all investigations (80) Check if patient needs other treatment (81) Check if patient needs chemo (82) Request admission (83) Check if patient needs radio (84) Book appointment for chemotherapy (85) Devise plan for treatment
	Patient	(86) Receive information to wait (87) visit specialist
	Specialist	(88) Perform suitable treatment according to patient situation (89) Check if patient needs tests (90) Check if the patient needs admission (91) Perform medical appraisal (92) Book appointment imaging department (93) Request another appointment (94) Send advices and instructions to patient (95) Check if patient needs another appointment (96) Take notes, review history and old tests (97) Update patient file (98) Request admission from admission clerk (99) Order test
Service 4	Receptionist (outpatient department)	(1) Send the list to registrar (2) Make list of patients who have not attended their appointments
	Registrar	(3) Inform patient's specialist to update file (4) Inform patient to visit specialist (5) Check if patient changed hospital (6) Contact patient (7) Update patient's file (8) Find patient's address

Table F.3: BPAOntoSOA services

Service ID	Search Space Elements (SSE)	Detailed functions
Service 1	- General reception - Patient	• Request appointment • Book an appointment by phone • Visit clinic • Transfer patient to emergency • Inform patient to visit cancer detection unit • Check if patient diagnosed

		• Inform patient to visit specialist • Check if patient has appointment • Check if patient in DB • Register patient's details • Make appointment • Check if emergency • Send the list to registrar • Make list of patients who have not attended their appointments • Visit clinic • Pay or come to agreement • Guide patient to combined clinic • Check if patient has medical insurance • Receive payment • Receive patient visiting clinic • Check patient's appointment
Service 2	- Patient - Clerk	• Visit department • Visit admission department • Check if patient is emergency case • Inform the patient to visit department • Add patient to waiting list • Check room availability
Service 3	- Patient - Receptionist$_{9}$ (Cancer detection unit) - Doctor (Diagnostician)	• Pay or come to an agreement • Visit doctor • Book appointment by phone • Visit Clinic • Inform patient to visit doctor • Check if patient medically insured • Register Patient details • Book appointment • Check if the patient is in DB • Receive patient visiting doctor • Refer patient to special combined clinic • Review lab and imaging results • Perform clinical appraisal • Check if patient needs imaging investigations • Book appointment for patient • Take notes and review history • Check if patient needs investigations • Order test Update patient's file
Service 4	- Inpatient care specialist and nurses	• Follow-up patient (resident doctor) • Make appointment in outpatient clinic with patient • Check if patient needs to remain in hospital • Perform treatment according to patient state • Check patient's financial state • Transfer patient to imaging department • Follow up patient state (resident doctor) • Check if patient needs imaging investigation • Send sample to lab • Check if patient needs lab test

		(specialist) Review resident doctor's orders, diagnoses patients and review old tests
Service 5	- Patient - Combined clinic (specialists)	Visit combined clinic • Check if patient needs tests • Order test • Check if the patient needs admission • Inform patient to visit radio • Receive patient visiting combined clinic • Book appointment for radiotherapy • Inform a patient to visit chemo • Book appointment imaging department • Continue treatment • Review patient's history and all investigations • Check if patient needs other treatment • Check if patient needs chemo • Request admission • Check if patient needs radio • Book appointment for chemotherapy Devise plan for treatment
Service 6	- Inpatient care specialists and nurses	• Begin surgery • Check if patient needs surgery • Transfer patient to chemo department • Check if patient needs chemotherapy • Check if patient needs imaging tests • Transfer patient to lab • Check if patient needs tests • Continue treatment • Receive patient visiting department and his papers • Check if patient needs other treatment • Open admission file • Transfer patient to radiotherapy department • Check if patient needs radiotherapy • Transfer patient to imaging department • Update patient file • Add notes to file
Service 7	- Patient - Specialist - Clerk [Accounts clerk] - Receptionist (Inpatient care) - Receptionist (department-specific) - Receptionist (Outpatient clinic)	• Receive information to wait • Visit specialist • Perform suitable treatment according to patient situation • Check if patient needs tests • Check if the patient needs admission • Perform medical appraisal • Book appointment imaging department • Request another appointment • Send advices and instructions to patient • Check if patient needs another appointment • Take notes, review history and old tests • Update patient file • Request admission from admission clerk • Order test • Approve patient's financial state • Check patient's financial state

		• Collect patient's files who have been discharged • Add collected data to database • Collect data • Send patient's file • Add collected data to database • Collect data • Send reports • Send patients' files • Add results into database • Collect data • Send list of patients • Collect patient files
Service 8	- Receptionist - Medical Records - Registrar	• Request patient's file • Check files • Return patient's file • Find patient's appointments • Register file's details • Find patient's file • Save patient's file in library • Send patient's file • Open file • Check files • Check if there is a new patient • Add additional information • Check for additional information • Add primary tumor • Extract main details about cancer patient • Make copies of pathology reports and death certificates • Add required details in JCR form • Generate reports about cancer incidents in the hospital • Check if primary tumor exists in database • Add patient's details to database • Check if patient exists in database • Check if there is any contradictable data • Inform a specialist about contradictable data
Service 9	- Registrar - $Clerk_3$ [Medical records clerk]	• Inform patient's specialist to update file • Inform patient to visit specialist • Check if patient changed hospital • Contact patient • Update patient's file • Find patient's address • Send reports to managers • Generate main statistical report • Analyse collected data • Collect data from different departments