GQ-BPAOntoSOA: A Goal- and Object- based Semantic Framework for Deriving Software Services from an Organisation's Goals and Riva Business Process Architecture

Yousra Hani Odeh

A thesis submitted in partial fulfilment of the requirements of the University of the West of England, Bristol for the degree of Doctor of Philosophy

Software Engineering Group

Centre for Complex and Co-operative Systems (CCCS)

Faculty of Environment and Technology,

University of the West of England, Bristol

January 2015

Acknowledgment

I would like to thank my supervisors Dr. Mohammed Odeh and Dr. Stewart Green for their support and guidance during this PhD journey. Their effort is highly appreciated.

Achieving a PhD is truly finishing a journey that is full of risks, ups and downs, where its survivors are few. This journey couldn't be achieved without the support from my parents, brother, sisters, advisors, family, colleagues and friends. This PhD journey required skills, tools, faith and most importantly, mind-set and soul that can prepare a risk taker. In fact, this research journey is not only a PhD thesis, it is a life experience that is full of lessons and challenges to develop a mind and character. During this journey, I have learned a lot and I hope that I faced the hits appropriately. This wouldn't be possible without kind and strong people around me.

I would like to thank my father who is a real partner in this journey. I deeply thank my mother, brother and sisters for their patience and faith. I am grateful to have them as a family. I don't forget to thank Nihad Al-Bitar who passed this life during my PhD. She left nothing but good deeds to remember. I am truly thankful for my friends from Jordan who have kept on checking and encouraging me in this journey. This has proved that distance is meaningless with us. A special thanks to my friend Alaa Al-Bakheet, who stayed close despite living far away.

Colleagues and friends in my office have shared with me every happy, sad and thrilling moment. I am delighted to thank Carmel Conefrey who is a very nice friend and office accompanier during days and weekends. She shared with me the thesis submission moments. I can't forget to thank Edward Wigley and Owen King who prepared tools and materials for my thesis binding. I thank Heather, Ben, Tom, Nick, Hanna and Jenna for their nice office companionship. Also, I don't forget to thank the polite faculty colleagues Abeer and Mahmoud for their availability and support. Finally, I thank every friend in UWE, Bristol and every friend in the UK.

Abstract

Understanding a business organisation is a primary activity that is required for deriving service-oriented systems that assist in carrying out the business activities of an organisation. Business IT alignment is one of the hot topics that concerns with aligning business needs and system needs in order to keep a business organisation competitive in a market. One example in this area is the BPAOntoSOA framework that aligned business process architecture and the service-oriented model of computing. The BPAOntoSOA framework is a semantically enriched framework for deriving service oriented architecture candidate software services from a Riva-based business process architecture. The BPAOntoSOA framework was recently proposed in order to align the candidate software services to the business processes presented in a Riva business process architecture. The activities of the BPAOntoSOA framework are structured into two-semantic-based layers that are formed in a top-down manner. The top layer, the BPAOnt ontology instantiation layer, concerned with conceptualising the Riva business process architecture and the associated business process models. The bottom layer, which is the software service identification layer, concerned with the semantic identification of the service-oriented architecture candidate software services and their associated capabilities. In this layer, RPA clusters were used to describe the derived candidate software service. Ontologies were used in order to support addressing the semantic representation. However, the BPAOntoSOA framework has two limitations. First, the derived candidate software services are identified without considering the business goals. Second, the desired quality of service requirements that constrain the functionality of the software services are absent. This research is concerned with resolving these two limitations within the BPAOntoSOA framework.

In this research, the original BPAOntoSOA framework has been extended into the GQ-BPAOntoSOA framework. A new semantic-based layer has been added into the two original layers. The new layer is concerned with conceptualising the goal- and quality- oriented models in order to address their absence in the original BPAOntoSOA framework. The new layer is called the GQOnt ontology instantiation layer. This extension has highlighted the need for aligning the models within the original BPAOnt intonation layer with the ones in the new layer. This is because the BPAOnt was the base for the identification of the candidate software services and capabilities. Therefore, a novel alignment approach has been proposed in order to address this need. Also, the original service identification approach is refined in order to adapt with the integration of goals and quality requirements.

The GQ-BPAOntoSOA framework, which is a goal-based and quality-linked extended BPAOntoSOA framework, has been evaluated using the Cancer Care Registration process. This is the same case study used in the evaluation of the BPAOntoSOA framework. And this is required in order to investigate the implication of integrating goals and quality requirements into the pre-existing BPAOntoSOA framework-driven candidate software services. This has shown that: (1) the GQOnt ontology does not only contribute to the extension of the BPAOntoSOA framework, yet it also contributes to providing a semantic representation of a business strategy view for an organisation. The GQOnt ontology acts as an independent repository of knowledge in order to have an early agreement between stakeholders with regard to business goals and quality requirements. The semantic representation could be reused for different purposes with respect to the needs. (2) the alignment approach has bridged the gap between goal-oriented models and Riva-based business process architectures. (3) the Riva business process architecture modelling method and business process models have been enriched with the integration of goals and quality requirements in order to provide a rich representation of business process architecture and process models that reflect an important information for the given organisation. (4) The service identification approach used in the original BPAOntoSOA framework has been enriched with goals and quality requirements. This has affected the identification of candidate software services (clusters) and their capabilities. Also, the derived candidate software services have conformed to service-oriented architecture principles. Accordingly, This research has bridged the gap between the BPAOntoSOA framework and the business goals and quality requirements. This is anticipated to lead to highly consistent, correct and complete software service specifications.

Table of Contents

CHAPTER FOUR

CHAPTER FIVE

CHAPTER SIX

CHAPTER SEVEN

CHAPTER EIGHT

Chapter One: Introduction

1.1 Research Context

Nowadays, business organisations are increasingly moving towards developing their software service-oriented systems stemming from understanding their business domain and associated activities, which are rich in knowledge, actively engaged in attaining business goals and the driver for deriving the related software services. These activities are rich in knowledge because of the overall re/used information involved in the documentation and design of the business strategies, information categorisation based on their business life time (e.g., short-term or operational and long-term or strategical) and finally the information embodied in business models that are represented for a particular purpose (e.g., business process architecture or detailed flow of work of business processes). Although few examples have been mentioned of the necessary business information that assists to some degree in determining the business needs, still there are many more beyond those simple examples.

Business activities are actively engaged in addressing goals, where the business should be carrying out its activities in a way that meet its market and environment needs that may be changing periodically or permanently. Otherwise, the business will drop out of the market. Hence, this leads to the need for some business activities to be activated, refined or dropped by adjustment. Finally, the understanding of the business domain and activities are described as the drivers for their corresponding software services, as they explicitly or implicitly embody the rationales beyond the need of the software system development.

Researchers in the software engineering field investigated bridging the gap between the organisation's business and its software systems from different perspectives in order to continuously adapt to the business needs (i.e., Business/IT Alignment (BIA) approaches). Many of these bridging approaches are still not fully automatic (Aversano et al, 2012), but maybe considered partially automated (Odeh and Kamm, 2003). In addition, many of these approaches are not simple to follow and they still need to be critically evaluated in practice to inform their validity.

A recent bridging, or a BIA attempt, is the research work of the semantically enriched framework for deriving Service-Oriented Architecture (SOA) candidate software services stemming from a Riva-based business process architecture, namely the (BPAOntoSOA) framework (Yousef, 2010). In this framework, SOA software services and their capabilities are semantically generated from a Riva-based Business Process Architecture (BPA) model and its associated process models. The first layer of the BPAOntoSOA framework is concerned with the semantic representation of the Riva-based BPA and its associated business process models. Whereas the second layer is concerned with identifying the candidate software services satisfying the BPA and business process models specified in the first layer.

However, the BPAOntoSOA framework does not address the relevant business goals and quality requirements (Yousef, 2010). This is maybe considered directly related to the Riva BPA method itself, as it does not take into consideration the business goals of the organisation being modelled. In addition, no attempt was taken in the BPAOntoSOA framework to link business goals and quality requirements ahead of the BPA design process before even the semantic enrichment of the associated modelled BPA. Thus, the candidate software services result this from instantiating the BPAOntoSOA framework are not business goals based and/or quality requirements driven. The need to address the quality of service requirements was also suggested as a further direction in the PhD thesis work of the BPAOntoSOA framework in (Yousef, 2010).

This research is aimed at addressing these two requirements (i.e., business goals and quality requirements) ahead of the process of the BPAOntSOA framework instantiation and critically informing the impact of these two requirements on the generated candidate software services. The impact must be investigated not only from the functional aspects but also from the additional/reused of new/extended candidate software services, respectively.

The presence of a business strategy view may assist in setting up a place in order to initiate goal-oriented modelling for addressing the aims above. This is because a business strategy view "captures the strategic goals that derive an organisation forward" and it is currently absent in the BPAOntoSOA framework (OMG, 2013). Thus, a business strategy view allows adding a strategic dimension to the original BPAOntoSOA framework. This addition extends the original framework into Goal-based and Quality-linked BPAOntoSOA (GQ-BPAOntoSOA framework), which is aimed as one of the main outcomes in this thesis. Mainly, having a business strategy view present a head of a business process architecture and associated business process models for the alignment purpose may assist in specifying software services that address business goals and quality requirements. This is anticipated reducing the gap between business and systems for the sake of an effective Business IT alignment (i.e., goal-based alignment). The presence of BSV contributes to improving a BIA via improving the level of IT with regard to addressing business goals. Also, it is anticipated to contribute to IT rapid transformation in order to meet business goals that can increase the competiveness and derive the success for an organisation. Employing a BSV in an organisation that carries out a BIA contributes to pathing the way for a structured link between the specified software services and the business goals for traceability. Finally, a BSV contributes to capturing an organisation's goal-based broad view and its scope using goals for business understanding. By extending the BPAOntoSOA framework, it is anticipated to contribute to semi-automating the alignment between business and IT from the point of view of addressing business goals and quality requirements. Accordingly, the software services are semi-automatically identified and specified with respect to addressing business goals and quality requirements. Hence, this aims at surviving the organisation and/or keeping it competitive. However, a misalignment may occur for another reason (e.g., risk mitigation).

This chapter acts as the cornerstone of the thesis research work through setting the scene for the following chapters. The chapter is structured as follows: Section 1.2 presents the research problem statement along with the research motivations. The research aim and objectives are presented in Section 1.3. The thesis work is based on evaluating the research hypothesis and answering the research questions formulated in Section 1.4. Section 1.5 presents a glance through the research paradigm adopted to address this work. The main contributions of the research are described in Section 1.6. During representing each chapter's work, few approaches are followed that will assist in determining each chapter's contributions with regard to addressing the research questions and phases within the research process. Those approaches and thesis structure are presented in Section 1.7. Finally, related publications resulted from this research are shown in Section 1.8.

1.2 The Research Problem Statement and Motivations

Since this research is concerned with adopting the entire BPAOntoSOA framework (Yousef, 2010), the focus of the research problem statement and associated research motivations are specialised and presented in relation to the BPAOntoSOA framework. The research problem statement directs to setting the research motivations and questions.

1.2.1 The Problem Statement

In this research, the term "problem" "does not necessarily mean that something is seriously wrong with a current situation" but it goes beyond that, as it could "indicate an interest in an issue where finding the right answers might help to improve an existing situation" (Sekaran and Bougie, 2013). The research problem statement is obtained from two sources: first, previous research in the organisation (i.e., the University of the West of England (UWE));

and second, the related literature. This section deals with expressing the research problems, but without going further in showing the anticipated contributions or results from solving the problem.

Accordingly, The *first resource* is the BPAOntoSOA framework and in particular the further directions section suggested at the end of the PhD thesis work conducted in the University of the West of England (UWE) in (Yousef, 2010). The *second resource* is the state-of-the-art literature in relation to the research domain.

The main research problem statement is:

"*The BPAOntoSOA framework-driven SOA-able candidate software services and associated capabilities lack addressing relevant business goals and desired quality requirements*"

An elaboration of the above problem statement is carried out through first conducting a critical appraisal of the BPAOntoSOA framework (Yousef, 2010) in order to detect and identify what gaps exist in the original framework in relation to the above problem statement that has generated further sub problems:

1) *There is a lack of addressing business goals in the BPAOntoSOA framework*: the identification of organisational goals is what distinguishes an organisation from others in the same business domain. The business-oriented layer (i.e., the first or the top layer) in the BPAOntoSOA framework concentrates on understanding the business of an organisation rather than considering the objectives or the rationales beyond the business itself (Yousef, 2010). The absence of such goal-oriented models that support the integration emerges as another omission within this problem. Consequently, this suggests the need to link the goals of an organisation in the process of instantiating the BPAOntoSOA framework.

2) *There is a lack of addressing business related quality requirements:* currently, the integration of quality requirements into the first layer of the BPAOntoSOA framework is not addressed and thereby the organisation's BPA and business process models are modelled with no consideration for their desired characteristics (e.g., security). Most BPA modelling approaches concentrate on the functional or operational behaviour of business organisations and do not address quality requirements. This, however, is anticipated to impact the associated business process models and the resultant generated software services due to the absence of

considerable key quality attributes such as performance, reliability, etc. Approaches that support the quality requirements representation emerge as another lack within this problem.

3) *Business goals are not deriving the candidate software services identification:* the current approach taken by the BPAOntoSOA framework does not take into consideration linking business goals into the Riva BPA model. Hence, this gap in the first layer of the BPAOntoSOA framework impacts the identification and specification of software services in the second layer of this framework.

4) *There is a lack of integrating the quality of service requirements in the candidate software services identification:* the current approach taken by the BPAOntoSOA framework fails in integrating and representing the key quality attributes in the Riva BPA model and its associated BPMs. Consequently, this gap impacts the identification and specification of the software services in the second layer.

5) The *absence of a business strategy view:* as the instantiation of the BPAOntoSOA framework starts from the business process architecture and associated process models, a business strategy view (OMG, 2013) is not a key component of the original framework. This is a further limitation of the BPAOntoSOA framework stemming from the Riva method itself as a BPA modelling method that does not take into consideration the business strategy view, obtained goals and key business quality requirements.

1.2.2 The Research Motivations

Research motivations are determined from the importance of solving the research problems. In general, the main features of the BPAOntoSOA framework motivated carrying out this research. The original BPAOntoSOA framework, that is domain independent, is a recent and promising research trend for identifying SOA-able candidate software services and associated capabilities automatically using essential models for an organisation such as BPAs and associated business process models. In addition, the BPAOntoSOA framework cooperated in growing significantly the area of bridging the gap between business processes and systems (Odeh and Kamm, 2003). Each of the sub problems in Section 1.2.1 derived a corresponding motivation:

1) ***M1:*** The integration of business goals into a BPA and associated business process models may assist in determining their objectives; and therefore determining their need of usage or operation. Hence, this is anticipated to assist stakeholders in deriving their decisions regarding reusing or improving a business process. In addition, identifying the goals of business process models paves the way for identifying appropriate roles and activities that address the goals. This is anticipated reducing effort needed for business process design.

2) ***M2***: Integrating quality requirements into business process models that illustrate similar operation paves the way to feature a particular business process model from another. That is, if two business process models are designed to show the flow of the work of a transaction operation in a bank, where the first one considers security as a quality requirement and the second one does not, then the first business process is highly desired. In real world transaction business, security is a quality requirement. However, for some reason, they are absent in the business process model although security is required. This would distract a business analyst or designer from improving a feature that already exists in real environment, but absent in particular business process model. Hence, having quality requirements integrated and present will highly assist in improving business process models and their associated desired qualities.

3) ***M3 and M4***: Integrating goals and quality of service requirements into candidate software services contributes to enrich their specifications. Also, their integration may significantly lead to higher level of consistency, completeness and correctness regarding the identified software services. A software service is described consistent if it is coherent (i.e., no two software services share same goals and quality requirements). The completeness of software services is quantitative. It is determined with the comparison of the number of software services derived using the original framework. Simply, completeness means nothing is missing regarding the number of the original software services. The correctness of software services means that they can be accurately mapped to their drivers in BPMs, BPA and BSV. The consistency, completeness and correctness of software services are discussed in detail in Chapter 7 as part of the evaluation framework.

4) ***M5:*** Presenting a business strategy view for an organisation can assist in engaging appropriate stakeholders using goals identified in the business strategic view. Using goals appears as a simple and clear way to improve the business strategy for an organisation. Since a business strategic view embodies business goals, then they can be used to derive the BPAOntoSOA framework.

The motivations above are logically driven from the sub problems. This is because the BPAOntoSOA framework is very recent and not practiced yet in real institutes. Hence, neither statistics are generated nor evidences are shown regarding the efficiency and capability of the BPAOntoSOA framework. Therefore, this research work is concerned with extending the BPAOntoSOA framework in order to be practically exploited in further business context.

1.3 Research Aim and Objectives

This research aims to follow the BPAOntoSOA framework's path in contributing to the SOA paradigm by extending the original BPAOntoSOA framework in order to derive SOA candidate software services that take into consideration business goals and quality requirements that overall are stemming from the business understanding. The general research aim is as stated below:

"To investigate the implication of integrating the business goals and quality requirements on the BPAOntoSOA-driven SOA –able candidate services"

The research aim stated above is elaborated into a number of objectives that must be achieved in order to address fully the aim. The satisfaction of the research aim is based upon three main objectives.

Objective 1. **To represent semantically the business strategy view for an organisation that employs the BPAOntoSOA framework**.

Since a business strategy view is absent in the BPAOntoSOA framework, it is necessary to include it in the BPAOntoSOA framework. In fact, the integration of the business strategy view that comprises goal-oriented models along with the related soft goal models (i.e., quality-oriented models (Chung et al, 2000)). The role of semantic business strategic view leads for the alignment purpose with a BPA of an organisation, and particularly the Riva-based BPA (Ould, 2005). For this objective, it is anticipated that the semantic harmony of two goal-oriented approaches (i.e., one that deals with hard goals and another that deals with soft goals) will be investigated, and how both can contribute to generating a goal-based and quality integrated Riva-based BPA for the second objective.

Objective 2. **To bridge the gap between the business strategic models (resulting from achieving first objective) and the Riva-based BPA**.

In particular, the aim here is to generate a semantically enriched goal-based and quality integrated Riva-based BPA. This bridging aims to extend the work of the current Riva method in order to enhance the current service identification method of the BPAOntoSOA framework (Yousef, 2010). Furthermore, achieving this objective is anticipated to widen the practice of goals linking with business process architecture and its associated business process models.

Objective 3. **To utilise the goal-driven and quality-linked Riva-based BPA in order to enhance the current service identification method.**

In order to achieve this objective, it is aimed to investigate the adaptability of the current service identification approach using the extension added to the Riva-based BPA in relation to business goals and quality requirements. This concludes with the overall BPAOntoSOA framework as in contributing to bridge the gap between the business process and systems in (Odeh and Kamm, 2003).

In conclusion, by achieving the above stated research aim and associated objectives, it can be possible to inform the extent to which business goal and quality requirements when integrated into a Riva BPA can impact the candidate SOA services in identification and their respective quality requirements.

1.4 Research Hypothesis and Associated Research Questions

This section presents the hypothesis of this research work:

"Using the BPAOntoSOA framework, it may be possible to semantically derive goal-based and quality-linked SOA services from the integration of business goals into Riva-based business process architectures."

Accordingly, it is aimed to investigate the utilisation of the appropriate goal-oriented approaches as for representing both hard and soft goals. Such an investigation must take into the account understanding of business organisation in order to assist in linking business goals into the BPAOntoSOA framework. Thus, the business strategy view (OMG, 2013) via its models steers the way integrating business hard and soft goal into the BPAOntoSOA framework and particularly starting with the Riva method. This is because the Rive method is systematically utilised in the derivation of the identified candidate services for SOA-based systems (Yousef, 2010).

This research work involves conceptualising the desired business strategic view in order to align it with the Riva-based BPA. This alignment is anticipated to enrich the Riva method in order to derive the goal-based and quality-integrated candidate software services from the extended Riva-based BPA. In order to test the above hypothesis, a set of research questions have been formulated that are inline with the research objectives stated in the earlier section.

Research Question 1: What are the current shortcomings of the BPAOntoSOA framework in relation to integrating business goals and quality requirements? (Work of Chapters 2, 3, 4, 5 and 6)

Research Question 2: How can the gap between the goal-oriented models (business strategy view) and the Riva-based business process architecture be bridged? Moreover, how can goal-oriented models be aligned with a pre-existing Riva-based business process architecture model? (Work of Chapters 2, 3, 4 and 5)

Research Question 3: How can the BPAOntoSOA framework be utilised in order to identify QoS requirements for its pre-existing identified candidate software services using the goal-based and quality integrated Riva BPA? (Work of Chapters 5 and 6)

Research Question 4: What is the implication of integrating business goals and quality requirements on the derived candidate SOA services compared to the original BPAOntoSOA framework? (Work of Chapters 6, 7 and 8).

1.5 The Research Paradigm

The qualitative research questions, objectives, problems and aforementioned hypothesis constitute a qualitative research paradigm that involves a deeper understanding of the BPAOntoSOA framework (Yousef, 2010). The research process has been designed and conducted in order to extend the BPAOntoSOA framework by integrating goals and quality requirements as a way of enhancing the current service identification method, and thereby deriving a goal-based and quality-impacted software services from the associated enriched Riva BPA.

Accordingly, it is necessary to understand the relevant knowledge and literature work that includes the BPAOntoSOA framework, goal-oriented approaches, quality- or soft goal-oriented approaches, the Riva-based BPA method, SOA principles, QoS requirements and ontologies. The BPAOntoSOA framework needs an extension; thus, a set of design decisions has been appended to the current design in order to produce a new framework: the GQ-BPAOntoSOA framework. The extended framework is evaluated in order to identify any limitations.

The research work is comprehensively evaluated using the case study research strategy and in particular using the Cancer Care and Registration (CCR) in Jordan (AbuRub, 2006). In addition, the former UWE's CEMS Faculty of Administration (Ould and Green, 2004) is used in order to initially investigate the relationship between the proposed business strategic view, the corresponding Riva-based BPA and this relationship implication on the pre-existing candidate services.

The research process phases are described as being iterative and interleaved. The research process consists of: the preliminary phase, the early theoretical framework design phase, the investigation phase, the original BPAOntoSOA framework enhancement phase, the conceptual framework development phase, and finally the application and evaluation phase.

1.6 The Research Contributions

The primary contribution of this research is the ability to create goal-based and quality-linked SOA-able software services stemming from the alignment of a business strategic view with the business process architectures for an organisation. The derivation of the services is automated to some extent. The research contributions are summarised below.

1) ***C1***: A semantic representation of the business strategy view using an elegant synthesis of goal-oriented approaches and soft goal-oriented approaches has been designed. The view embodies the behavioural and the non-behavioural concepts that will propagate into the entire framework. The representation has been evaluated using

pilot and comprehensive case studies similar to the ones used in evaluating the BPAOntoSOA framework by Yousef (Yousef, 2010).

2) ***C2***: The gap between goal-oriented models and Riva-based business process architectures has been bridged. This work contributed to aligning a Riva-based BPA with a business strategy view.

3) ***C3***: The current Riva business process architecture modelling method has been enriched. This enrichment is achieved through deriving the architecture from an understanding of business goals. In addition, quality requirements have been integrated into the method and generated new concepts. Accordingly, this has enabled the associated business processes models to be linked to business goals and quality requirements with backward and forward traceability support.

4) ***C4:*** The service identification approach used in the original BPAOntoSOA framework (Yousef, 2010) has been enriched, where the candidate software services are derived from a goal-based and quality linked Riva process architecture. In addition, the service identification approach addressed the quality of service requirements.

5) ***C5:*** The full forward and backward semantic traceability has been addressed from business goals and quality requirements to Riva-based BPA and its associated business process models to the generated candidate SOA services.

1.7 The Organisation of the Thesis

This section introduces traceability mechanisms in each chapter in relation to the thesis research questions and research process phases. In addition, it presents the structure of the thesis with a brief summary of the work conducted in each chapter.

1.7.1 The Research Traceability Support Approaches

Each chapter will end with a discussion section that critiques the work presented and finishes with a conclusion with research traceability to the related research questions and the research process phases. A researcher may not be interested in knowing where every research question was answered and/or where each research was taking place regarding the thesis. These approaches not only support traceability of the conducted research by the researcher, yet they facilitate dealing of this topic from the reader's point of view. Two approaches carry out this effort are explained next.

1.7.1.1 The Research Questions Flagging Approach

This approach was originally proposed and conducted in thesis work in (Kossmann, 2010). A table is designed with respect to the research thesis chapters and the identified research questions as shown in Figure 1.1. If a chapter contributes to answering one research question or more, then the intersection of the chapter column and the research question row must be flagged within the table. Thus, this table provides traceability to the research question answering state.

1.7.1.2 Traceability to Adopted Research Process Phases

This approach appears original and has been adopted in this research. Following this approach, the thesis chapters are mapped to the relevant research process phases, which are taking place within the work of each chapter. Figure 1.2 illustrates this approach. The two tables, shown in Figures 1.1 and 1.2, are married in order to derive Figure 1.3 that shows the overall research traceability network.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1									
RQ2									
RQ3									
RQ4									

Figure 1.1: Research Plan Phases VS Thesis Chapters

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									
RP2									
RP3									
RP4(ai)									
RP4(aii)									
RP4(bi)									
RP4(bii)									
RP4(biii)									
RP5									
RP6									

Figure 1.2: Research Plan Phases VS Thesis Chapters

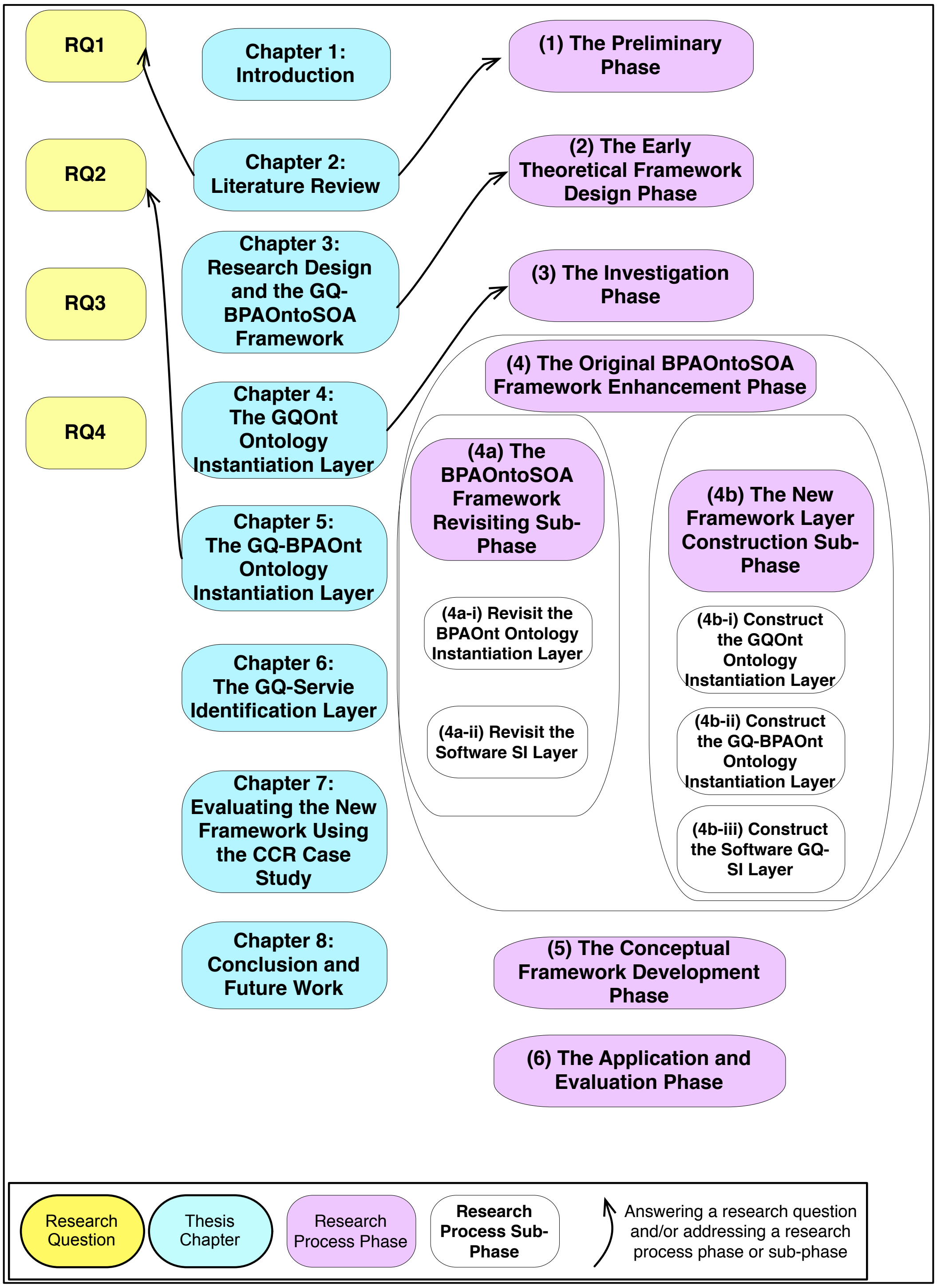


Figure 1.3: Research Traceability Network Using Research Questions, Doctoral Thesis Chapters and Research Process Phases.

1.7.2 Thesis Structure Overview

The PhD thesis is structured as follows,

- **Chapter 1** acts as the cornerstone of the research where its contents are the drivers of the rest of the other chapters. It presents the research context, the research problem statement, the anticipated main contribution, the research hypothesis, the research questions, research aim and objectives. In addition, it manifests two approaches that have been employed in the thesis chapters in order to trace research questions and adopted research phases.

- **Chapter 2** represents the state-of-the-art literature and background with a road map to the review of related work. In particular, this chapter presents an overview of current non-functional requirement classifications, goal-oriented approaches, current business process architecture modelling methods, current alignment approaches of goals with BPAs and current alignment approaches of BPAs and BPMs. In addition, the BPAOntoSOA framework is briefly discussed and reviewed (Yousef, 2010).

- **Chapter 3** presents the research design including the adopted qualitative research methodology. Moreover, this chapter presents the architectural design of the proposed framework, which is the GQ-BPAOntoSOA framework, and conducts a comparison between the new framework and the original BPAOntoSOA framework.

- **Chapter 4** reports on designing and development of the first layer of the new GQ-BPAOntoSOA framework. In addition, the ontology development of the business strategy view that involves the integration of goal-oriented models and soft goal models is presented. The initial evaluation is carried out using the CCR's patient reception process at the end of this chapter.

- **Chapter 5** critically reports on the alignment of the business strategy view with the Riva-based BPA. This chapter presents a refined Riva BPA modelling method. The relation between the business strategy view and the Riva BPA is initially demonstrated using the UWE's CEMS Faculty Administration process.

- **Chapter 6** reports on the application of the BPAOntoSOA framework service identification method on the semantically enriched Riva-based BPA presented in

Chapter 5. In this chapter, the SI approach is enhanced with the integration of goals and quality requirements.

- **Chapter 7** presents a comprehensive evaluation of the GQ-BPAOntoSOA framework using an evaluation framework. The GQ-BPAOntoSOA framework is applied using the CCR case study, where the outcomes of this application are compared with the BPAOntoSOA's outcomes.

- **Chapter 8** concludes the work of this thesis with further research directions. In this chapter, a brief comparison between the original framework and the extended one is presented.

1.8 List of Research-Related Publications

During this doctoral research journey, few publications generated from conducting this research as below:

- Odeh, Y. and Odeh, M. (2011) A NEW CLASSIFICATION OF NON- FUNCTIONAL REQUIREMENTS FOR SERVICE-OREINTED SOFTWARE ENGINEERING, In: 12th International Arab Conference in Information Technology (ACIT). Riyadh, 11-14 December 2011.

- Odeh, Y., Odeh, M. and Green, S. (2013) Aligning the Riva-based Business Process Architectures with Business Goals Using the i* Framework, In: the Third International Conference on Business Intelligence and Technology (BUSTECH 2013). Valencia, 27th May - 1st June 2013. Valencia: Think Mind, pp. 15-20.

- Odeh, Y. (2014) Integrating the Quality Requirements into the Riva-based Business Process Architectures Using the Synthesis of the *i** and the NFR Frameworks. In: The first International Conference on Software Engineering (ICOSE 2014). Dubai, 22-23 August 2014. Singapore: Lecture Notes on Software Engineering, pp.125-130.

- Odeh, Y. (2014) The Goal and Quality Ontology: A Semantic Business Strategy View. Semantic Web Journal. [In preparation].

- Odeh, Y. (2014) The GQ-BPAOntoSOA Framework: A Further Evolution of the BPAOntoSOA Framework through Semantic Alignment of Business Goals And Quality Requirements with Riva-based Business Process Architecture. Journal of Software Engineering Application. [In preparation].

- Odeh, Y. (2014) Bridging the Gap Between Goal Models, Business Process Architectures and Service-Oriented Environment. Information and Software Technology Journal. [In preparation].

Chapter Two: Literature Review

2.1 Introduction and Literature Review Road Map

Business/IT organisations that are operating in a common business domain are continuously moving to align their business needs with their systems' needs in order to compete or at least to survive in the domain. Ascertaining the *business needs* of an organisation involves eliciting and understanding its business goals and desired quality requirements and then designing business process models and business process architecture and many more models. Ascertaining a *system's needs* involves eliciting, analysing and implementing its functional and non-functional requirements. A recent semantic-based alignment framework is the BPAOntoSOA framework that supports the alignment of business needs with systems' needs from the functional perspective (Yousef, 2010). In particular, the BPAOntoSOA framework aligned business process architectures with the service-oriented model of computing through deriving candidate software services and their associated capabilities (i.e., functional requirements) from a Riva-based BPA and associated BPMs (Yousef, 2010). The BPAOntoSOA framework driven candidate software services are identified so that they meet SOA principles (Erl, 2007). In addition, the BPAOntoSOA framework was designed to operate in dynamic environments where business processes are continuously changing to meet business needs.

However, the BPAOntoSOA framework does not support the strategical level of alignment in two ways. First, the BPAOntoSOA driven candidate services are not goal-based and thereby they are identified regardless of whether or not they address business goals. Second, the candidate services were identified with only their functional capabilities (i.e., without considering the associated QoS requirements that constrain the capabilities (e.g., security)). The presence of quality of service requirements is very necessary, as a user may reject a given service's capabilities if it does not address the desired quality.

In this section, the road map of the research's literature review is presented. Figure 2.1 illustrates the road map by considering five main relevant areas, which are shown in the blue second row. Each of the next sections presents a main area from the second row of Figure 2.1. Section 2.2 presents and discusses the definition of business goals in the field. In addition, the current goal-oriented approaches in the requirements engineering process are reviewed that are anticipated to pave the way for addressing the limitations of the BPAOntoSOA framework. In Section 2.3, the current business process architecture approaches are reviewed

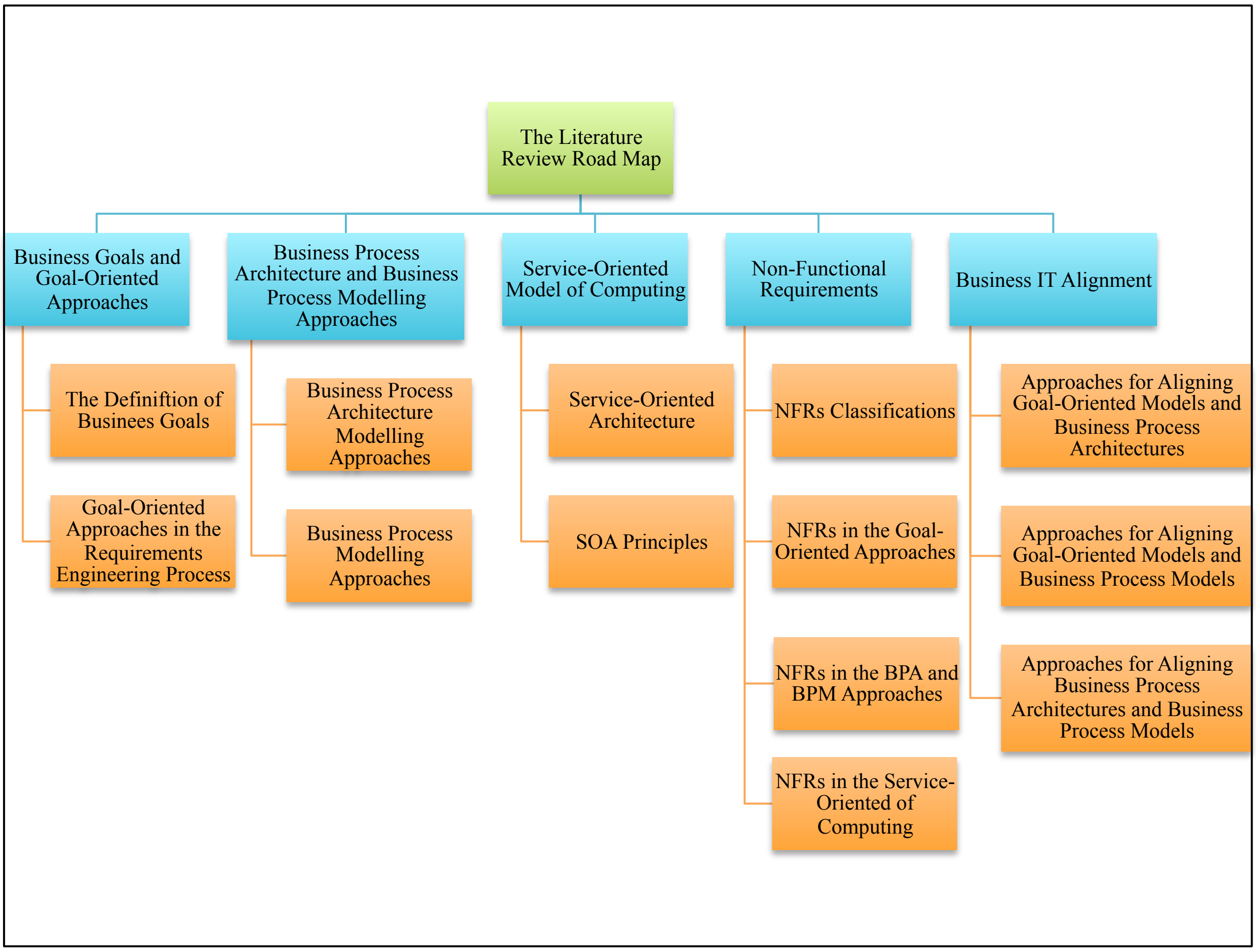


Figure 2.1: The Roadmap of the Relevant Background for the Research.

along with the current business process (Ould, 2005) method is presented with a generating the candidate services in an (Yousef, 2010). The service-oriented model of the SOA and its principles in Section 2.4. NFRs in the software engineering field are section involves presenting the participation business process architectures and business current attempts at aligning GO models and aligning the business process architecture and BPAOntoSOA framework is presented in this reviewed related work and concludes the

modelling approaches. The Riva-based BPA particular attention due to its key role in application of the BPAOntoSOA framework of computing is presented with a deep review In Section 2.5, the current classifications of presented. Moreover, the review in this of the NFRs in the goal-oriented models, process models. Section 2.6 reviews the the BPA, aligning GO models and BPMs and associated process models. The section. Finally, Section 2.7 discusses the chapter.

2.2 Goals and Goal-Oriented Modelling Approaches

Although many research work in the software engineering and particularly in the Requirements Engineering (RE) field are paying an attention to goals and their approaches that are found large in quantity, still the goal-oriented approaches are not widely practiced in most of the enterprises. And although goal-oriented approaches are not widely practiced in enterprises, still they influence the derivation of the business services in one way or another that many of them are automated and executed using software systems. From the two above contradictory statements, the importance of goals and their approaches, that analyse goals in order to exploit the business services that address the associated goals, are highlighted.

The Goal-Oriented (GO) approaches employ the concept of goal in order to articulate the strategic view of an enterprise. The GO approach is one of the methods used to arrive at a common understanding of a strategic view. According to the Object Management Group (OMG), a business strategy view of an enterprise is defined as follows:

"Business Strategy view (BSV): captures the strategic goals that drive an organization forward. The goals may be decomposed into various tactical approaches for achieving these goals and for providing traceability through the organization. These strategic goals are mapped to metrics that provide on-going evaluation of how successfully the organization is achieving its goals" (OMG, 2013)

The aforementioned definition embodies the rationale for the BSV. The OMG asserts that the Business Strategy View (BSV) contributes alongside other views (e.g., business capability view and business operational view) to establishing the business architecture view for an enterprise (OMG, 2013). The GO approaches output GO models that represent goals and the relationships between them. In this research, one of the objectives is identifying a major shortcoming within the BPAOntoSOA framework regarding goals (Yousef, 2010). It is apparent that a BSV is absent from the output (i.e., candidate software services) from the BPAOntoSOA framework applied to an enterprise, as will be shown in Section 2.6. Therefore, it is necessary to gain a deep understanding of the concepts that constitute the BSV, namely goals and their modelling approaches. In addition, it is necessary to consider the BSV anticipated cooperation with another models such as the business process architecture and associated business processes in order to steer the way involving the BSV into the BPAOntoSOA framework. In the next sub-sections, definitions of the concept of goal in the literature are presented.

2.2.1 The Definition of Goal

The term goal in the RE field has been defined from different perspectives, but mostly from the perspective of business (Young, 2004) or systems (Celements and Bass, 2010) (Lamsweerde, 2001) (Loucopoulos and Karakostas, 1995). On the one hand, most definitions have mentioned the goal concept either in a business context (i.e., business-oriented or business goals) or system context (i.e., system-oriented or system goals) without addressing the linkage between the both and their implications. On the other hand, others tend to agree that goals are stakeholder-centred. For example, Eric Yu defined the goal as "condition or state of affairs in the world that the actor would like to achieve" (Yu, 1995). Furthermore, (Roland et al, 1998) defined goal as "something that some stakeholders hope to achieve in the future". Definitions of goal are gathered from the literature and classified in Table 2.1. The goal classifications found in the literature are summarised in Figure 2.2. It is apparent that the literature lacks a universal goal classification due to the absence of an agreement of a comprehensive definition of goals. However, in this research the Anton's view of goals recovers this absence: goals are "*the high level objectives of business, organization or system that capture the reason why a system is needed and guide decisions at various levels within enterprise*" (Anton et al, 1994).

The above definition is the most appropriate, as it is comprehensive and relevant for addressing the BSV for the BIA and particularly within the BPAOntoSOA framework. Moreover, it highlights how system's objectives are supported by the objectives of a business organisation and hence, this implicitly enrols the business IT alignment as a research area. Anton's (Anton et al, 1994) goal's definition of goal determines the level of objectives that are being high in their abstraction to capture a reason. This view of a goal is required as it is anticipated to initiate the design of goal models in this research. The business, organisation or system goal works to guide the decisions by decomposing the high level objective into lower ones.

Table 2.1: Goal Definitions in the Literature.

Perspective	Definition of Goal	Reference
Business	The objective of an enterprise or organisations. It forms the driver of business requirements that are the essential activities in an enterprise.	(Young, 2004)
	-An enterprise goal is a desired state of affairs that needs to be attained. - A business process goal is an intended to achieve defined business objectives aiming to create value to customers.	(Kavakli and Loucopoulos, 1999)
System	Business goal expresses why system is being developed at all, and what stakeholders in the developing organisation, the customer organisation, and beyond aspire to achieve through its production and use.	(Celements and Bass, 2010)
	RE views goals as the intentions that capture the rationale for the system to be built and distinguishes between two categories of goals: hard goals (or simply goals) and soft goals.	(Soffer and Wand, 2005)
	It is described in term of business requirement that justifies the reason of developing systems. The essential activities in an organization define the business requirements that are derived from business goals.	(Young, 2004)
	Is an objective of system under consideration should achieve.	(Lamsweerde, 2001)
Stakeholder	Condition or state of affairs in the world that the actor would like to achieve.	(Yu, 1995)
	Something that some stakeholders hope to achieve in the future.	(Roland et al, 1998)

In Figure 2.2, goal types are classified into hard goals and soft goals according to the precision of the statement of their fulfilment criteria (Yu, 1995) (Slimane et al, 2009) (Chung et al, 2000). When the achievement criterion of a goal is sharply defined, this is a hard goal (Slimane et al, 2009). For example, for a student registering in a university program it is easy to determine the satisfaction of this goal as either "yes" the student is registered or, "no" the student is not registered. And a goal is said to be soft goal if its satisfaction criterion is not sharply defined and so requires an interpretation (Chung et al, 2000) (Slimane et al, 2009)(Yu, 1995). Therefore, the soft goal satisfaction varies in magnitude continuously, from satisfied through a range of degrees of satisfaction to not satisfied. For example, consider the goal of a student registering in a university program quickly. The soft goal in the example is "quickly" that is not fulfilled in clear-cut sense. Therefore, a set of approaches or operations should be identified in order to address the soft goal "quickly". The soft goals are quality requirements or constraints. Therefore, researchers consider soft goals as non-functional requirements in the software service engineering field, as will be shown in Section 2.5. It is

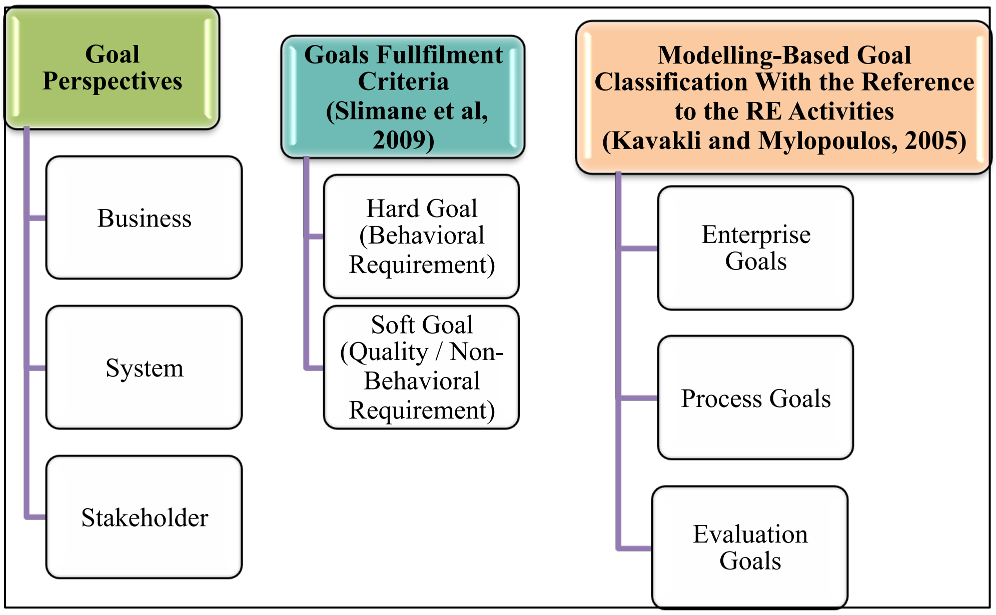


Figure 2.2: Summary of Goal Definition Classification in the Literature

1. Growth and continuity of the organization
2. Meeting financial objectives
3. Meeting personal objectives
4. Meeting responsibility to employees
5. Meeting responsibility to society
6. Meeting responsibility to country
7. Meeting responsibility to shareholders
8. Managing market position
9. Improving business processes
10. Managing quality and reputation of products

Figure 2.3: High Level Business Goal [Source: (Clements and Bass, 2010), Used with the author's permission].

important to distinguish between hard goals (i.e., they embody a behavioural or functional requirement) and soft goals (i.e., they embody a non-behavioural or quality requirement), as both will be involved in the BSV design or the GO model. Another goal classification is proposed in (Kavakli and Mylopoulos, 2005) based on the GO modelling approach in relation to the RE activities. In particular, they classified goals into three categories: enterprise goals, process goals and evaluation goals. Finally, a set of high-level business goals was proposed in (Clements and Bass, 2010), as shown in Figure 2.3, who claimed that any organisation must have at least one of its ultimate objectives belonging to the proposed set. They categorised goals by performing a structured literature review of current business goal categories.

2.2.2 Goal-Oriented Approaches in the Requirements Engineering

A Goal-Oriented (GO) approach is defined as an analytical method used in RE that "encourages the modelling of goals in order to understand or describe problems associated with business structures and processes and their supporting systems" (Kavakli and Mylopoulos, 2005).

The RE School is divided into two schools of thoughts (Wieringa, 2005). The first school is concerned with Problem-Oriented (PORE) approaches that are required to understand the current problem (Wieringa, 2005). The second school is concerned with generating Solution-Oriented (SORE) approaches (Wieringa, 2005). If a GO approach example intends to understand a problem, then the approach belongs to the first school (Wieringa, 2005). Hence, the Goal-Oriented modelling in Requirements Engineering (GORE) research emerged. For example, the i* framework (Yu, 1995) that is discussed later in detail in this Section 2.2.2.1.

The Volere requirements specification template (Volere, 2013), as a solution-oriented approach (Wieringa, 2005), has implicitly involved an early practice of goals in the project drivers' section before getting into system specification details. The IEEE-380 standard is another example of SORE (IEEE, 1993). This emphasises the importance of goals as an early trigger to system requirements and design models. Tropos, (Bresciani et al, 2004), is a software development methodology that aims to generate a system to-be in an environment using a goal-oriented approach. Tropos utilises the *i** framework from early requirements analysis activity to the detailed design for implementation to cover the whole software development process rather than being involved in one phase within the RE process (Sommerville, 2007). However, the methodology does not generate early to-be essential business models (e.g., business process architecture) that would assist in business IT alignment.

Researchers in the GORE community have generated most of the GO approaches and proposed guidelines that would pave the way for the selection of appropriate GO approach(s). Two rich and systematic guidelines of GORE are proposed in (Horkoff and Yu, 2011) and (Kavakli and Mylopoulos, 2005) (Poels et al ,2013) aided this research in selecting the appropriate GO approach for the BSV design in order to meet the research objectives. Figure 2.4 illustrates a framework guideline that aids the selection of GO approach(s) that meets the researcher's objectives.

All GO approaches that appear in Figure 2.4 agreed on refining and decomposing abstract goals to possess measurable entities (i.e., operationalisations (Chung et al, 2000)). The elaboration is carried out either through top-down approach (Chung et al, 2000) or in a network manner (Lamsweerde, 2001) (Yu, 1995) (Kavakli and Loucopoulos, 1999). Also, some of goal-oriented approaches highlighted actors as who are actively participating once a goal has been identified (Yu, 1995). Other GO approaches carried on elaborating goals without enacting actors at all (Chung et al, 2000), while others stopped the refinement once a goal is assigned to an actor (Lamsweerde, 2001) (Kavakli and Loucopoulos, 1999).

A different form of a GO approach was proposed in (Plosch et al, 2010) as a systematic approach that derives testable quality requirements from two independent GO approaches. The former is a GO approach and the latter employs goals regarding obstacles that obstruct the achievement of such business goals as shown in Figure 2.5. This approach refines goals to obstacles (O) and then finds out ideas (I) that overcome these obstacles. However, this approach is very complicated due that two GO approaches are engaged and this requires an extra critical knowledge on goals.

Bittencourt and others have successfully proposed a systematic approach for identifying quality requirements from business models of an organisation and facilitated an early investigation to information systems regards quality attributes (Bittencourt et al, 2010).

Clements and Bass proposed systematic GO approach to derive quality requirements to reflect on software architectures (Clements and Bass, 2010). This approach begins with a canonical set of high-level business goals as depicted in Figure 2.3. Metrics are used as an approach for relating quality attribute with related business goal(s) as in Figure 2.6.

It is apparent from the literature that the GO approaches did not participate in the old RE processes for the software development (Boehm et al, 1976). Many of current RE processes are still following the traditional trend in the engineering of systems' requirements through answering the *what* questions that are concerned with knowing what services that system must provide without addressing a critical understanding of the business domain (Sommerville, 2010) (Kotonya and Sommerville, 1998) (Boehm et al, 1976). The GO approaches are now apparently participating within the current evolved RE processes taking into the account the significance of involving the understanding of the business enterprises' goals and activities while generating their systems' specifications (Kavakli and Mylopoulos, 2005) (IBM, 2001). A goal-driven RE process have evolved the traditional RE process through being concerned with answering the *why* questions, which relate to knowing the rationales beyond the system services. In addition, answering the *how* questions, which relate to eliciting and understanding the techniques that address the rationales (Kavakli and Mylopoulos, 2005). A multidimensional classification of the GORE approaches was carried out in (Kavakli and Mylopoulos, 2005) as shown in Figure 2.4. In other words, the classification does not only involve the GO approaches usage within RE activities, but it involves their subjectiveness and their formality level of representation.

Goal-oriented Approaches / ***Framework Components***		Cognitive Task Analysis	***i*(*** strategic dependency model)	Goal-based workflow	EKD	F^3 (OM)	ISAC	SIBYL	The reasoning loop model	REMAP	KAOS	GBRAM	Goal-scenario coupling	NFR framework	GSN	GQM
Usage	understand current org. situation	✓	✓	✓	✓											
	understand the need for change					✓	✓									
	provide the deliberation context within which RE occurs							✓	✓	✓						
	relate business goals to system components										✓	✓	✓	✓		
	evaluate system specs against stakeholder goals														✓	✓
Subject	enterprise goals	✓	✓	✓	✓						✓	✓	✓	✓		
	process goals					✓	✓	✓	✓	✓						
	evaluation goals														✓	✓
Representation	formal		✓							✓	✓			✓		
	semi-formal	✓		✓	✓	✓		✓	✓			✓	✓		✓	✓
	informal						✓									
Development	way-of-working		◆		◆		◆				◆	◆	◆	◆		
	tool support	M	MF	M	M	MG	MG	M		MF	MF	MG	MG	MF	M	MG

◆ = suggest a number of steps and associated strategies
M = support for model construction, F = formal reasoning support, G = process guidance

Figure 2.4: The Goal-Oriented Approaches Classification Regarding the Requirements Engineering Process (Source: (Kavakli and Mylopoulos, 2005), Used with the permission of the author and IGI Global: Copyright IGI Global, reprinted with permission of the publisher)

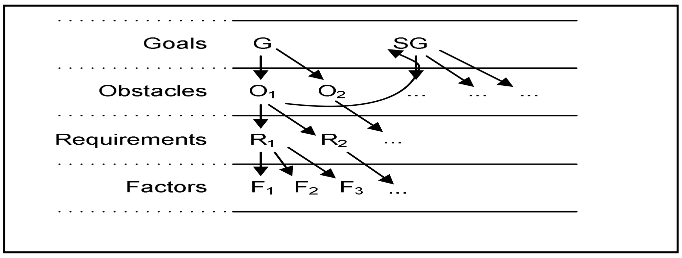


Figure 2.5: Goal-oriented Refinement regard Obstacles [Source: (Plosch et al, 2010), Used with the author's permission]

	Performance	Security	….
Business goal 1			
Business goal 2			
…			

Figure 2.6: Relating Business Goals to Quality Attributes [Source: (Clements and Bass, 2010), Used with the author's permission]

2.2.2.1 The i* Framework (Yu's Approach)

Previously in Figure 2.4, the *i** framework, pronounced as –i star-, is a GO approach that was categorised within the approaches proposed for understanding the current situation of an enterprise in relation to its goals as part of the RE elicitation activity (Kavakli and Mylopoulos, 2005). The *i** framework tends to early understand a business organisation through articulating the *distributed* ***i****ntentions* among its social actors in the organisation or business domain. Therefore, researchers consider the *i** framework as a modelling approach that assists in analysing and re/designing organisations (Chung et al, 2000).

The *i** framework is based on two strategic models: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model. The first describes the intentional structure of the organisation through configuring a network of dependency relationships between organisational actors. The second elaborates the first by articulating the rationales that organisational actors have to address in addition to its role in showing the internal link between the strategic dependency relations designed in the SD model. There is no particular sequence regarding the generation relation between the two models. There is no starting point to start reading and understanding a SD model or SR model.

The *i** framework has been applied in a number areas such as RE, business process reengineering, organisational impacts analysis and software process modelling. This research seeks to benefit from its application in the first two areas to achieve the research objectives. The SD and the SR models are explained next.

The Strategic Dependency (SD) Model

The purpose of the SD model is to represent a business process in an intentional structure rather than using conventional modelling approaches that pay the attention to entities, activities and their flow of work. The intentional structure is designed as a network of dependency relationships between actors who have the freedom to interact within an agreed boundary. The SD model in the *i** framework is designed as follows:

1. Identify the actors within the process or the organisation. If needed, actors are categorised into roles, agents and positions.
2. Sketch the dependency relationships between the actors where each relation is denoted by its dependum (i.e., goal, soft goal, task or resource) from the depender to the dependee.
3. Determine the dependency strength for each dependency relation as either open, committed or critical.

In the *i** framework, an actor is defined as an active entity - a human (e.g., doctor) or non-human, (e.g., flight reservation system)- that conducts a set of actions to fulfil a goal. Each dependency relation means a depender actor depends on another actor, namely a dependee in order to attain an objective called the dependum. If a depender did not attain the required dependum from a dependee , then a failure may occur while trying to address an objective and the depender becomes vulnerable. Dependency relations differ based on the participating dependum. Figure 2.7 depicts four dependency types between actors. For example, it illustrates how a patient (depnder) depends on a physician (dependee) for sickness to be cured (goal dependum). The SD model is very rich in business concepts relevant to this research (e.g., goal, soft goal, task, resource, actor, etc.).

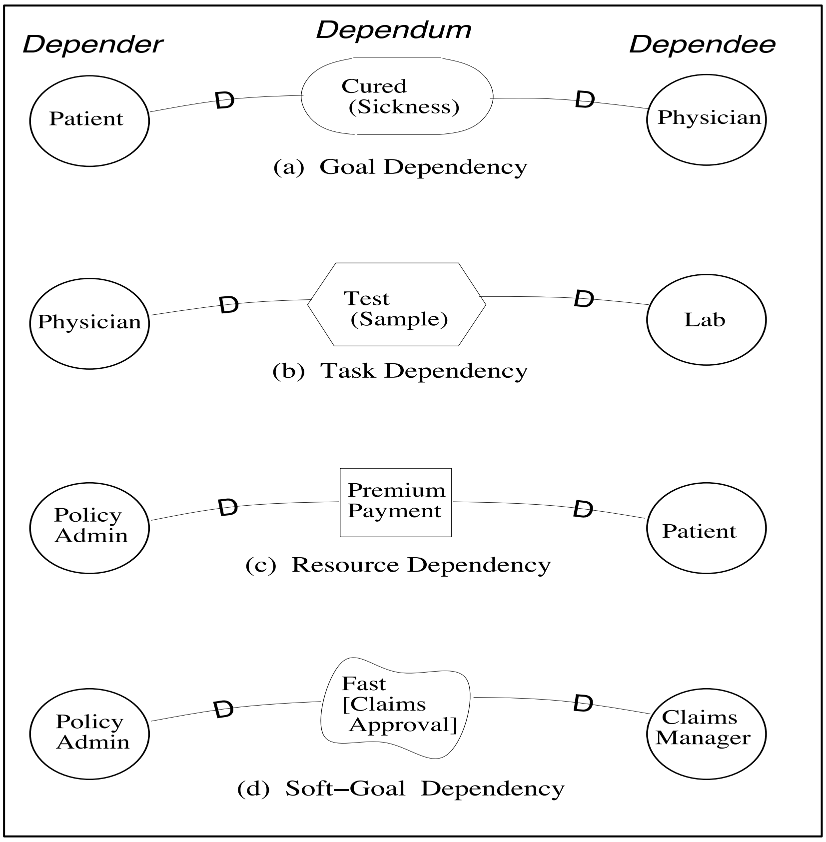


Figure 2.7: The Dependency Types in the SD Model in the *i Framework. [Source: (Yu, 1995), Used with the author's permission]**

Dependency types

The dependency relations in the SD model are categorised based on the participating dependum within the relation. In Figure 2.7, the goal dependency is the first type of dependency relation, where a goal stands for a state or condition to be achieved in the domain (Yu, 1995). In this dependency, the depender does not care how the dependee will achieve the goal as the latter has the freedom to choose the mechanism that filfill it. But the depender becomes vulnerable if it did not attain the goal for the dependee. For example, in Figure 2.7 the patient does not care how the physician will cure her/his sickness.

The second dependency type appears with a task dependum that simply means an activity. And here the depender does not specify why the dependency is required, but how the dependency is performed through using the notion of task. For example, a physician depends on a lab to test a sample by providing the latter with a set of instructions to follow as depicted in Figure 2.7. Also here, if task in not performed as specified then the depender will be vulnerable.

The resource dependency requires a depender that depends on dependee for the availability of a resource that is either a physical entity or information. Again, a depender becomes vulnerable if the resource is unavailable. For example, a policy admin depends on a patient to bring the premium payment as shown in Figure 2.7.

Finally, the soft goal dependency refers to a required quality requirement or non-functional requirement, where a depender depends on a dependee to address the soft goal. Addressing a soft goal on the dependee's side may be achieved through applying specified methods (e.g., NFR framework (Chung et al, 2000)). In other words, the soft goal dependum requires some refinements using the decomposition notion in order to be addressed rather than being sharply addressed in a clear-cut sense as goal and task. The dependee knows how to achieve the soft goal, but the final decision is taken by the depender to choose how the soft goal is achieved. Figure 2.7 illustrates a soft goal dependency, where a policy admin depends on a manager for a fast claim approval. Note that the soft goal embodies a quality attribute that is fast and associated parameter that is claim approval, as a target.

Dependency strength

A dependency relation has a strength that is measured in different degrees. The stronger the degree, the more vulnerable the depender will be in achieving its goals. The strength is ordered as follows: open, committed and then critical. On the dependee side, it performs great effort in achieving the dependum making it in its high priorities. Figure 2.8 illustrates the dependency degrees.

The least dependency strength is open and represented using "O" from a depender's side in a dependency relationship. An open dependency does not put the depender in risk regarding achieving its goals. Therefore, a dependee does not put great effort in addressing the dependum for a depender. The dependee claims about the capability of addressing and delivering a dependum for a depender.

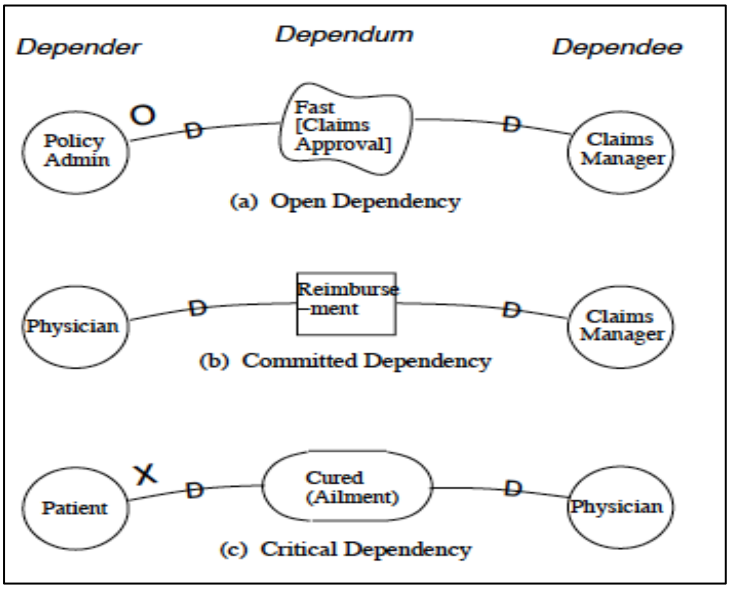


Figure 2.8: The Dependency Strength in the *i Framework.**
[Source: (Yu, 1995), Used with the author's permission].

The second strength degree is committed as it appears in Figure 2.8. A dependee will do a better effort than the open dependency strength by trying its best achieving the dependum. In this dependency, the depender pays attention to its viability.

Finally, a critical dependency is represented using "X" from a depender's side. This strength is the strongest dependency degree, where a dependee must do its best to deliver a dependum to a depnder due to a consequent serious loss on the latter. The depender is not only concerned in this dependency, but about broader ones that are the dependee's dependencies.

Actor categorisation

The *i** framework provides a rich internal structure of an actor that is graphically presented in a circle notation in the external dependencies in the SD model as shown in figures 2.7 and 2.8. The actor is maybe a complex unit that is associated with intentions and requires some refinement, categorisation and grouping into subunits that are role, agent and position.

A role refers to an abstract behaviour with a set of responsibilities. The term role is used in SD model dependencies when it is not necessary to know who plays the role. And then it is possible transferring a responsibility to another unit. An agent refers to an actual physical entity such a human individual (e.g., Ted) or explicit hardware/software (e.g., UWE e-mail system). It is not easy to transfer their behaviour to another entity. A position falls between the role and agent, where a set of roles carried out by one agent. Therefore, a "position covers a role where an agent occupies a position." (Yu, 1995). Finally, an example of the SD model is borrowed from the health care domain is shown in Figure 2.9.

The Strategic Rationale (SR) Model

The SR model elaborates the corresponding SD model through representing the actor's internal structure using GO components that are goals, soft goals, tasks, resources and the relationships between them. GO components are connected with each others using task decomposition and/or means-end relationships within an actor boundary. Types of the GO components and relationships are shown in Figure 2.10. Goal, soft goal, task and resource have been already explained in the means of dependums in the SD model. The task decomposition relationship decomposes a parent task into sub task(s), sub goal(s) and/or sub resource(s). A soft goal participates in the task decomposition relation as a constraint for the task. The means-end relationship refers that a mean, which is usually a task, is required to address the end. Multiple means for an end assist in generating alternatives as a way for improving the addressing of the mean, or as a way of mitigating the risk if another mean did not address the end. The purpose of GO components and the relationships between them contribute to addressing a particular rationale. An example of the SR model is shown in Figure 2.11.

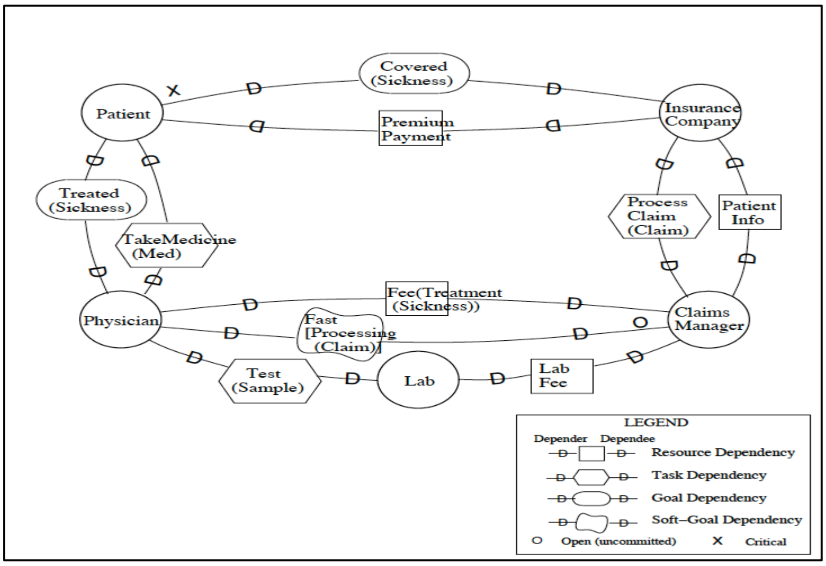


Figure 2.9: An Example of a SD Model from the Health Care Domain [Source: (Yu, 1995), Used with the author's permission]

In Figure 2.11, a physician depends on a claims manager for providing an approval of treatment represented as a resource dependency. The claims manager responds to this dependency via “approve treatment” task that is decomposed using two task decomposition relationships into a sub-goal “treatment be assessed” and sub-task “sign approval document”. An example of a mean-end relationship is illustrated from “assess treatment” task as the mean to “treatment be assessed ” goal as the end.

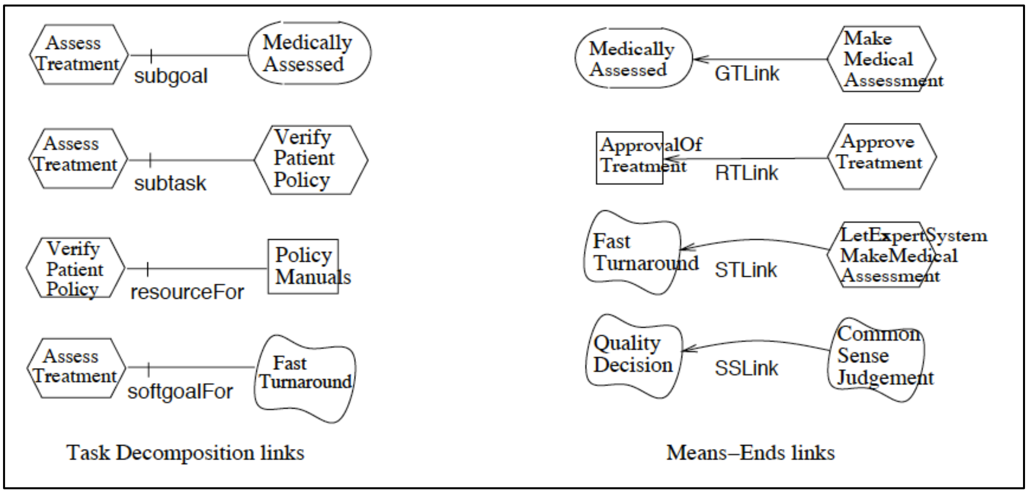


Figure 2.10: The SR model’s relations in the i* framework [Source: (Yu, 1995), Used with the author’s permission].

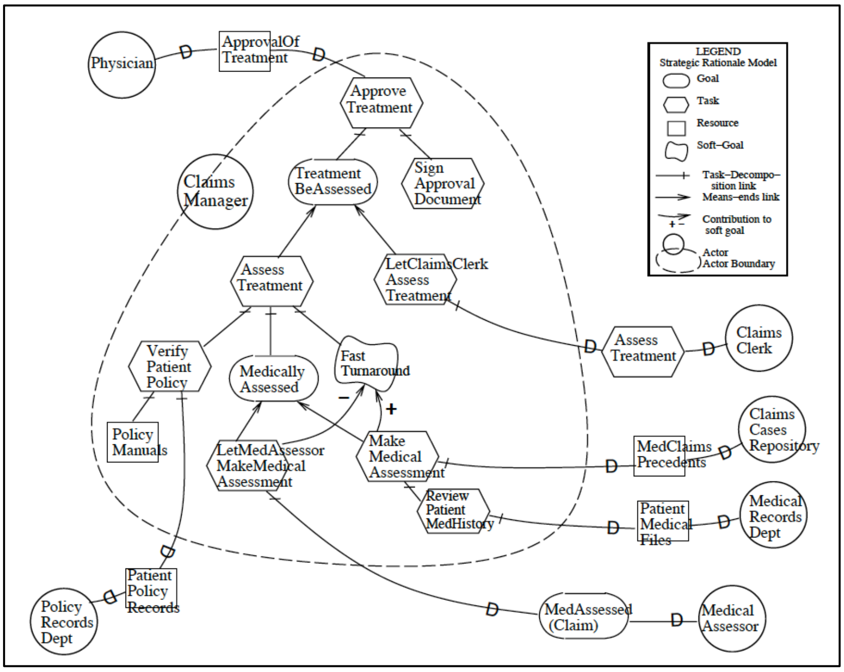


Figure 2.11: SR Model Example from the Health Care Domain [Source: (Yu, 1995), Used with the author’s permission].

2.2.2.2 The NFR Framework (Chung's Approach)

The Non-Functional Requirement (NFR) framework is a soft GO approach that aims to link soft goals (i.e., non-functional requirements) into software systems using Softgoal Interdependency Graphs (SIGs) with full bidirectional traceability. This framework is the most acknowledged quality-oriented approach in software engineering and development (Chung et al, 2000). Therefore, the NFR framework is a candidate research work to participate and contribute to the work in this thesis. A detailed description about this approach is shown below.

Since a soft goal or NFR is not addressed in a clear-cut sense, the generated models from the NFR framework show how a particular NFR is addressed. The NFR framework models provide a rich resource of knowledge and space for stakeholders' communication, collaboration and reuse for engineering and developing NFRs for software systems within an organisation (Chung et al, 2000). An example of a NFR is security designed using the NFR framework is shown in Figure 2.12. In this figure, a security of accounts is considered as an ambiguous quality requirement and requires clarification. Therefore, security is designed using the NFR framework and refined into clearer soft goals until obtaining the approaches or operations that address the security of accounts. Those approaches are presented at the bottom of the NFR framework such as "identifying users" is one of the alternatives designed in order to address the "authorisation access to account information" and consequently addressing "security".

The SIG graph for a particular NFR is generated after carrying out the following steps (Chung et al, 2000):

1) Obtain the required knowledge with regard to the system's domain, its functional requirements and NFRs with their eligible techniques. The required knowledge about the NFRs, with regard to their development techniques, concepts, trade-offs, etc., is elicited from current business documents, industry or related organisations and academia.

2) Identify the top or major NFRs of the system, namely NFR type soft goals. Each soft goal consists of a main NFR (e.g., security, user-friendliness) and an associated subject of matter parameter, namely target (e.g., flight booking). A main NFR is denoted by a cloud notation at the top of the diagram.

3) Refine the main NFR (i.e., NFR type soft goal) downward into sub soft goals (off springs) to some extent systematically, which is determined by the requirement engineers and developers. These soft goals are represented using a normal cloud notation. The offspring soft goals contribute to achieve a parent soft goal in a bottom-up manner using AND or OR interdependencies (i.e., decomposition relationships). Whereas, a refinement or what is called decomposition is processed in a top-down manner. The refinement and the contribution are required to eliminate a NFR meaning's ambiguity. The decomposition process iterates in order to eliminate ambiguity regarding the parent soft goal. This decomposition facilitates the derivation of well-identified solutions that contribute to meeting an ultimate NFR. The AND or OR contribution interdependency link (i.e., decomposition relationships) is used when a set of soft goals are required to fulfil a higher one. For example, in Figure 2.12, a security NFR is decomposed, using AND decomposition relationship, into three sub soft goals that are: "integrity of accounts", "confidentiality of accounts" and "availability of accounts". The security of account NFR is addressed if only the three sub soft goals are addressed.

4) Determine the priority soft goals that must be successfully addressed in the first place.

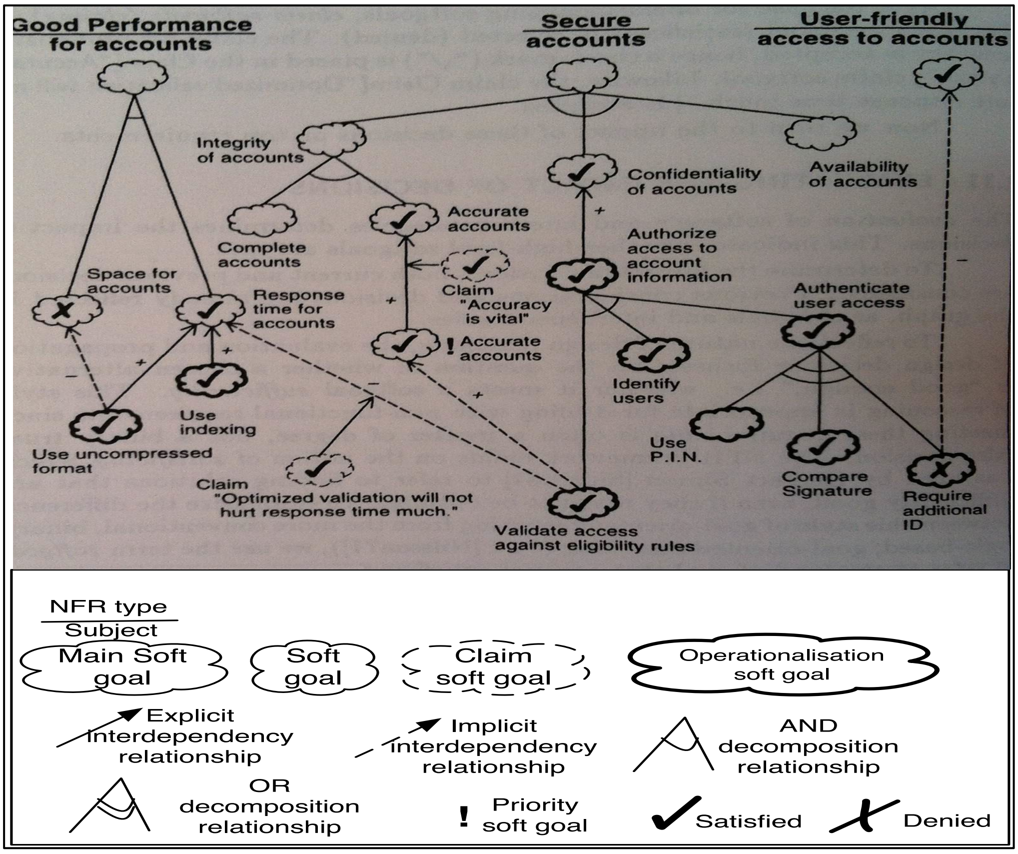


Figure 2.12: The Security of Accounts NFR framework [Source: (Chung et al, 2000), Used with the permission of author and publisher © 2000 by Springer Science+Business Media New York and Kluwer Academic, Non-functional Requirements in Software Engineering, The NFR Framework in Action, 2000, page 38, Lawrence Chung, Brain Nixon, Eric Yu and John Mylopoulos, © 2000 All rights reserved].

5) Derive the operationalisations, which are data, constraints and operations, which actually fulfil the associated NFR soft goal parent. An operationalisation is designed using a thick cloud notation.

6) Deal with both positive and negative interdependencies that are either explicit or implicit between the soft goals and operationalisations. An explicit interdependency relationship is used to determine the contribution of a soft goal to another one locally (i.e., within the one NFR framework diagram). An implicit interdependency relationship is used when there is a need to represent the contribution of soft goals globally (i.e., among soft goals in more than one NFR framework diagram). The positive or negative interdependency is used when a soft goal or an operationalisation supports or harms the fulfilment of another soft goal, respectively. A positive interdependency is either a make (++) or help (+) where the negative interdependency is either a break (--) or hurt (-). Sometimes help (+) is replaced by the term some+ and hurt (-) is replaced by the term some-. For example, Figure 2.12 illustrates a help positive explicit interdependency from an operationalization "use indexing" to the soft goal "response time for accounts" and a negative implicit interdependency relationship between the operationalisation "validate access against eligibility rules" and the soft goal " response time for accounts".

7) If necessary, support the design with rationales, so called claim soft goals, in order to assist in carrying out the design decisions. A claim soft goal associates the interdependencies in order to prioritise or rationalise trade-offs. It is denoted by dashed cloud notation in the SIG.

8) Select available and interesting alternatives that contribute to fulfil a NFR. Alternatives involve operationalisation and/or claim soft goals. Once alternatives are decided, they are considered for the software system design. The selected alternatives would create a history that is anticipated to support in guiding decisions for similar systems that require same NFRs.

9) Evaluate the impact of alternative decisions. The evaluation is carried out using a label propagation approach. That is, the selected alternatives are labelled with satisfied (S), weakly satisfied (W+), unknown (U), conflict (C), weakly denied (W-) or denied (D). The individual alternative soft goal impact values for label propagation are shown in Figure 2.13, where the evaluation of non-individual alternative impact is carried out automatically or guided by the developer decisions. In Figure 2.13, if an individual offspring (presented in columns) soft goal is labelled with "S" and participates within a break (--) explicit interdependency relationship, then the impact of this labels the parent (presented in rows) with "D". Few examples of automatic propagation for non-individual alternative soft goals are shown in Figure 2.14. In

this figure, if two offspring soft goals are labelled with "S" and participated within two explicit interdependency make(++) relationships, then their parent soft goal will be automatically labelled with "S". The rest two examples in Figure 2.14 are interpreted in the same manner.

It is necessary to denote that a decomposition of soft goals is performed downwards into off spring soft goals, where their contribution is carried out upwards to the parent. The aforementioned steps are not sequential. Each step in the framework iterates as far as required.

The NFR framework is described as a soft-goal independent framework, because it adapts with various NFR types, both general and specialised, to produce their SIGs (Chung et al, 2000). It is a systematic method and rich in the quality-based concepts which implicitly embody the quality of software service requirements. Consequently, these features have facilitated an NFR's elicitation and maintaining their SIGs in catalogues for reuse. However, rationales beyond the business that requires the NFRs are not highlighted (first concern). In addition, the engagement of this framework in a business process architecture, and particularly in a Riva business process architecting method, to develop a business-based software services is still absent (second concern).

Evaluation Catalogue								
Individual impact of **offspring** with **label**:	Upon **parent label**, given offspring-parent contribution type:							
	Break (--)	Some -	Hurt (-)	?	Help (+)	Some+	Make (++)	=
D	W+	W+	W+	U	W-	W-	D	D
C	C	C	C	U	C	C	C	C
U	U	U	U	U	U	U	U	U
S	D	W-	W-	U	W+	W+	S	S

Figure 2.13: The individual impact of off springs to parents [Source: (Chung et al, 2000), Used with the permission of author and publisher © 2000 by Springer Science+Business Media New York and Kluwer Academic, Non-functional Requirements in Software Engineering, Softgoal Interdependency Graphs, 2000, page 74, Lawrence Chung, Brain Nixon, Eric Yu and John Mylopoulos, © 2000 All rights reserved].

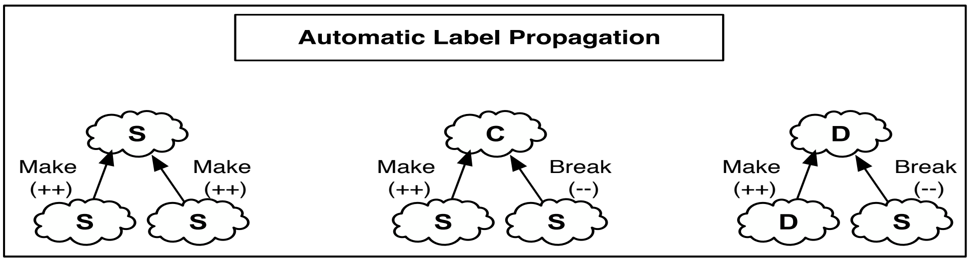


Figure 2.14: Examples of Automatic Propagation for Evaluating Non-individual Offspring Impacts to their Parent [Used with the permission of author and publisher © 2000 by Springer Science+Business Media New York and Kluwer Academic, Non-functional Requirements in Software Engineering, Softgoal Interdependency Graphs, 2000, page 76, Lawrence Chung, Brain Nixon, Eric Yu and John Mylopoulos, © 2000 All rights reserved].

2.3 Process Architectures and Business Process Modelling

A Business Process Architecture (BPA) and its Business Process Models (BPMs) provide a common understanding of the on-going activities, the current main business activities and the way they are connected in an organisation. Dijkman et al defined the BPA as "an organised overview of business processes with their relations and guidelines that determine how they must be organised (Dijkman et al, 2011). Having a business process architecture in place, the business processes themselves can then be modelled in a different stage of the Business Process Management life cycle". Where a BPM can simply be defined as "a set of partially ordered activities intended to reach a goal" (Hammer and Champy, 1993). Both are designed in the form of blueprints that describe the "as-is" environment, and they can be used to assist in designing the "to-be" environment. With regard to the OMG's business architecture key views (OMG, 2012), a BPA can be considered as an approach to represent the OMG's business capability view as shown in Figure 2.15. Similarly, a BPM can be considered as an approach to address the OMG's business operational view shown in Figure 2.16, for its business architecture construction. In this research, concepts that operationalise the relevant OMG's business architecture views and their linkage are highlighted in order to achieve the research objectives.

Below, the significance of having the BPA along with its BPMs is listed:

- The "as-is" BPA and its associated BPMs form a foundation of a desired "to-be" architecture and models. The "as-is" design paves the way to construct the desired improvements, as it embodies the absence of the required processes or activities.
- Recently, the harmony between the BPA and the BPMs encouraged the derivation of candidate software services (Yousef, 2010). Therefore, their role does not only serve the purpose of addressing a common understanding and communication between business-related stakeholders.
- The existence of a BPA and BPMs is necessary to achieve their alignment with the strategic models for an organisation (Silvius, 2007).
- Since they both represent the on going (i.e., continuing) and the current (i.e., present) activities, then the belated business requirements will be exploited in a one way or another from them. Hence, this facilitates early design decisions and an initiation of an early implementation of these requirements.

Researchers debated BPA modelling approaches and attempted to classify them in order to guide practitioners in selecting the appropriate approach according to their domain needs and interests. A recent empirical investigation was carried out to ascertain the BPA designs employed in practice (Malinova et al, 2013). It resulted in a conceptual framework that embodies factors stemming from the domain of practice in order to address the appropriate design required for business needs and development (Malinova et al, 2013). Thus, they attempted to classify the existing approaches as shown in Figure 2.16. (Yousef, 2010) classified the BPA approaches based on their methodological perspective, where (Dijkman et al, 2011) classified them into five categories based on their structure types regarding the way the processes and their relations are identified.

From the reviewed literature, the detailed representation of process models is not useful unless they are architected for the following situations:

- For some reason, the stakeholders need a representation of the process type classifications along with the relation between the processes. For example, (Ould, 2005) proposed the Riva BPA that explicitly addresses the process classifications, as explained in Section 2.3.1.
- A stakeholder or an architect may be interested in discovering the insourcing (i.e., supplied and operated within an organisation) and the outsourcing (i.e., operated within an organisation, but from an outside supplier) processes in the organisation. The best way is to refer to the BPA, which explicitly provides this representation. For example, the Riva BPA method supports this kind of representation and denotes the outsourcing in a BPM by an outside relation from a cloud notation.

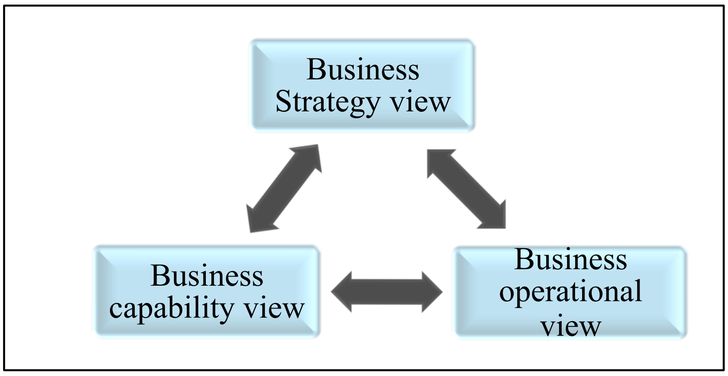


Figure 2.15: The OMG's View of the Strategic, Business Capability and Business Operational Views.

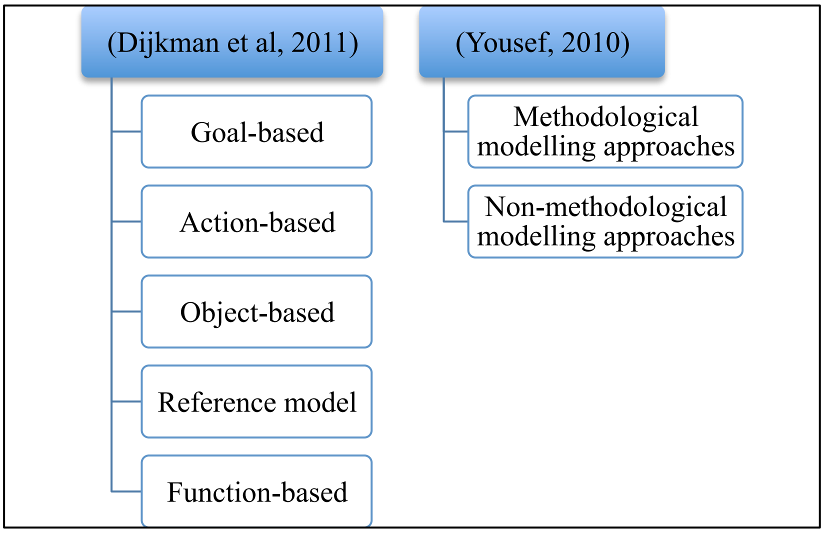


Figure 2.16: The BPA Classifications in the Literature.

- A BPA facilitates an early agreement and representation of the cohesion processes operating in the organisation. In addition, the dependency or the connecting relations among processes in their corresponding architecture generate an easy to view traceability network of the processes.

In the light of the above, a BPA provides a systematic organisation and classification of process models that steers the way for their required redesign in order to meet business needs. Kavakli and Loucopoulos (Kavakli and Loucopoulos, 1999) proposed a goal-driven systematic approach to model business processes by putting forward business goals and objectives. Current BPA modelling approaches have concentrated on activities and tasks (i.e., what must be done) rather than business processes (i.e., how it is done). They illustrated their approach as shown in Figure 2.17. Their approach is considered as a part of large enterprise knowledge modelling framework known as the Enterprise Knowledge Development (EKD). The goal graph shown in Figure 2.17 adopts the stakeholders' objectives in an enterprise and

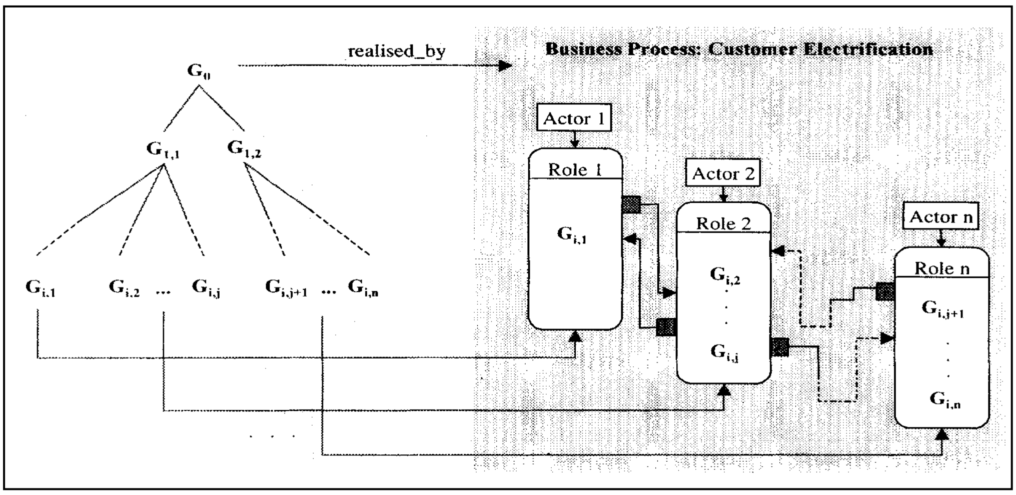


Figure 2.17: Goals and Objectives are Realised by Business Processes [Source: (Kavakli and Loucopoulos, 1999), Used with the author's permission].

refines them to low-level goals namely, operationalisations. The resulting operationalisations are mapped to role-oriented process models to satisfy goals. However, business enterprise elements and entities are not explicitly determined. This would leave a gap between goals and the business activity workflow.

Barros and Julio proposed a method for BPA design based on four essential types of macro processes that are: value chain macro processes, development macro processes, planning macro processes, and management macro processes (Barros and Julio, 2011). They call BPs as macro processes, as each of them is a collection of processes. They claim that enterprises' processes must be extracted from those four macro processes that are related with relationships in order to derive what is called enterprise process architecture. The architecture and the processes where designed using Business Process Modelling Notation (BPMN). This method has been evaluated in practice and particularly in the health care domain (Barros and Julio, 2011).

From the previous literature, it is apparent that the most followed notion in the BPA design is essentially based on composition (i.e., linking relevant processes) and decomposition (e.g., low-level, high-level and in between operating processes). However, the way the BPA is designed varies from one to another based on the type of the organisation and needs (Malinova et al, 2013).

The modelling approaches for BPAs vary in the literature based on a motif-orientation (e.g., goal-based and/or object-based) of the BPA derivation. For example, if an organisation is interested in establishing a BPA from the fundamental business blocks perspective, then it is preferable to adopt the object-based BPA modelling approaches such as the Riva method. In this thesis, it is claimed that a reengineering of a BPA is required in order to align it with business/IT needs. The usefulness of a BPA model is explained in the following reasons:

1) The big growth of need to improve an as-is BPMs motivates the designers to abstractly elicit and identify the required improvements from the early stages in order to put them in the BPM life cycle for development afterwards. A BPA model is a candidate place to improve the as-is BPMs.
2) A BPA model is a good example of a blueprint. In other words, an organisation may require the execution of a particular set of interrelated business processes, which are already identified in their architecture. Thus, the organisation has the potentiality to reuse the blueprint in order to reduce the effort (i.e., time and cost) allocated another similar BPA design.

3) The BPA design activity does rationally initiate the business process management life cycle.
4) The kind of concepts used in a BPA model have the capability to engage with the associated earlier related models (e.g., goal models), and as well with the associated later related models (e.g., business process models). This is because of the strategic place that a BPA model occupies in the discipline and its ability to bridge the gap with map-able models (i.e., models that have the possibility to map their concepts with others).

2.3.1 The Riva BPA Method

The Riva BPA modelling method proposed by Ould (Ould, 2005) was classified as an object-based methodological approach (i.e., systematic) in (Dijkman et al, 2011). Recently, researchers have become more interested in involving the Riva method in their work through using it to generate the enterprise information architecture (Ahmad and Odeh, 2012). Also, the Riva BPA method showed an essential contribution in the service-oriented environment (Yousef, 2010). Furthermore, few researchers attempted to transfer the architecting activity of BP into real practice (Ould and Green, 2004).

Ould has considered a BPA as a blueprint of the overall chunking of interrelated business processes, that each of them has a lifetime, which the organisation is interested in (Ould, 2005). According to Ould, a BPA identifies all of processes in an organisation and the relationships between them. These processes could be running concurrently and each is initially derived from an Essential Business Entity (EBE) of an organisation's business domain. Ould recommended a good business process division in a business organisation, where the business processes are dynamically related. His recommendation is proposed in order to avoid complex process design and to increase maintainability for further modifications due to the agile nature of such organisations. He claimed that organisations running same business can have the same BPA. Figure 2.18 illustrates a summary of the Riva-based BPA construction process.

As this research aims to reuse the work of Yousef (2010), then the Riva method deserves a good literature space for its detailed explanation among other alternatives. The Riva steps, illustrated Figure 2.18, are individually explained in detail along with an example adopted from a Riva BPA modelling workshop (Ould and Green, 2004).

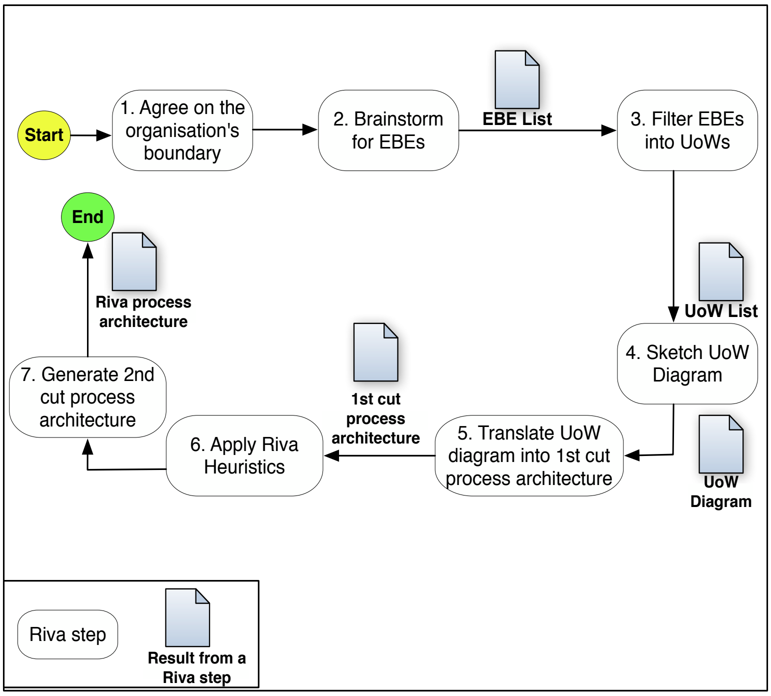


Figure 2.18: The Fundamental Steps for Riva-BPA Modelling Process.

Step 1: identify the present organisation and agree on the business boundary. A boundary need not necessarily be around the whole organisation. For example, Administration of CEMS Faculty in UWE.

Step 2: brainstorm for Essential Business Entities (EBEs). An EBE is identified as the entity that substantially characterises the business that the organisation is in and forms a subject of matter of the business. The term essential is used in the EBE because the architects must be interested and/or reminded that they are involved in searching for the entities that are the object of the business's essence and nothing else. In reality, an EBE may represent a concrete or physical entity (e.g., letter, exam paper and award handbook) in the agreed business boundary of the agreed organisation. The EBE may also represent something abstract (e.g., graduation day, award and module run), or even entirely abstract (e.g., student request to transfer award). In this step, the architect must be able to distinguish between the EBE and between the Designed Business Entity (DBE) that embodies the way the stakeholders chose to do the business, and it is not a subject of the matter. It is easy to distinguish between an EBE and a DBE if the entity represents an example of a matter. For example, award handbook is an EBE but BSc award handbook for the environment faculty is a DBE, as it represents an instance of its corresponding EBE. Another example, online submission is the way the stakeholders have chosen to do the business of the EBE submission, therefore, online submission is a DBE. Ould proposed filters for the EBEs in order to revisit the list for EBEs rechecking, that is by putting 'a' or 'the' in front of the candidate entity, and if it does not make sense, then it must be taken out of the EBE list. Another filter is by being aware about the essence of the entity in the business through thinking if it forms the food and the drink of the business. If the entity is a role and it is not part of the essence of the business, then it must not be an EBE. The output from this step is the EBE list for the CEMS Faculty example with 94 EBEs as attached in Appendix A.

It is necessary to denote that this step was found difficult by the practitioners as it depends on the brainstorming activity by gathering group of people and starting to think loudly. Recently, a BP-based algorithm is proposed in (Yousef, 2010) in order to tackle the brainstorming difficulty, as she depended on exploiting the EBEs from the role-oriented BPMs of the agreed organisation that requires the design of its Riva BPA model. Although it is another approach of identifying EBEs, it is limited at the presence of those BPMs.

Step 3: Classify the EBEs resultant from the step above into entities that possess an interesting lifetime for the organisation and that are not. Call the ones with lifetime as Essential Unit of Works (E-UoW) or just (UoW) and bracket the rest of the EBEs. If an EBE is too small, or its within another EBE and does not have interesting lifetime, then it must be bracketed, as it is not a UoW. The outcome from this step is the UoW list in order to sketch he UoW diagram next. The UoWs appear in bold text (31 UoWs out 94 EBEs) where the EBEs are bracketed in Appendix A for the UWE's CEMS Faculty of Administration example.

Step 4: Sketch the UoW diagram by creating the dynamic relationships of "generate" between any two related UoWs. A UoW is represented using a hexagon. Any UoW that appears to be outsourced is generated from the outside world using the cloud notation. However, a UoW may not be related to another one or outsourced. An architect may wish to associate a dynamic relation with cardinality (e.g., one-to-one or one-to-many). The generated UoW diagram for the CEMS Faculty of Administration is attached in Appendix.

Step 5: Producing the 1st cut architecture from the UoW diagram via determining the relationship type between any two related UoWs. In particular, Ould hypothesised that any dynamic relationship (i.e., generates) in a UoW diagram is either a task force or service function. In addition, each UoW corresponds a Case Process (CP), Case Management Process (CMP) and Case Strategic Process (CSP). A CP represents instances from the original process, where the CMP manages the function of those instances. The CSPs are omitted unless they are part of the interest. A translation of the task force relationship and the service function are shown in Figures 2.19 and 2.20 respectively. An example of 1st cut architecture resulting from the UoW diagram is attached in Appendix A.

Step 6: Apply the Riva heuristics. Ould advised an architect to follow a set of heuristics and apply them on the 1st cut architecture in order to produce the 2nd cut architecture that represents the Riva BPA model. The heuristics are:

1- **Folding a task force CMP into the requesting CP**: in the task force relationships, the CMP is encapsulated into its requesting CP yielding to two CPs, where one initiates the request and another is delivering the request after being processed.
2- **Dealing with 1:1 'generates' relationships:** this case involves four situations that are as next. First, if the 'generates' relation is a service function, then all CMPs of the CPs should exist in the 2nd cut architecture. Second, if a process is a large process, has its own lifetime and cannot be worked within another process, then the CMP must exist in the BPA model. Third, an architect is advised to merge two UoWs into one in case that both of them are turned into one UoW after a while. In this case, Ould claims that their CMPs are undistinguishable and thereby unneeded. Therefore, a one

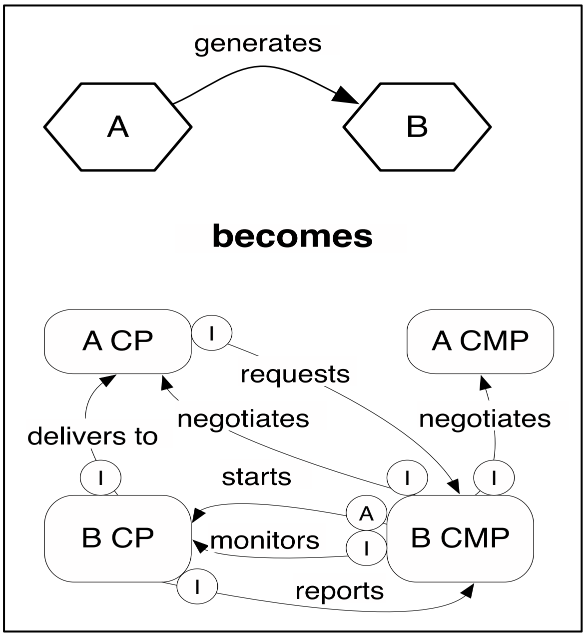


Figure 2.19: Translating the Dynamic Relationship 'generates' into Service Relationship.

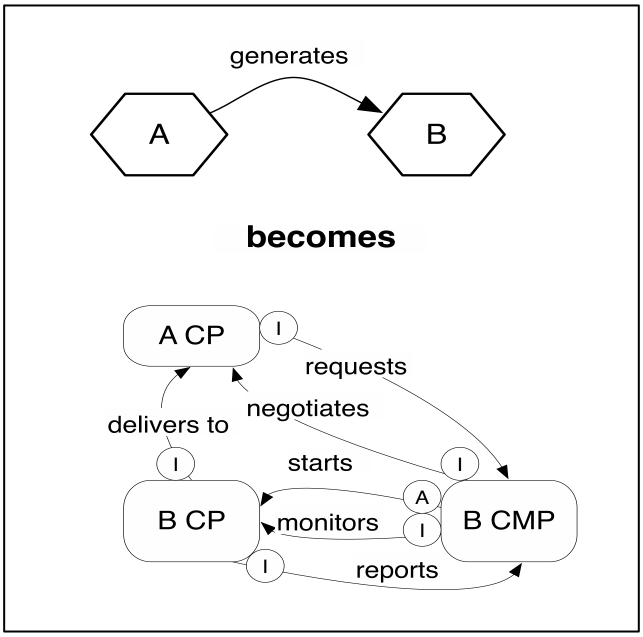


Figure 2.20: Translating the Dynamic Relationship 'generates' into Taskforce Relationship.

UoW that yields one CP and one required CMP is the best design option available. Finally, one UoW generates none or only one another UoW taking into the account the need of both their CPs and CMPs, as each of them represents a separated lifetime. In this case, all CPs and CMPs for the two UoWs are required to exist in the Riva BPA model.

3- **Dealing with delivery interactions and delivery chains:** for some reason a chain of delivery relations exists between CPs because of the chain of the generate relationships. In this case, Ould suggested short-circuiting the chain by making the delivery from the first latest CP to the first requesting CP.
4- **Dealing with collections:** the architect should be able to distinguish between the lifetime of an individual UoW and their collection. In particular, the collection of those UoWs requires a handling for their lifetime. In this situation, the UoW that represents the collection of the individuals must design its CP and CMP in the BPA.
5- **Dealing with empty CMPs:** It is rare to have empty CMPs. In particular, this happens when a case management of the instances is unrequired specially when the instance of the case process is only one.

Step 7: Arrange and gather what has been generated from Step 6 in order to have the 2^{nd} cut architecture, which is the ultimate Riva-BPA model. The resulted 2^{nd} cut architect for the CEMS Faculty Administration example is attached in Appendix A.

Recently, the Riva method was formally represented (i.e., ontologised) using Ontology Web Language- Description Logic (OWL-DL) ontology language, namely semantic riva BPA (srBPA) Ontology in (Yousef, 2010). The purpose of this ontology is to conceptualise a Riva-based BPA for a particular organisation. The srBPA ontology was originally developed within the BPAOntoSOA framework and evaluated in (Yousef, 2010). In addition, it is characterised as a domain independent ontology and was evaluated using an education and health care domains.

2.3.2 The Business Process Modelling

Researchers attempted to find and agree a definition for business processes stemming from their common characteristics (Lindsay et al, 2003) (Ould, 1995). They distinguished between the notions of processes in the environment (i.e., production processes, office processes, coordination processes, machine processes, material processes, information processes and business processes) (Lindsay et al, 2003). A few researchers recognised the absence of a standardised techniques in the BP management and they allege that, "*this lack of standarisation in representing a business process is in part caused by the absence of an* *adequate definition of a business process*" (Lindsay et al, 2003). In the BP management community, one group concentrated on the goal of the BP (i.e., what is to be accomplished) (Hammer and Champy, 1993) and another group concentrated on the flow of work of the BP (i.e., how it is accomplished) (Eriksson and Penker, 2000). The BPMs were proposed in order to simplify the representation of the real environment in order to improve the current activities taking into the account that the real environment is very complicated and not static (Lindsay et al, 2003).

The importance of the BPM stem from the following needs:

1) The BPMs assist the stakeholders at arriving to a common and detailed understanding of the current processes and agreeing on the roles, activities and deliverables. The understanding of a current BP using the model assists in carrying out the process correctly. Recently, BPMs were employed as an approach in order to investigate the complexity of processes (Abu Rub and Issa, 2012). This has delivered a further deep understanding of the body of the process.

2) A current BPM aids the mangers and designers in improving a current BP. In particular, a BPM directs them in a one way or another to identify problems, threats, opportunities and strengths. Accordingly, this alerts designers for a redesigning or reengineering process for the BP itself and its models taking into the account remaining the previous models for reuse in the future.

3) BPMs have been used to derive the functional and the non-functional requirements for software systems (Yousef, 2010) (Biittencourt et al, 2010) (Jaramillo, 2011)

BPM approaches are categorised according to their perspective into (Kavakli and Loucopoulos, 1999):

- Activity-oriented approaches: they concentrate on the workflow of business tasks, their order and any conditions such as UML Activity Diagram, International Defence Industry Fair (IDEF0, 1993), etc.
- Role-oriented approaches: those are approaches that group business activities and responsibilities to relate them to a particular role that could be a person or system that usually collaborate to achieve goals such as RAD (Ould, 2006) and BPMN (OMG, 2011).
- Product-oriented approaches: these approaches gather their activities based on the output with ignoring the behaviour of the organisation and its associated goals and processes (Kavakli and Loucopoulos, 1999).

- Speech act-oriented approaches (Kueng et al, 1996). The speech act-oriented approaches are based on the language or the theory of speech act (Kueng et al, 1996) (Aburub and Issa, 2012).
- Goal-based BPs (Nurcan et al, 2005). Finally, the goal-based approaches identify the BP as a set of goals that consist of a set of strategies (Nurcan et al, 2005).

In this research, the role-oriented business process modelling is chosen so concerns are separated and easily handled. This separation also increases the understandability and reusability of roles in business models. Also, it is anticipated to trace any changes that occur in a software system backward to roles in BPMs and goals that are depending on. Finally, the role-oriented models of this research's case study (i.e., CCR) are already modelled in (AbuRub, 2006) (Yousef, 2010) which paves the way to utilise and optimise them afterwards in order to satisfy research aims. One BPM modelled using the BPMN language example from the CCR is shown in Figure 2.21. Finally, it is advised to adopt a 'responsive' and 'adaptable' BPs in order to survive in the business that organisation is in (Lindsay et al, 2003).

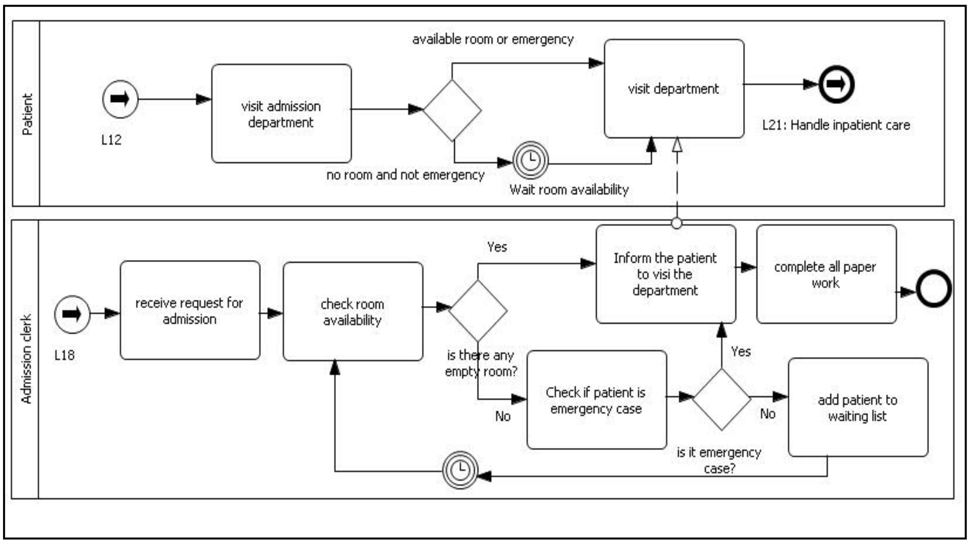


Figure 2.21: Handle a Patient Admission BPM Using the BMPN [Source: (Yousef, 2010), Used with the author's permission].

2.3.2.1 BPM Semantic Approaches and Applications

Researcher are moving towards choosing ontology to semantically represent their BPMs in order to address their aims and benefiting from the two advantages that ontology provides: interoperability and machine reasoning (Yousef, 2010) (Dobson et al, 2005). Also, ontology works as an organisation for the complex knowledge that is described using classes, restrictions and associated properties. This ontology-oriented organisation allows maintaining required knowledge in one repository and facilitates communication between stakeholders.

With regard to the attempts to ontologies BPM approaches, the SUPER project, which is an EU 6th framework programme funded project, was founded for the major objective "to raise Business Process Management (BPM) to the business level, where it belongs, from the IT level where it mostly resides now. This resulted in development of tools enabling deployment of Semantic Business Process Management." (SUPER, 2008 b).

In deliverable 4.5 of the SUPER project, the sBPMN ontology was important middleware required to map the XML Process Definition Language (XPDL) into the Business Process Modelling Ontology (BPMO) (SUPER, 2008 a), "which provides a high-level model of business processes, integrating organisational aspects, process workflow and services. The goal is to support a number of BPM life-cycle activities at the semantic level, including modelling, querying, translation and execution" (Cabral and Domingue, 2009). The BPAOntoSOA framework borrowed the sBPMN ontology and employed it in the sBPMN ontology instantiator component in order to conceptulise the BPMs of the Riva BPA (Yousef, 2010).

The sBPMN ontology comprised of a "..hierarchy of concepts along with its attributes and a set of axioms that allow to automatically check if a business process diagram is well-formed" (SUPER, 2008 a). Furthermore, the BPMO was translated into Business Process Execution Language (BPEL) in order to bridge the gap between business people and IT people and to allow them to navigate between the two views (Cabral and Domingue, 2009).

2.4 Service-Oriented Computing (SOC)

Service-Oriented Computing (SOC) is increasingly becoming the mainstream for developing complex distributed software services. According to Erl's (Erl, 2007), "Service-oriented computing represents a new generation computing platform that encompasses the service-oriented paradigm and service-oriented architecture with the ultimate goal of creating and assembling one or more service inventory". Erl considers Service-Oriented Architecture (SOA) as " a distinct form of technology architecture designed in support of service-oriented solution logic which is comprised of services and service compositions shaped by and designed in accordance with service-orientation".

Services are developed as language and platform independent and reused by different systems considering location transparency (Erl, 2007). (Papazoglou et al, 2007) have illustrated SOC research as a roadmap in Figure 2.22, and considered the SOA as a logical approach to realise SOC. Another attempt in (Sommerville, 2007), who considered SOA as a new technology for developing distributed applications, where stand-alone services are fundamental components as shown in Figure 2.23.

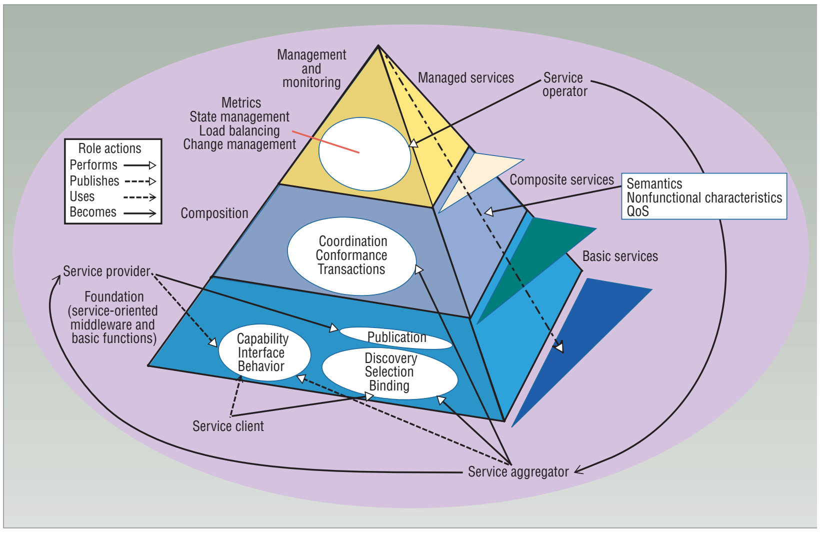


Figure 2.22: SOC Research Roadmap [Source: (Papazoglou,2007), Used with the author's permission]

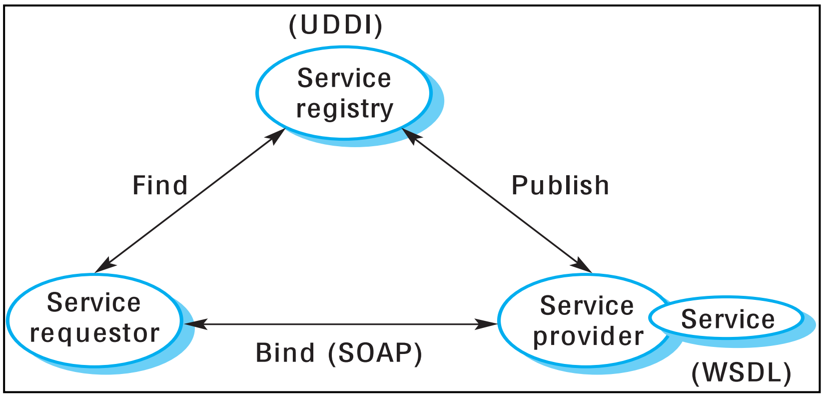


Figure 2.23: Service-Oriented Architecture [Source: (Sommerville,2007), Used with the author's permission].

Sommerville defined a service as "a loosely coupled, reusable software component that encapsulates a discrete functionality, which maybe distributed and programmatically accessed" (Sommerville, 2007). Web services are a promising technology that realise SOA and use the Internet as a medium for distributing and offering interoperability. He defined a wed service as "a service that is accessed using standard Internet and XML-based protocols". Web services (Sommerville, 2007) are implemented using open standards such Web Service Description Language (WSDL), Universal, Description, Discovery and Integration (UDDI) and Simple Object Access Protocol (SOAP). These three standards are based on as EXetnsible Markup Language (XML) that is "a human and machine-readable notation that allows the definition of structured data where text is tagged with meaningful identifier" (Sommerville, 2007).

In an organisation, a business process stimulates the identification of software services using Service Identification (SI) approaches (Sommerville, 2007). SI is captured by deriving requirements from the understanding and analysis of associated business process in order to obtain a logical solution (Erl, 2007) that assists in deriving the functional requirements. All of a service description is implemented and specified using WSDL. A WSDL specification consists of three parts (Sommerville, 2007). The first part is interface that deals with specifying what services are provided and what format is used in the sent and the received message by the service. The second part in binding that specifies how to communicate with the web service. The third part specifies the location of the web service using Universal Resource Identifier (URI). An abstract picture of WSDL is shown in Figure 2.24. Nevertheless, WSDL lacks describing NFRs and service semantics (Sommerville, 2007) that are aimed to achieve in this research. The BPAOntoSOA framework proposed a new service identification method from a Riva-based BPA, but this framework neglected the integration of quality requirements into the modelled services.

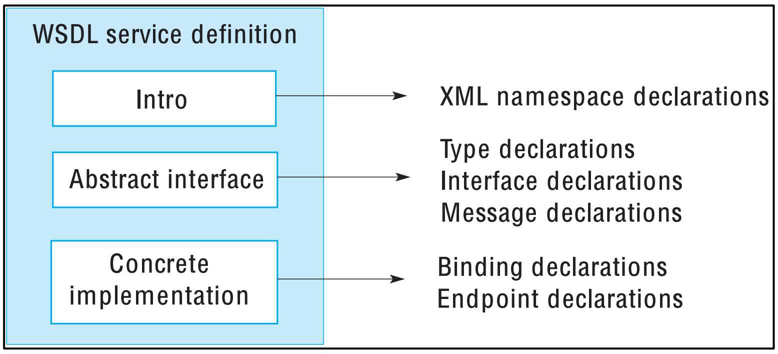


Figure 2.24: A WSDL Specification [Source: (Sommerville, 2007), Used with the author's permission].

Since SI approaches lack quality integration, this research will utilise the BPAOntoSOA framework SI approach regarding considering goals and quality requirements in deriving the identification of candidate software service's functional and non-functional requirements.

The components of a service specification are defined using UDDI that permits the users to discover the availability of a service. UUDI provides information about service provides, service provided by the service and its location, where those information are already described using WSDL (Sommerville, 2007). SOAP is a standard that specifies how messages are interchanged in order to support the communication between services for binding purposes. Finally, a business process model designed using BPMN can be described in the form of low-level language (i.e., XML-based) using the standard Web-Service BPEL (WS-BPEL).

Single services are composed and integrated to carry out a function of complex services that present a business process or workflow in a particular organisation. In order to develop high quality SOA solutions, ERL recommended that the following design principles should be followed (Erl, 2007):

1) Services share standardised contracts: A service contract consists of a technical and non-technical service contract. Technical contract consists of one or more service description documents defined by specific standards such as WSDL for web services associated with XML and attached WS-Polices. A contract for a service is referred to as " Terms of engagement, providing technical constraints and requirements as well as any semantic information the service owner wishes to make public ". Sharing a standardised contract would support or constraint the service discoverability in the first place (Erl, 2007).

2) Services are loosely coupled: Coupling between software services refers to a relation between components. This principle is addressed through minimising the dependency relationships. This would maximise the chance for service providers to easily evolve their services without affecting consumers (Erl, 2007).

3) Services abstract underlying logic: this principle means hiding the not absolutely required information within the right balance that yields the service to be a black box view. This abstraction avoids unnecessary access to service details. This principle concerns only the level of abstraction (Erl, 2007).

4) Services are reusable: this principle encourages reusing a service. Service reusability brings high returns such as maximising the chance of availability and scalability. In addition to building less and using more. That is, reusability of services pave the way to lessen the effort required for developing another similar related service (Erl, 2007).

5) Services are autonomous: This means that services own a self-governance capability by increasing the self-control in a runtime environment to increase reliability and behaviour predictability (Erl, 2007).

6) Services are stateless: this mans that the consumption of resources should be minimised by postponing the management of unneeded information for the current activity. This encourages increasing scalability in a SO environment. This principle is also intended to reduce usage of resources that would minimize state management. Therefore, services are designed stateful when needed (Erl, 2007).

7) Services are composable: This refers to the ability to effectively repeatedly aggregate with services to form a service-oriented solution. The composition should be carried out regardless the complexity of the composition and the size of the service (Erl, 2007). The separation of concerns principle importance is manifested when a composition is carried out.

8) Services are discoverable: this principle requires a service to be queried by a user or service to address discoverability and this should be considered during service design and analysis. It reduces redundancy for a service that provides same functionalities, thus yielding a cost and effort saving (Erl, 2007).

In this thesis, the above principles should be satisfied while deriving software services from high business goals and Riva-based BPA in order obtain a well-identified and designed software service ready for distribution.

2.5 Non-Functional Requirements (NFRs)

Kotonya and Sommerville considered Non-Functional Requirements (NFRs) as "restrictions and constraints among system services" (Kotonya and Sommerville, 1998). Kotonya's and Sommerville's NFR classification is considered as the most comprehensive because any classification of NFRs can be refined and ought to cover the following three categories: product requirements, process requirements and external requirements. The first category specifies the desired and recommended attributes that a system ought to possess. Any constraints and restrictions on the development process over the system will be under the second category. Finally, external requirements specify any organisations' regulations, facts, standards and others derived from system environment and placed on product and process.

International Organisation for Standarisation (ISO) (ISO/IEC, 2001) has classified service qualities into six quality characteristics that are divided further into sub-characteristics as technical characteristics as depicted in Figure 2.25. Each of the sub-characteristics could be measured with a specified metric.

In 2009, (Lamsweerde, 2009) classified NFRs as Quality of Service (QoS), compliance, architectural constraints and development constraints, as shown in Figure 2.26. The QoS is similar to such quality attributes as security, performance, etc. Compliance means to conform standards, an organisations' regulations, laws and facts, and architectural constraints means to

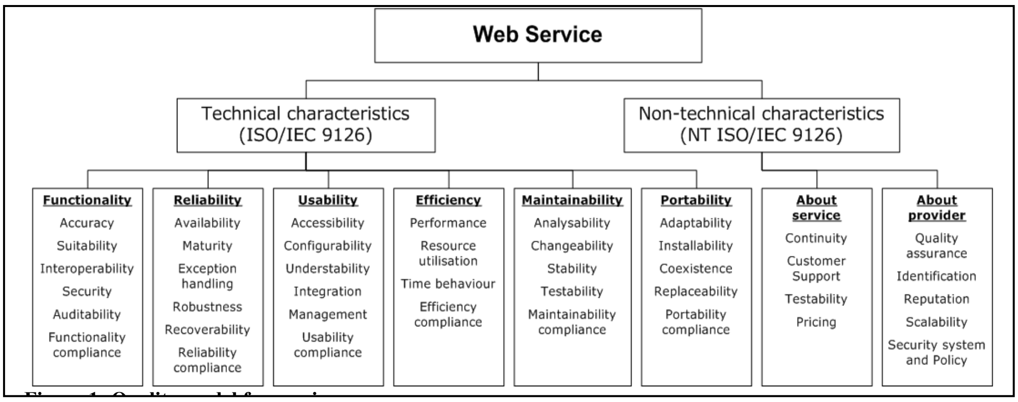


Figure 2.25: Quality Service Model [Source: (Ameller and Franch, 2008), Used with the author's permission].

apply structural constraints on the developed software to fit the environment. Finally, development constraints restrict aspects of the software development such as maintainability, delivery schedules, etc. Also, this classification can be considered as a comprehensive categorisation of NFRs.

2.5.1 The NFRs in the GO Approaches

The work of the NFR-framework (Chung et al, 2000) that has already been presented in Section 2.2.2.2 is a process-driven approach, where soft goals (i.e., NFRs) are decomposed and refined to sub-soft goals in order to obtain a low-level operationalisations of soft goals. Satisfaction of NFRs is not discrete but can vary in magnitude such as "satisfied", "weakly satisfied", "unknown", "conflict", "weakly denied" or "denied" (Chung et al, 2000). In addition, this approach illustrates trade-offs between soft goals and priority. This framework is considered the most comprehensive approach for determining NFRs satisfaction. (Burgess et al, 2009) optimised SIGs to get Soft goal Interdependency Rule set Graph (SIRG) as new automated technique for determining the optimal set of the low-level operationalisations to gain a better NFR satisfaction. However, SIRG is not evaluated yet.

In addition, the *i** framework employed NFRs in the form of soft goals for the understanding of the business of an organisation (Chung et al, 2000) (Yu, 1995). It follows a similar notion of the NFR framework for refining soft goals, yet the latter is essentially designed to integrate the NFRs into software systems rather than being interested in addressing the understanding of the business.

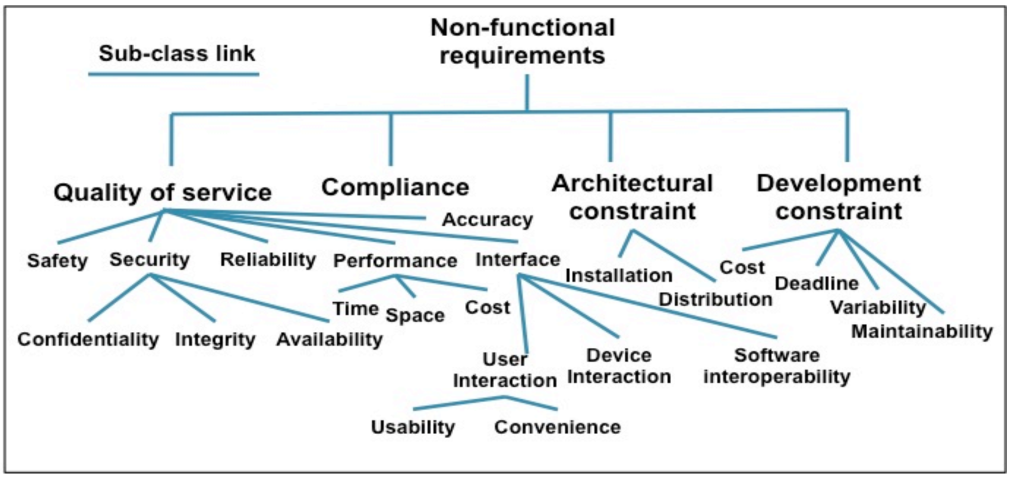


Figure 2.26: Non-Functional Requirements on Software Services [Source: (Lamsweerde, 2009), Used with the permission of the author and Wiley Publisher: Permission is hereby granted for the use requested subject to the usual acknowledgements (Axel van Lamsweerde, Non-Functional Requirements on Software Services, Requirements Engineering: From System Goals toUML Models to Software Specification, Wiley Publisher) © 2009 John wiley and sons ltd All rights reserved].

2.5.2 NFRs in the BPA and BPM Approaches

Current approaches for integrating quality requirements into BPAs and BPMs are almost absent. (AbuRub, 2007) employed the NFR framework approach in his research work and (Pavlovski and Zou, 2008) integrated NFRs to role-oriented BPMs. In addition, there is no current work on identifying QoS requirements for or from BPAs. However, the absence of research work regarding the integrating of quality requirements into a BPA does not necessarily mean their absence in practice. It seems that they are not widely addressed in the research community due to the very few allocated current attempts that if existed, they are not very well structured to address the NFRs in the BPA and BPMs. Researchers have proposed a number of approaches to identify quality requirements from high-level business goals. For example, (Plosch et al, 2010), (Clements and Bass, 2010) and (Bittencourt et al 2010) have used a goal-oriented approach to identify quality requirements.

One work proposed a classification of Quality on Business Processes (QoBP) (Heravizadeh et al, 2009) that is categorised into four categories of NFRs as shown in Figure 2.27. For example, in the function category in Figure 2.27,“ *A function is a basic building block in a business process that corresponds to an activity (task, process step) which needs to be executed*” (Heravizadeh et al, 2009). Security is a quality attribute for a BP function quality that is defined as “the capability of the function to protect information and data so that unauthorised resources cannot access them” (Heravizadeh et al, 2009).

Function	**Input/Output**	**Non-Human Resource**	**Human Resource**
Suitability	Accuracy	Suitability	Domain Knowledge
Accuracy	Objectivity	Accuracy	Qualification
Security	Believability	Security	Certification
Reliability	Reputation	Reliability	Experience
Understandability	Accessibility	Time Efficiency	Time Management
Learnability	Security	Resource Utilization	Communication Skills
Time Efficiency	Relevancy	Effectiveness	
Resource Utilization	Value-added	Safety	
Effectiveness	Timeliness	User Satisfaction	
Productivity	Completeness	Robustness	
Safety	Amount of Data	Availability	
User Satisfaction			
Robustness			

Figure 2.27: The quality Dimensions of Business Processes [Source: (Heravizadeh et al,2009), Used with the author’s permission].

2.5.3 Non-Functional Requirements and the Service-Oriented Model of Computing

NFRs are referred to as Quality of Service (QoS) in the service-oriented context. (O'Brien et al, 2007) have shown considerable effort in determining ten different quality attributes that affect SOA, observing issues both with each quality and with the recommended associated solutions in order to satisfy the quality characteristics. The research in this thesis is anticipated to determine different quality attributes on the derived services.

Ameller and Franch suggested a Service Level Agreement Monitor (SALMon) as a monitoring technique on SLA (Ameller and Franch, 2008). This work reported on the ISO/IEC 9126 based classification of the desired characteristics in relation to web services as shown in Figure 2.25 with the emphasis on the technical and non-technical characteristics of such services.

A taxonomy for NFRs along with service-centric systems is presented in (Galster and Bucherer, 2008). Their taxonomy is likely to be checklist rather than a guideline using three categories: process requirements, NF service requirements and NF external requirements as shown in Figure 2.28. This work featured the ability to apply the taxonomy on service level (fundamental components) as well as system level (composed services). Moreover, Galster and Bucherer suggested to formally or informally integrate their proposed taxonomy into the requirements engineering process as a template.

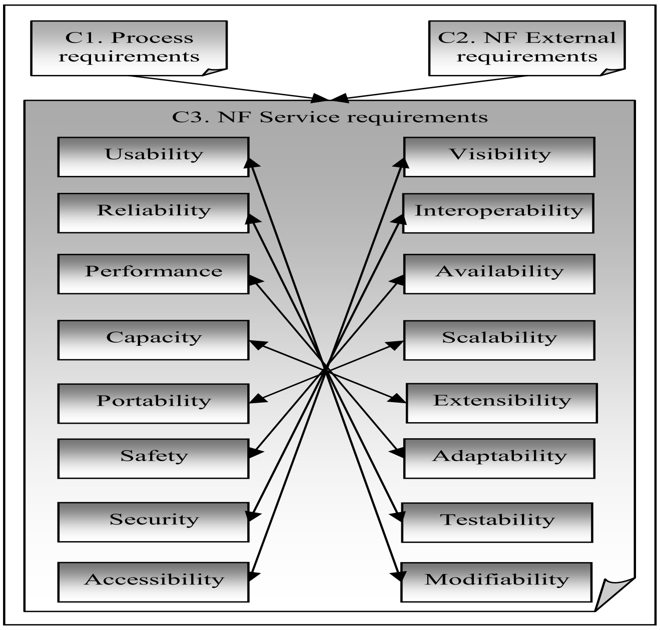


Figure 2.28: NFRs Taxonomy and Potential Interaction [Source: from (Galster and Bucherer, 2008), Used with the author's permission].

2.6 Business and IT Alignment

Considerable work has been carried out in business and in IT to develop an alignment method that fits with the strategies and operations of organisations and associated systems (Aversano et al, 2012). This section presents some of the current business/IT alignment approaches, as this area of research is very wide and complex. In particular, this area encompasses almost the entire work of organisations from the business point of view (e.g., budget, plans, risks, processes, structure, strategies, etc.) and the related ones from the IT perspective (e.g., budget, infrastructure, services, tools, processes, etc.) because that the two of them are strongly coupled at different levels (Aversano et al, 2012) (Chan and Reich, 2007). BIA is required in both individual standalone firms and in multinational firms (Silvius, 2007) (Ives et al, 1993). However, the BIA in the multinational firms can be associated with higher risks let alone the complexity of interaction between business roles achieving them globally (Ives et al, 1993).

Due to the size of this research area in the literature, Aversano et al have identified a number of terms used in the literature as synonyms for the Business and IT Alignment (BIA). For example, 'bridging the gap', 'harmony', 'linkage', 'fits with' and 'integration' (Aversano et al, 2012). However, following a particular BIA definition is very hard task to achieve because of the reasons outlined below:

1) The BIA encompasses almost all the relevant work in a business organisation that influence the IT and vice versa. This kind of tight coupling (i.e., dependent relation) will be shaped and identified from the many case studies and investigations carried out in order to arrive to an accurate definition. Although the identification of BIA requires more evaluation, participants in their firms can assert the strong dependency between the business and the IT.
2) The BIA research area has emerged only recently within the last two decades and therefore the literature available resources are still recent and still to be tried and tested. In addition, the related areas to business and IT are evolving independently in order to solve problems in their own scope and then to be aligned with the related business or IT. In particular, the considerable amount of work in BIA postpone effort required in order to arrive at a comprehensive, or at least an agreed, definition.
3) Researchers concluded that BIA in practice "is not a straight forward methodological process" and "has more fuzzy context, as it is implied by the methodologies" (Silvius, 2007).

However, this does not consider the absence of definitions of BIA. In simple words, Silvius defines the BIA as "*the amount to which the IT applications, infrastructure and organization, the business strategy and processes enables and shapes, as well as the process to realize this"* (Silvius, 2007). From this definition, his vision of BIA was expressed in the terms 'amount' and 'process'. The former denotes the 'state' of the alignment, where the latter denotes the "activities to reach a certain state of alignment" (Silvius, 2007). This brief and comprehensive definition highlights the large-scale research area needed to be addressed in order to cover this scope, although it will still evolve with the need of evaluation. The need for continuous alignment of IT to business needs is the key requirement for staying competitive in the business domain and market.

Understanding how business can be facilitated by IT and how IT can suggest kinds of business supports enables BIA (Luftman et al, 1999). Designing and reusing business models such as the BPA and BPMs is one of the approaches followed in order to obtain a common understanding between the stakeholders of the business of an organisation. In addition, the GO models seem necessary for the illustration of the business strategy view in order to arrive at an early agreement of business goals by the people in business and IT. This research will review relevant research work in this scope in BIA. So this boundary, (i.e., understanding the business and IT using GO models, BPAs and BPs and their integration together), is reviewed and presented next.

With regard to the current approaches that bridge the gap between GO models and the relevant BPAs, they are almost absent in the research community, but they are practiced in the domain. In addition, researchers noticed that the GO approaches are not widely practiced although they are too many in the research literature (Horkoff and Yu, 2011). However, one of the categories within a recent proposed classification for BPAs is the goal-based, as was shown in Section 2.3 (Dijkman et al, 2011). The approaches within this category draw attention to goal-driven BPAs. Those approaches consider the business process as a set of "related activities to achieve a certain goal" (Dijkman et al, 2011).

Considerable research work addressed the harmony between GO models and BPs because the design and the implementation of the latter is widely addressed in research and practice. The PRiM method re-engineered BP models to the *i** framework in order to assist in generating information system specifications (Grau et al, 2008). Also, few approaches have transformed the *i** framework goal-oriented models to BPMs (Decreus et al, 2009). In both attempts, concepts mapping appears as a common challenge (Decreus et al, 2009). Moreover, neither works attempted to model the BPA of their business processes. In addition, a strategy-driven business process modelling approach was proposed in order to find the "best fit", first, between business goals and operations, second, and the systems functions using two spiral models, where the first is an intentional model and the second is an operational model (Nurcan et al, 2005).

Recently, few attempts emerged that employed BPA model in their alignment of BPAs with BPs. For example, the BPAOntoSOA framework (i.e., the foundation of this research work suggested sector) has functionally aligned a Riva-based BPA and associated BPMN process models in order to derive candidate software services to be enacted in a service-oriented environment (Yousef, 2010). This work has complemented the work of the bridging the gap between business processes and systems initiative (Odeh and Kamm, 2003). In both attempts (Yousef, 2010) (Odeh and Kamm, 2003), a systematic mapping of concepts is performed to tackle the above challenge using conceptual modelling in the BPAOntoSOA framework, and rules in (Odeh and Kamm, 2003). The Riva method has aligned a BPA with its associated BPMs (Ould, 2006) but without an automation support. Recently, Solaimani and Bouwman proposed a generic framework that aligns business models (BPA is an example of a business model) (Solaimani and Bouwman, 2012). However, these approaches still lack the integration of GO models that should assist in directing the redesigning with respect to changes in business goals.

2.6.1 The BPAOntoSOA Framework

The BPAOntoSOA framework is a semantic framework that derives candidate software services and capabilities from a Riva BPA and its associated BPMs for an organisation. The framework is two-layered as shown in Figure 2.29. The first layer is the BPAOnt Ontology Instantiation Layer that is established in order to instantiate the semantic representation of a Riva BPA and its associated BPMs using OWL-DL, namely the BPAOnt. The second layer is the software service identification layer that uses the resultant BPAOnt from the first layer in order to identify the candidate software services and their associated capabilities. The derived candidate services are illustrated as clusters so-called Riva Process Architecture (RPA), where each cluster is comprised of members. The output from the BPAOntoSOA framework is the RPA clusters that conform to the SOA principles. In addition, the BPAOnt-driven candidate services associate their behavioural capabilities.

The BPAOntoSOA framework had a few limitations and its founder proposes further directions at the end of her PhD thesis in order to address them (Yousef, 2010). First, the BPAOntoSOA framework assumed the pre-existence of EBEs for a Riva-based BPA without ensuring that all required EBEs for the business have been identified. Therefore, Yousef suggested enhancing the BPAOntoSOA framework through involving domain ontologies (Yousef, 2010). Second, the QoS requirements of the resulting candidate software services are not identified along with the derived capabilities. The QoS requirements are necessary, as they constrain the software service according to what have been mentioned in Section 2.5. Third, she suggested extending the BPAOntoSOA framework by discovering services from the service registry (UDDI) to be achieved by an additional component or layer into the framework. Finally, the absence of rationales (i.e., business goals) beyond the driven SOA services and associated capabilities has noted. The second and the final limitations are the drivers of this research.

Next, the two layers and associated components that shape the original BPAOntoSOA framework are explained in detail. This part of the literature occupies a considerable amount of space, as it is considered the foundation of this research.

<u>The BPAOnt Ontology Instantiation Layer (Top Layer)</u>

The importance of this layer was derived from the need to have one repository that formally represented and reserved all "business knowledge" required for describing a BPA and its associated BPMs for an organisation. A BPAOnt ontology formally described "the concepts and relationships between them to provide common semantics to communicate between stakeholders" (Yousef, 2010). This layer comprises three main components, illustrated in grey colour in Figure 2.29, they are:

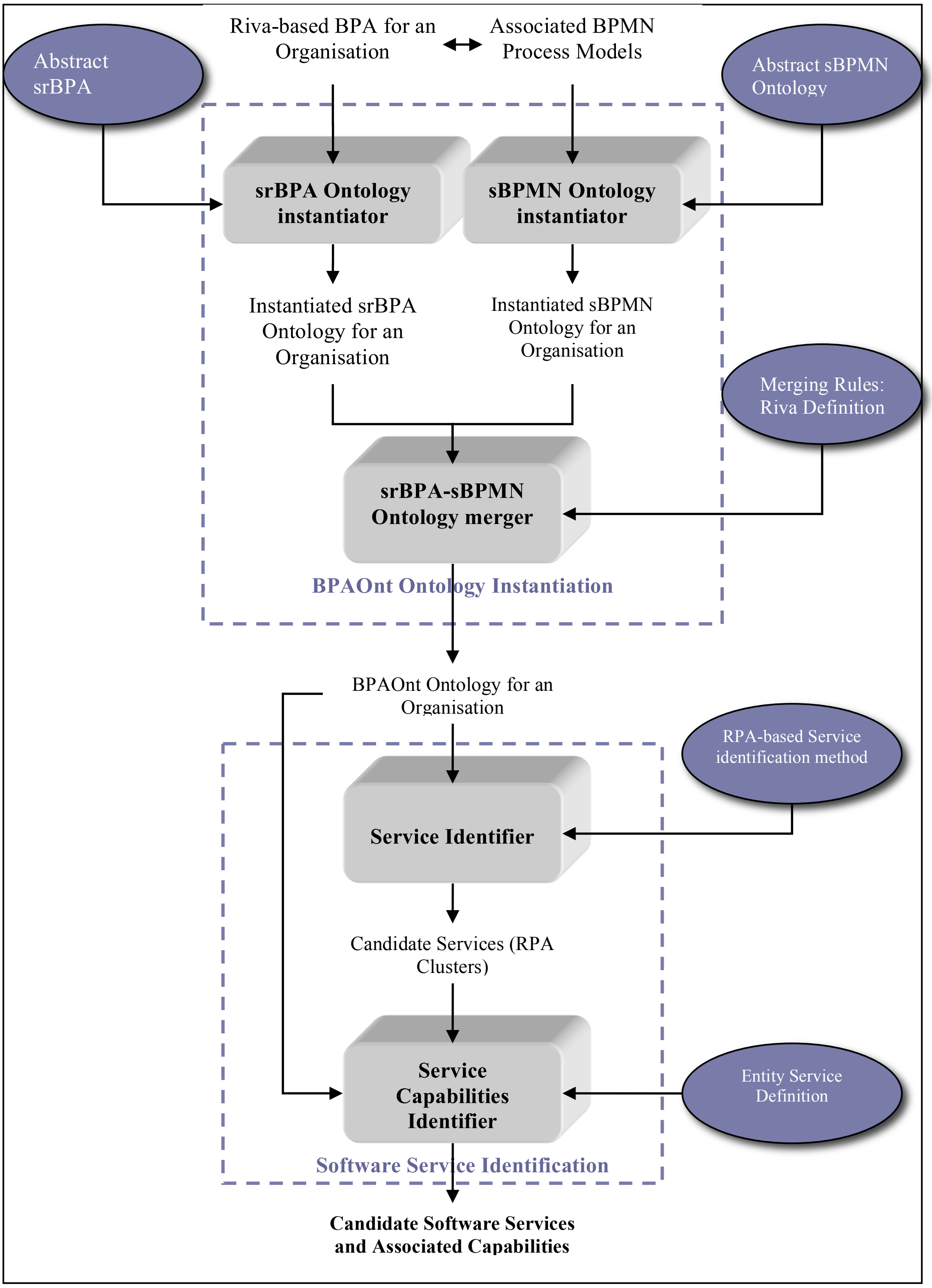


Figure 2.29: The Original BPAOntoSOA Framework [Source: (Yousef, 2010), Used with the author's permission].

1- The semantic Riva BPA (srBPA) ontology instantiator component: its main function is conceptualising a Riva BPA for a particular organisation using the developed srBPA ontology that implements (i.e., ontologises) the concepts of the Riva BPA, the relations between them and the rules that automate the Riva method steps.

2- The semantic BPMN (sBPMN) ontology instantiator component: the function of this component is to semantically represent the associated BPMs using the sBPMN ontology borrowed from the super project (SUPER, 2008 a).

3- The srBPA-sBPMN ontology merger component: the purpose of this component is to merge the two instantiated ontologies resulted from the above two components in order to derive the BPAOnt ontology for an organisation. The merging is carried out through defining two OWL-DL restrictions for the classes in srBPA ontology and mapped (i.e., linking concepts) to the sBPMN ontology. The two OWL-DL restrictions represent the Riva method rule, which states that for each CP and/or CMP in the 2nd cut architecture there is a corresponding BP presented in the BPMN approach. The result from this component is the output of the entire layer, namely the BPAOnt ontology instantiation of an organisation.

The Software Service Identification Layer (Bottom Layer)

This layer uses the instantiation of the BPAOnt (i.e., the result from the top layer) for a given organisation in order to semantically identify the candidate software services and capabilities. The layer comprises of two main components, which are illustrated in grey colour in the bottom layer in Figure 2.29, are:

1- The service identifier component: this component involves the application of the novel Service Identification (SI) approach proposed based on the Riva Process Architecture (RPA). The approach identified RPA clusters that are either associated with a standalone CP (first type cluster) or related CPs and/or CMPs (second type cluster) as shown in Figure 2.30. Two algorithms were proposed in order to implement the novel approach in (Yousef, 2010). In Figure 2.30, member of standalone CP cluster is a CP, where members of the second type cluster are CPs and/or CMPs that are connected to each other. The connected cluster relations are determined using the SWRL rules in Figure 2.31. Those members are the BPs in the Riva BPA, but they are addressing the SOA principles based on a critical understanding and analysis carried out by the founder. In particular, Yousef has mapped the characteristics of a RPA cluster to the service principles as shown in

Table 2.2. In the second type of a RPA cluster (i.e., related CPs and/or CMPs), members address the principle of loose coupling by remaining on the normal relationships and dismissing the conditional relationships between BPs in a RPA cluster as shown in Figure 2.32. The novel approach is described as an easy to understand and apply.

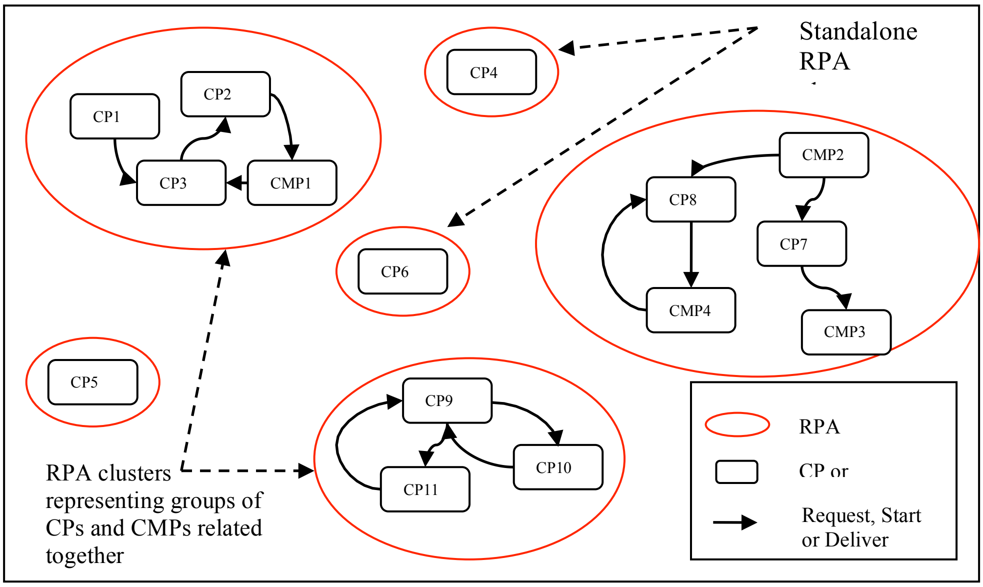


Figure 2.30: The Original RPA Cluster [Source: (Yousef, 2010), Used with the author's permission].

- *PA2Element(?e) ^ CP(?e)^ hasDeliverRelation(?e, ?d) belongsTo2ndCutDiagram(?d, PA_2nd_cut_Diagram) -> hasInOrOutRelations(?e, ?d)*
- *PA2Element(?e) ^ CP(?e)^ hasStartRelation(?e, ?s) belongsTo2ndCutDiagram(?s, PA_2nd_cut_Diagram)-> hasInOrOutRelations(?e, ?s)*
- *PA2Element(?e) ^ CP(?e)^ hasRequestRelation(?e, ?r) belongsTo2ndCutDiagram(?r, PA_2nd_cut_Diagram)-> hasInOrOutRelations(?e, ?r)*
- *PA2Element(?e) ^ CP(?e)^ Deliver(?d1) ^ belongsTo2ndCutDiagram(?d1, PA_2nd_cut_Diagram) ^ hasCPDestination(?d1, ?e) -> hasInOrOutRelations(?e,?d1)*
- *PA2Element(?e) ^ CP(?e)^ Start(?s1) ^ belongsTo2ndCutDiagram(?s1, PA_2nd_cut_Diagram)^ hasCPDestination(?s1, ?e) -> hasInOrOutRelations(?e,?s1)*
- *PA2Element(?e) ^ CP(?e)^ Request(?r1) ^ belongsTo2ndCutDiagram(?r1, PA_2nd_cut_Diagram) ^ hasCPDestination(?r1, ?e) -> hasInOrOutRelations(?e,?r1)*
- *PA2Element(?e2) ^ CMP(?e2)^ hasStartRelation(?e2, ?s2) ^ belongsTo2ndCutDiagram(?s2, PA_2nd_cut_Diagram)-> hasInOrOutRelations(?e2, ?s2)*
- *PA2Element(?e2) ^ CMP(?e2)^ Request(?r2) ^ belongsTo2ndCutDiagram(?r2, PA_2nd_cut_Diagram) ^ hasCPDestination(?r2, ?e2) -> hasInOrOutRelations(?e2,?r2)*

Figure 2.31: The SWRL Rules that fulfil the hasInOrOutRelations [Source: (Yousef, 2010), Used with the author's permission].

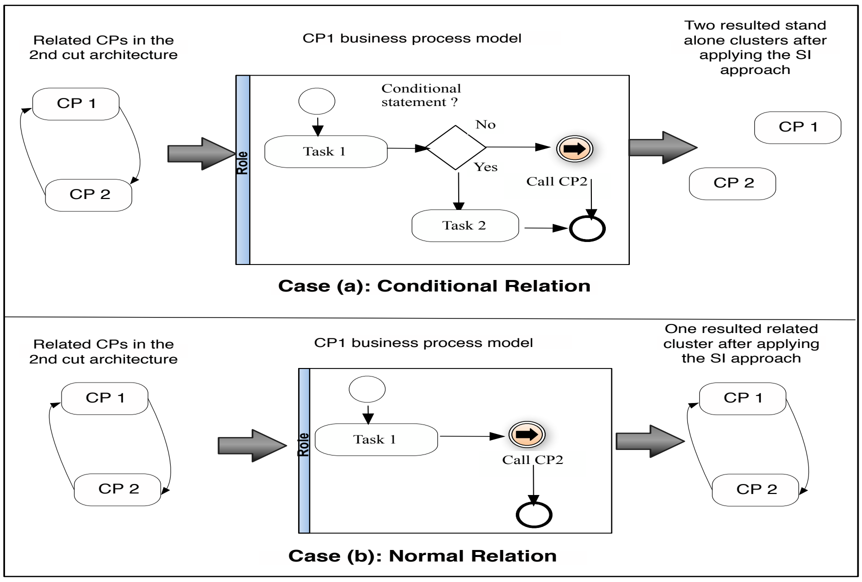


Figure 2.32: An Example of Conditional and Normal Relations for Riva BPA-driven Clusters in the Work of the BPAOntoSOA Framework

2- The service capability identifier component: this component uses the sBPMN ontology in the BPAOnt in order to identify the functional boundary through extracting the designed capabilities (e.g., send task, receive task and user task) of each member and considered them for the cluster's capabilities. Yousef implemented this thought using the SWRL rules (Yousef, 2010) as shown in Figure 2.33.

RPA_Cluster(?C) ^ CP(?cp)^ hasMembers (?C, ?cp) ^ hasCorrespondingProcess (?cp, ?p) ^ sendTask (?st) ^ hasGraphicalElemetsProcess (?p, ?st) ^ -> hasCapability(?C,?st)

RPA_Cluster(?C) ^ CP(?cp)^ hasMembers (?C, ?cp) ^ hasCorrespondingProcess (?cp, ?p) ^ receiveTask (?rt) ^ hasGraphicalElemetsProcess (?p, ?rt) ^ -> hasCapability(?C,?rt)

RPA_Cluster(?C) ^ CP(?cp)^ hasMembers (?C, ?cp) ^ hasCorrespondingProcess (?cp, ?p) ^ UserTask (?ut) ^ hasGraphicalElemetsProcess (?p, ?ut) ^ -> hasCapability(?C,?ut)

RPA_Cluster(?C) ^ CMP(?cmp)^ hasMembers (?C, ?cmp) ^ hasCorrespondingProcess (?cmp, ?p) ^ sendTask (?st) ^ hasGraphicalElemetsProcess (?p, ?st) ^ -> hasCapability(?C,?st)

RPA_Cluster(?C) ^ CMP(?cmp)^ hasMembers (?C, ?cmp) ^ hasCorrespondingProcess (?cmp, ?p) ^ receiveTask (?rt) ^ hasGraphicalElemetsProcess (?p, ?rt) ^ -> hasCapability(?C,?rt)

RPA_Cluster(?C) ^ CMP(?cmp)^ hasMembers (?C, ?cmp) ^ hasCorrespondingProcess (?cmp, ?p) ^ UserTask (?ut) ^ hasGraphicalElemetsProcess (?p, ?ut) ^ -> hasCapability(?C,?ut)

Figure 2.33: The SWRL Rules Used to Set the Property Value *hasCapabilities* with the *User*, *Send* and R*eceive* Tasks for each *RPA_Cluster* Class Instance [Source: (Yousef, 2010), Used with the author's permission].

Table 2.2: Mapping the Characteristics of the RPA Clusters in the BPAOntoSOA Framework to SOA Principles [Source: (Yousef, 2010), Used with the author's permission].

Characteristics of RPA Clusters, According to Riva Definition	Service Principles and/or Definitions	Mapped RPA cluster Characteristics to Service Definitions and/or Principles
Each CP in an RPA cluster handles an instance of a unit of work, and Each CMP in an RPA cluster manages the flow of instances of a unit of work, where units of work are initially EBEs with lifetimes handled by members of an organisation.	Entity Service definition	The functional boundary of each RPA cluster is based upon one or more business entities.
The first type of RPA clusters are stand-alone CPs, where they do not have require, start or deliver relations with other CPs or CMPs.	Principle of Loose Coupling	Stand alone CPs of RPA clusters have low dependability on other clusters.
The second type of RPA clusters is a set of CPs and CMPs that are related together through request, start and/or deliver relations.	Principle of Loose Coupling	RPA clusters that group CPs and CMPs have low dependability on other clusters.
Each CP and CMP corresponds to a process which is comprised of a set of functionalities.	Principle of Abstracting Underlying Logic	RPA clusters act as black boxes, where they abstract the underlying functionalities that are considered service capabilities.
RPA clusters are concerned with one or more related entities, where granularity level is finer than a BPA or a BPM and is coarser than tasks, and is also coarser or equal to a CP in granularity.	Principle of Reusability/ Principle of Composability	RPA clusters are highly reusable and are composable. The granularity level is not too coarse-grained nor too fine-grained.
CPs and CMPs are related through require, start and deliver relations (i.e. relations between CPs and CMPs are request/response relation, not conversational). The conversational relations between roles are included within each CP or CMP.	Principle of Statelessness	RPA clusters minimise the amount of state information they manage.

The work of the BPAOntoSOA framework generated the BPAOnt as an independent product that can be employed in order to instantiate the conceptualisation of an enterprise. In addition, the BPAOnt is a rich source of knowledge required to initiate the SI process. The BPAOntoSOA framework automated the derivation of candidate software services and their associated capabilities to some extent. In addition, it proposed another way of generating the

Riva BPA using the BPMs rather than using the results from a focus group. Finally, a forward and backward traceability network was resulted that trace the EBEs to UoWs to the identified candidate software service and associated capabilities.

2.7 Discussion and Conclusion

This chapter presented related research that positions this PhD work. Mainly, this research is positioned in the BIA area that is aimed at adjusting the software systems to business needs for an organisation. This area is highly required because of the quickly needed finding systems and solutions that address a business. Complex business problems need complex solutions. Thus, systems are developed in order to let the business organisation survive and to stay competitive in the domain. The software services are developed after understanding the business world. Part of this understanding requires attention to three essential areas and their intersections in the business world (i.e., why business, what business and how business is done). The three areas, their intersections and their related research work are shown in Figure 2.34.

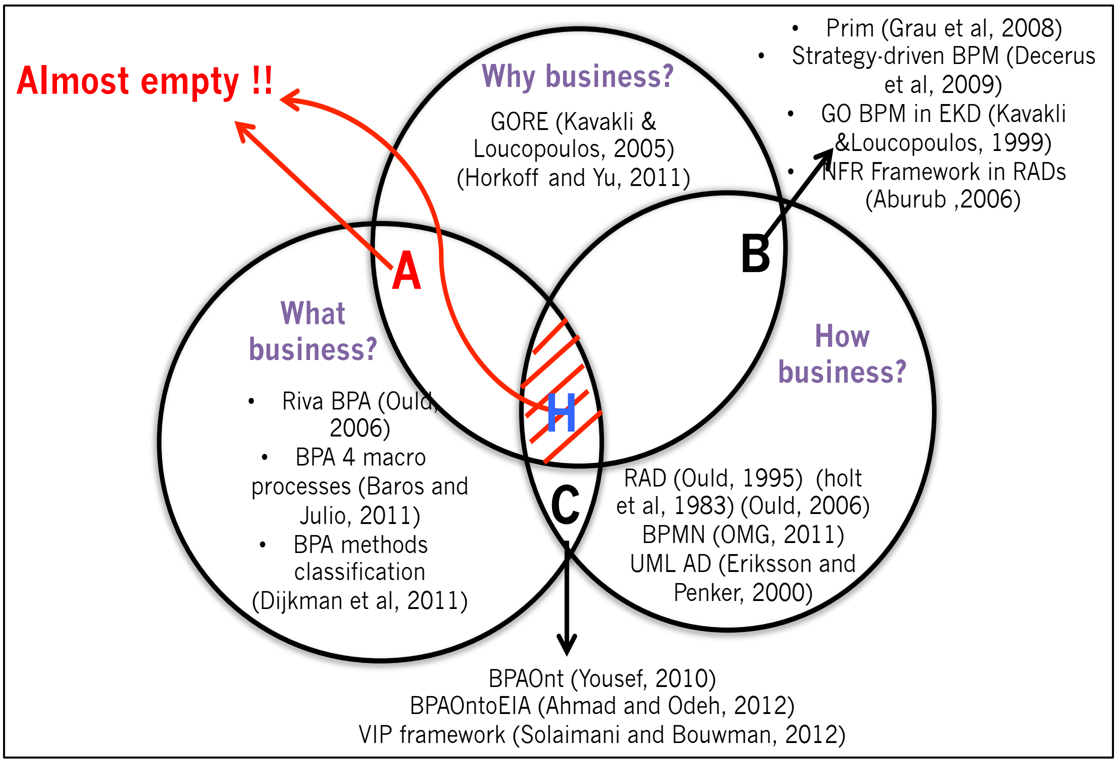


Figure 2.34: Representing the Business World Using Venn Diagram.

In Figure 2.34 and from the literature in Section 2.6, it is apparent that areas A (i.e., bridging the gap between BPAs and GO models) and H (i.e., the intersection between the three essential business areas) are almost absent. In fact, this PhD work is positioned in addressing these two areas in the business world in order to specify the software services in the system world. The absence of areas A and H in Figure 2.34 motivates conducting this research in order to develop a bridge that connects between the three essential business areas. This connection is anticipated to enrich the understanding and description of business world that can facilitate the development of related systems.

The work of this PhD is different from the related work reviewed in the literature. This is because it aims at semi automating a bridge between the three essential business areas in BIA using simple approaches for a stand-alone organisation, which is almost absent. In fact, this PhD research is interested in identifying candidate software services from the three essential business areas in Figure 2.34. The current BIA approaches in the literature are considered too complex, not structured and some are not evaluated. Some research work employed an automatic SI approach in order to bridge the gap between business world and systems (Yousef, 2010). However, it is neither goal-based nor quality-linked. In addition, the BIA approaches involve very wide research areas so they cannot all be addressed in this research project. The work of this PhD is essentially based on reusing as-is models in order to derive the to-be ones using the notion of alignment. The notion of alignment is desired in BIA because it allows an organisation stay competitive with a lower effort as possible.

The decision of basing this PhD work on the BPAOntoSOA framework has been taken after analysing the three essential areas and their intersections in the business world shown in Figure 2.34. It was apparent that the research work within each individual area is more than the ones in the intersections (i.e., areas A, B and C) in Figure 2.34. Whereas the heart of the three areas is almost empty (i.e., area H). The more the intersections between the areas the less research work found in the literature, the richer models designed and the harder to address. Therefore, it is necessary to employ and found simple approaches. The intersection between the areas is desired and important because it involves showing the linkage between them that assists in enriching the business models of an organisation that aims at identifying business-driven software services. The BPAOntoSOA framework is positioned in BIA area and addressed a good part of the aim above in bridging the gap between BPAs (i.e., what business area) and BPMs (i.e., how business) and has fully automated the identification of the software services using simple approaches. With regard to Figure 2.34, the BPAOntoSOA framework is positioned in the individual areas of what business, how business and their intersection (i.e., area C) in order to derive software services. Since the BPAOntoSOA framework is not goal-based, then by default the area of why business and its intersections with the rest areas are absent or not addressed. It is necessary to highlight that the BPAOntoSOA framework is not the only work in area C in Figure 2.34. However, the rest are not considered in this research as they are too complex, not automated and some are not evaluated. The BPAOntoSOA framework is domain independent and very recent work in the BIA that employs a simple fully automated SI approach for deriving services from essential business models such as BPA and its associated BPMs. Hence, the BPAOntoSOA framework is nominated as the most appropriate foundation for this PhD work.

In this section, the original BPAOntoSOA framework is presented, where some limitations are pointed out, as it forms the foundation of this research. In addition, The BPAOntoSOA framework is a recent BIA approach, which attempts to employ simple methods with promising outputs. The critical analysis that follows is carried out with regard to the main categories presented in the road map in Figure 2.1.

The BPAOntoSOA framework resulted in a semantic identification of candidate software services and associated capabilities. The identification process, which is based on the BPAOnt, was fully automated and resulted in software services that conform to the SOA principles (Erl, 2007). But, the BPAOntoSOA framework does not involve a consideration of goals and quality requirements. This resulted candidate services are not goal-based and lack QoS attributes. Hence, this shortcoming may result a rejection from the user side to those services that may not fully address user's goals and desired quality requirements.

Rana's SI approach is based on a Riva BPA, rather than BPMs in (Yousef, 2010), stemming from the need to identify the services from high abstract models (e.g., BPA) taking into the account the need to conform to the SOA principles (Erl, 2007). However, business goals are considered even higher than a BPA. Recalling the definition of a BSV in Section 2.1, business goals are captured in a BSV that is designed earlier than any another model. In addition, business goals and the GO approaches are anticipated to stimulate the derivation of business models such as a Riva BPA and thus, the BPAOntoSOA framework.

From the review of the original BPAOntoSOA framework, it is apparent that the SI function does not consider business goals and/or QoS, and this is because the Riva method does not consider them in the BPA model. Hence, this signals the need for refining the Riva method through bridging the gap between its BPAs and GO approaches that are designed to address both goals (Kavakli and Mylopoulos, 2005) and quality requirements (Chung et al, 2000).

The GO approaches are reviewed and selected in order to employ them in this thesis for some reasons. First, recalling the definition of a BSV in Section 2.1, the GO approaches are reviewed in order to assist in designing a BSV that is currently absent in the BPAOntoSOA framework. Second, GO approaches were taken because they "encourage the modeling of goals in order to understand or describe problems associated with business structures and processes and their supporting systems" (Kavakli and Mylopoulos, 2005). Third, GO approaches provide a rich representation regarding business goals. For example, they support the representation of hard goals (i.e., functional) and soft goals (i.e., non-functional). Also, they support the representation of strategic and tactical goals. Strategic goals refer to the high level goals that are abstract and decomposable into lower goals in order to achieve them. These lower goals are called tactical goals. However, the notion of the GO approaches is not easy to apply and their integration into the BPAOnt instantiation layer may complicate the entire structure of the BPAOntoSOA framework. Therefore, it is suggested to follow the separation of concerns principle (Kotonya and Sommerville, 1998) in order to design a separate layer for the business goals and the quality requirements.

With regard to the semantic notion of the original framework, it is intended to conceptualise the approaches employed within the new layer in order to maintain the consistency of the semantic integration between the three layers.

In this research, it is recommended that to use a Riva-based BPA due to its simplicity and it's structured easy-to-apply steps. In addition, this research has the advantage of reusing the Cancer Care Registration (CCR) Riva-based BPA (Yousef, 2010) to serve the research aims. Although the Riva method is simple, systematic, easy to understand and easy to employ in business process management field and in the software engineering field, still it fails in addressing the following points as follow:

1- It is a one-view model that represents just a required function-oriented view of the organisation's business. In particular, it does not feature its desired quality attributes and its fulfilment of goals.
2- It is not derived from rationales. Instead it is derived only from entities (i.e., EBEs).
3- In Riva, the analyst establishes the BPA model through brainstorming its fundamental blocks that are the EBEs, which is not an easy step in the Riva method. Therefore, the brainstorming activity may miss some required elements, and/or identify some elements that are not required. This is because of the nature of the brainstorming techniques that depends on a set of people to generate an idea that in the best case the questions are identified to make up the problem. This limitation will be addressed in

this research by replacing brainstorming with a more structured technique to generate those elements.

The original BPAOntoSOA framework required the existence of the Riva BPA and associated BPMNs. Therefore, the new framework is anticipated to reuse them using the notion of alignment (i.e., stemming from the literature of BIA) rather than designing them from scratch. The new layer is anticipated to strongly influence this alignment.

Since this research is derived from the original BPAOntoSOA framework, it inherits the common literature and reuses its background in (Yousef, 2010) in order to address a deeper understanding regarding the original framework's roots. However, particularly during the review of the current SI approaches, the novel SI approach proposed in the BPAOntoSOA framework seemed the most simple approach and the only one based on the Riva BPA and associated BPMNs. Accordingly, in this research, it is suggested to employ this original novel SI approach (i.e., the one that derives RPA clusters from a Riva-based BPA) without putting considerable effort in reviewing and documenting the SI approaches in this literature chapter.

This chapter concludes with the aim to establish a new layer into the BPAOntoSOA framework that is aimed at conceptualising a BSV for an organisation. In addition, it is required to bridge the gap between (i.e., to align) the newly established layer and the original BPAOnt ontology instantiation layer. The integration of the new layer is anticipated to influence the derivation of the BPAOnt instantiation layer and particularly the Riva BPA.

For the GO approaches that are suggested to be employed in the new layer, the *i** framework is the most appropriate candidate, as it is applied when there is a need to understand the business of an organisation. In addition, it early pays attention to actors as active entities, where many GO approaches do not. It is important to early identify the actors in this stage instead of postponing it to the BPM. Furthermore, the *i** framework is rich in concepts that are mappable to concepts in BPAs and BPMs. The NFR framework is the most well structured quality-oriented approach for addressing quality requirements that are not defined in a clear-cut manner. Accordingly, it is anticipated to conceptualise the *i** framework and the NFR framework. In addition, investigating the relation between the two will be necessary. The new layer is anticipated to enhance the functionality of the original BPAOntoSOA framework and supporting agility in order to address organisation's goals and its new business-driven requirements.

Finally, the work of this chapter has addressed the answering of research questions shown in Figure 2.35. In addition, the work has been part of the research process phases indicated in Figure 2.36.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1		✓							🚩
RQ2		✓							🚩
RQ3									⌛
RQ4									⌛

Figure 2.35: Answering Research Questions for Chapter 2.

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1		✓							**Active**
RP2									⌛
RP3									⌛
RP4(ai)									⌛
RP4(aii)									⌛
RP4(bi)									⌛
RP4(bii)									⌛
RP4(biii)									⌛
RP5									⌛
RP6									⌛

Figure 2.36: Active Research Process Phases for the Work of Chapter 2.

Chapter Three: Bridging the Gap between Business Strategy View, BPA and Service-oriented Software: A Further Research Evolution of the BPAOntoSOA (GQ-BPAOntoSOA)

3.1 Introduction

This chapter aims to introduce the research design framework aimed at bridging the gap between business goals, quality requirements and the BPAOntoSOA framework; resulting in the GQ-BPAOntoSOA framework. The BPAOntoSOA framework is the foundation of this research work. However, the absence of a business strategy view in this framework, that is identified by OMG as discussed in Chapter 2 in (OMG, 2013), has limited the functionality of the BPAOntoSOA framework. In particular, the Riva BPA-driven candidate software services do not stem from business goals. In addition, the derived software services and their capabilities are not constrained by the desired QoS requirements. These two shortcomings motivated undertaking this research in order to resolve the absence of goals and quality requirements that to set up the business strategy view for an organisation.

The chapter is structured as follows: Section 3.2 presents the research methodology employed in this research. In Section 3.3, presents research strategies that fit with the nature of this research using the work of Section 3.2. The research process that shows the research stages from the beginning until the end will be presented in Section 3.4. The new framework characteristics and requirements are presented that realise the framework in Section 3.5. In Section 3.6, the GQ-BPAOntoSOA architectural framework is presented and explained from its architectural point of view. Besides, we explicitly refer to the components borrowed and used from the original framework. Since the GQ-BPAOntoSOA framework stems from the BPAOntoSOA framework, then a comparison must be conducted between the two frameworks with the respect to a set of comparison aspects as presented in Section 3.7. Finally, Section 3.8 summarises the work of the chapter.

3.2 The Research Methodology

Behavioural science and design science are two well-known research paradigms that are employed in information systems research. Behavioural science is hypothesis-driven paradigm that aims at proving/disapproving the hypothesis at the end of the research. The design science is a problem-solving paradigm that aims at constructing, implementing and evaluating an artefact (Hevner and Chatterjee, 2010). "Design Science Research (DSR) is a research paradigm is which a designer answers questions relevant to human problems via the creation of innovative artefacts, thereby contributing new knowledge to the body of scientific evidence. The design artefacts are both useful and fundamental in understanding that problem" (Henver and Chatterjee, 2010). Recalling that this PhD begun by identifying the research problem statement, aims and contributions in Chapter 1. It is apparent that this research aims at solving a problem in the first place through developing an artefact that is goal-based and quality-linked extension of the BPAOntoSOA framework. Therefore, the nature of this research fits well with the DSR.

A process had been introduced in (Peffers et al, 2006) in order to guide the application of the DSR method. According to (Peffers et al, 2006), three main objectives should be addressed in DSR that are: 1-"be consistent with design sciences processes in other disciplines", 2- "provide a nominal process for conducting the research", and 3- "provide a mental model for what DS research output looks like". The DSR method process consists of six steps as shown in Figure 3.1. In this figure, it is apparent how the steps are overlapped. In section 3.3, the research process phases of this PhD are introduced and related to DSR steps. Next, each step is described and then related to this PhD research chapters.

1- *Problem identification and motivation*: Since the DSR can be described as a problem-driven method, then the DSR method begins with identifying a research problem. It is possible elaborating this problem into sub problems in order to manage the solving. In addition, each problem can derive a corresponding motivation that is the value from the anticipated solution. Resources that feed this step are the state of the problem, knowing experiences, related artefact and the importance of the solution.

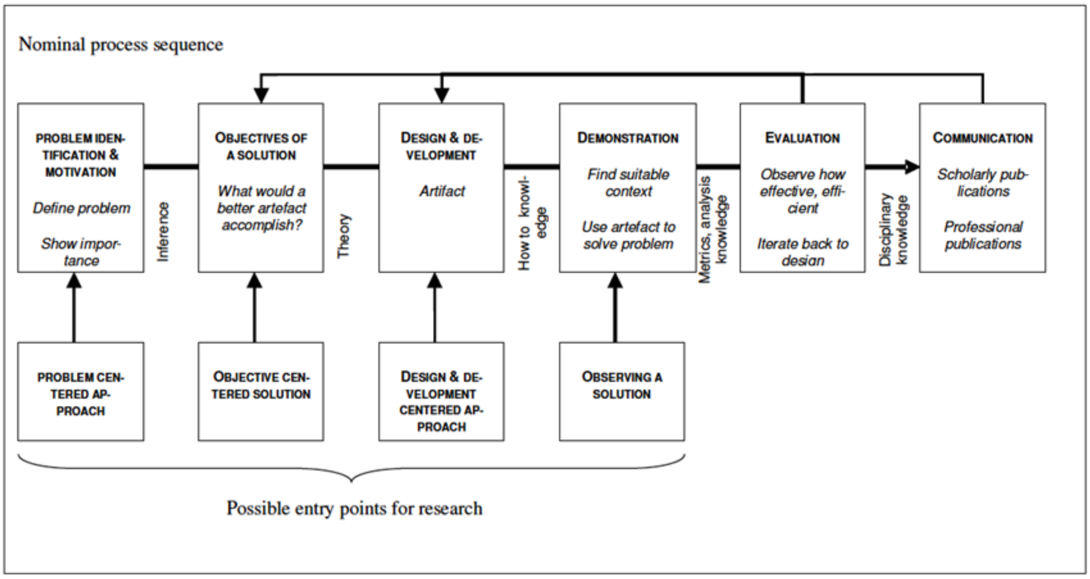


Figure 3.1: The Design Science Research Method Process [Source: (Peffers et al, 2006), Used with the author's permission].

In this PhD, Chapters 1 and 2 address this step. Chapter 1 begins with the research context, then research problem(s) are identified with their corresponding motivations. In addition, the research questions, aims, and objectives are introduced. Chapter 1 ends with the contributions of this research. Chapter 2 covers the knowledge of the state of the problem that involves business goals and its approaches, quality requirements, BPAs, BPMs and BIA approaches.

2- *Objectives of a solution*: In this step, the problem definition in step 1 is used in order to derive the objective of a solution. An objective of a solution can be quantitative, that determine whether the proposed solution is better than others ones, or qualitative, that determine of the proposed solution (i.e., artefact) is attempting to solve a problem that has not addressed before. Knowledge of the state of the problem and the efficiency of the current solutions are required resources for this step.
In this PhD, Chapter 1 involved presenting a qualitative objective of a solution, as the BPAOntoSOA framework is recent and no attempts have been conducted regarding investigating the integration of goals and quality requirements into the original framework. Chapter 2 presents the state of the problem.

3- *Design and development*: This step aims at constructing the solution (i.e., artefact) and then developing it. This involves identifying the artefact functionalities, architecture and then constructing the actual artefact. Knowledge of theory is a required resource while moving from the previous step.

In this PhD, this step is covered within the work of Chapters 3,4,5 and 6. The entire design of the GQ-BPAOntoSOA framework is introduced in Chapter 3, the first layer is addressed in Chapter 4, the second layer is addressed in Chapter 5, where the third layer is addressed in Chapter 6.

4- *Demonstration*: This step involves using the developed artefact in an experience, case study or simulation. The required resource for this step is knowledge with regard to the way how this artefact should be used in order to address resolving the research problem. In this PhD, this step is covered in Chapter 7 and used the artefact (i.e., the GQ-BPAOntoSOA framework) in a case study from the health care domain.

5- *Evaluation*: This involves assessing the effectiveness of the artefact regarding resolving the problem. The objectives of the solution are compared to the actual results generated from using this artefact. Thus, knowledge analysis techniques and/or metric are required. If the artefact needs effectiveness improvement, then researchers can decide returning to step 3 (i.e., design and development). Alternatively, a researcher can continue to the next step (i.e., communication) and consider this as further research work.

6- *Communication*: This is the final step in DSR process. In this step, researchers highlight the importance of the problem, artefact, its novelty, its contribution and its effectiveness to the community that involve researchers and interested audience. The resulted work of a research conducted using the DSR is published. This step is addressed in Chapter 8, whereas publications are listed in Chapter 1.

3.3 The Research Strategies

This section presents research strategies that fit with the nature of this research. Knowledge of the state of the problem assisted in shaping the boundaries for the selected DSR paradigm. For example, the research is bounded by context and addresses limited qualitative concepts.

The nature of this research with its qualitative objectives of a solution rests on different research strategies that are for example case studies, grounded theory, action research and ethnography (Collis and Hussey, 2009).

A case study is defined as “a methodology that is used to explore a single phenomena in a natural setting using a variety of methods to obtain in-depth knowledge” (Collis and Hussey, 2009). Basically, it focuses “on collecting information about a specific object, event, or activity, such as a particular business unit or organisation” (Sekaran and Bougie, 2013). The beauty of case studies appears in their ability to provide “both qualitative and quantitative data for analysis and interpretation (Sekaran and Bougie, 2013). Such a good example of a rich case study is CCR case study in Jordan (Aburub, 2006).

Grouneded theory is “a systematic set of procedures to develop an inductively derived theory from the data” (Sekaran and Bougie, 2013). The researcher keeps on refining and adjusting the resulted theory with any new data until they both fit together.

In the present research, the CCR case study method is employed within the DSR research paradigm. The study is reused rather being conducted by the researcher of this thesis, as it is originally derived from (Aburub, 2006). The study was found to be useful for evaluating related research work in (Yousef, 2010) and (Ahmad and Odeh, 2012). The important characteristic of this study is that it has been used in the evaluation of the original BPAOntoSOA framework and this is anticipated to answer the research questions in Section 1.4.

3.4 The Research Process Phases

In this section, the research process conducted is presented in order to solve the identified research problem that mainly sets on the need to identify the QoS requirements that constrain the capabilities of the BPAOntoSOA-driven candidate services. This requires integrating goals and quality requirements into the BPAOntoSOA framework as was presented in Chapter 1.

The research process refers to the systematic or structured plan representation of the activities and phases conducted with respect to the time allocated in order to show the journey from the beginning (i.e., identifying research problem hypothesis) until the end (i.e., discussing hypothesis testing results). Hence, in this thesis, the research process design aims to produce a refined original framework (i.e., a refined BPAOntoSOA framework) that aligns goals and associated soft goals (i.e., quality requirements) with pre-existing Riva-based BPAs in order to address the missing QoS requirements for the pre-existing candidate services. Figure 3.2 depicts the research process scheme and highlights the systematic and interleaved research phases along with directions towards a solution for the already identified research problem. It is necessary to highlight that the research processes depicted in Figure 3.2 is an elaboration of the DSR process shown in Figure 3.1.

The research data comprise primary and secondary data (Collis and Hussey, 2009) (Eriksson and Kovalainen, 2008). In the former, interviews were conducted to investigate the feasibility of deriving goal-oriented models for the corresponding partial Riva-based BPA of the CEMS's Faculty of Administration (Green and Ould, 2004). The investigation is used twice in the research process, first time in finding the alignment relationship between the GO models and Riva BPA and second in evaluating the impact of identifying services from goal-based and quality-linked BPA. In the latter, the CCR case study already exists in nominal form (i.e., textual and visual data) (Aburub, 2006) (Yousef, 2010). Moreover, relevant research publications have been explored in order to gain further knowledge and a deeper understanding regarding the research problem. Within the research design, the CEMS's Faculty Administration example and the CCR case study as for the research evaluation. The first is employed as a pilot case study research strategy for aligning GO models with a Riva BPA and using the results in another purpose that is investigating the impact of the aligned BPA with goals on the pre-existing candidate services. The CCR case study is employed in the comprehensive evaluation of the proposed framework (i.e., the GQ-BPAOntoSOA framework) in Chapter 7.

The CCR case study appears particularly appropriate for this research for four reasons. **First**, it fits very well into the qualitative research paradigm. **Second**, the BPAOntoSOA framework was evaluated using this case study in (Yousef, 2010). This is needed in order to compare the outcome from the original BPAOntoSOA framework with the ones from the extended framework. **Third**, relevant research data have been already collected for the CCR case study in relation to goals and quality requirements. And finally, the second and the third reasons assist in utilising the research time period to evaluate the GQ-BPAOntoSOA framework. In fact, it will benefit this research through saving time for applying the BPAOntoSOA framework again in order to analyse its results against the results of the proposed framework in the research of (Yousef, 2010).

The research process runs in six main phases as depicted in Figure 3.2. The phases are: 1- the preliminary phase, 2- the early theoretical framework design phase, 3- the investigation phase, 4- the original BPAOntoSOA framework enhancement phase, 5- the conceptual framework development phase, and finally 6- the application and evaluation phase. Although the entire design looks systematic, still some phases are mutually dependent. All these phases iterate and are performed in respect to a time schedule that is a significant part of the research design (Eriksson and Kovalainen, 2008). It is important to highlight that each output from a phase form the input of the next phase. Next, each phase within the process is discussed in detail.

3.4.1 The Preliminary Phase

The preliminary phase acts as the cornerstone towards determining or shaping the research boundaries as depicted in Figure 3.3. This phase iterates and requires refining the outcomes of each element in Figure 3.3 in order to adjust the research process settings. This phase is literature-driven and involves the identification of gaps in the related discipline and consequently refining the research problem statement that is presented in Section 1.4. In addition, this phase determines the research motivations and aim that are fulfilled by the achievement of the associated objectives through answering research questions that in turn attempt to assert or reject the identified research hypothesis.

The state of the art literature has already been reviewed in Chapter 2 based on the roadmap presented in Section 2.1. The main output of this phase is obtaining a critical and a deeper understanding of the BPAOntoSOA semantic framework that will be extended in the later chapters, (i.e., Chapters 4,5 and 6) in addition to the agreed research problem statement regarding the BPAOntoSOA framework. Also, Ontology development tools such as Protégé as well as an applicable case study are both determined in the preliminary phase. The CCR process was selected as the research strategy to evaluate the GQ-BPAOntoSOA framework. Consequently, the preliminary phase resulted in Chapters 1 and 2, the research timeline estimation and the next research phase.

With regard to the DSR method, this phase addresses its first and second steps as discussed in Section 3.2.

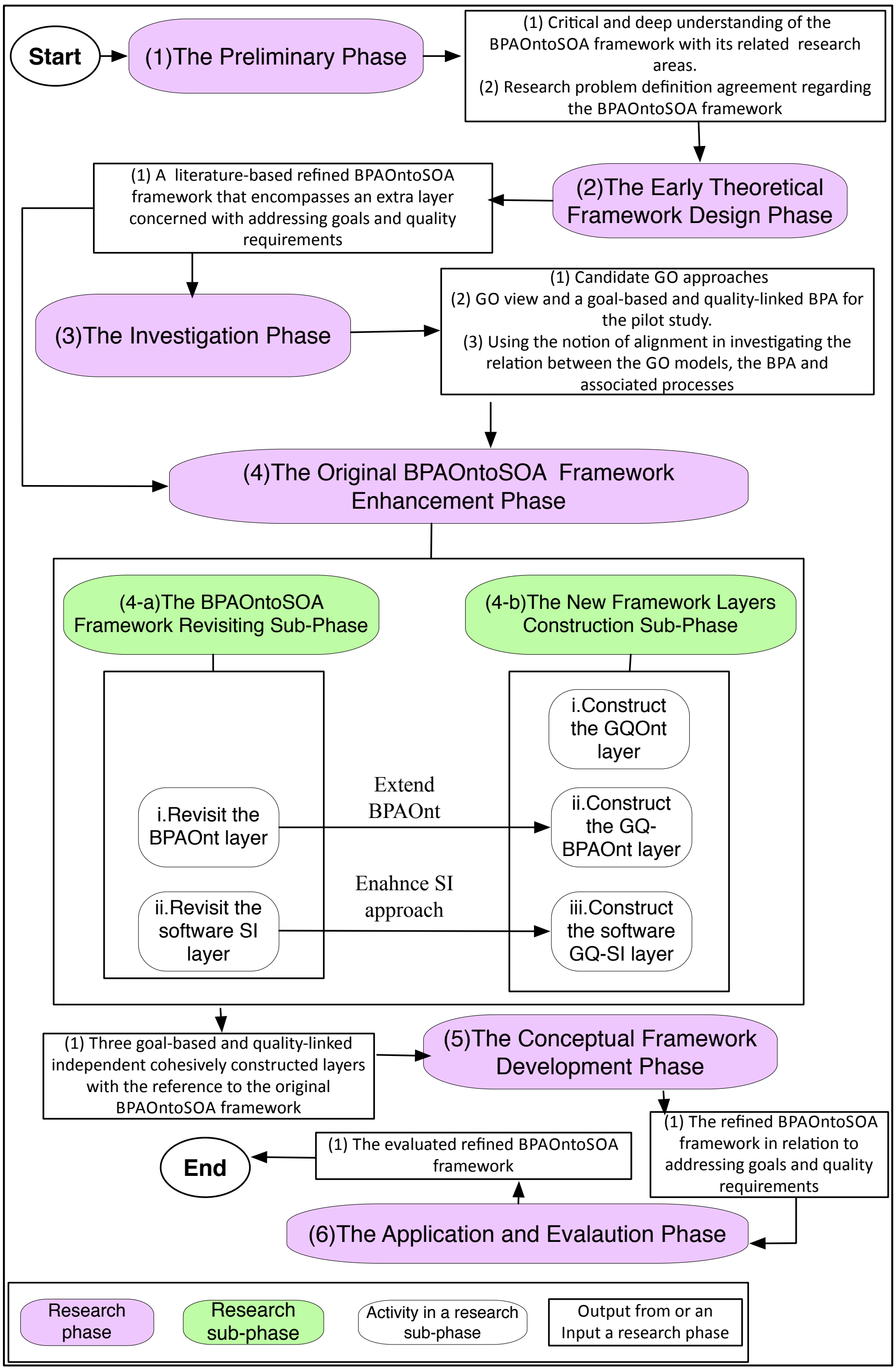


Figure 3.2: The Research Process Phases

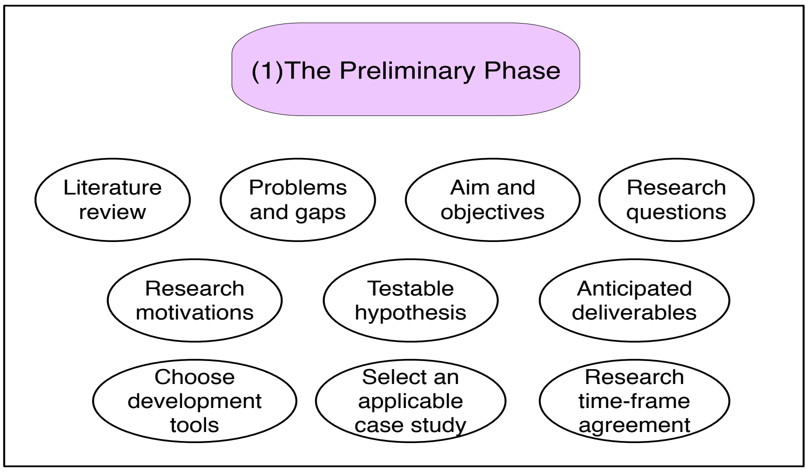


Figure 3.3: The Research Activities in the Preliminary Phase

3.4.2 The Early Theoretical Framework Design Phase

In this phase, the researcher bases the input on the preliminary phase's outcomes and the literature review carried out in order to direct the initial design decisions of the GQ-BPAOntoSOA framework. The theoretical initial design of the framework is the outcome of this phase that is clearly reusing the BPAOntoSOA framework.

It was apparent from the literature and, particularly in the BIA attempts within a dynamic environment, that the BPAOntoSOA framework is one of the few key candidate approaches that utilises a BPA model in generating a semantic representation of candidate software services (Yousef, 2010). However, the BPAOntoSOA framework has two limitations **First**, the generated services are not goal-based and this is mainly because the Riva methodology employed is not a goal-driven approach. **Second**, the BPAOntoSOA framework generates the SOA service capabilities that do not address their associated QoS requirements. This limitation manifests another weakness in the employed Riva BPA methodology, where the derived business processes or candidate services from a Riva-BPA did not address the desired quality characteristics (Ould, 2005). This is mainly because the Riva method does not consider quality requirements while modelling a BPA. Hence, this encourages new design decisions to meet service's goals and soft goals. Moreover, a goal-based semantic representation motivates the identification of associated soft goals that in turn constrain the goals achievement. Consequently, these soft goals are anticipated to lead the identification of QoS requirements for the derived candidate services to address the desired characteristics of services' behaviour at run time.

The above two major shortcomings justify the need to undertake this research and these are already presented in the research problem statement in Chapter 1 (Section 1.2). Therefore, this phase begins to engage these two rich qualitative concepts, which are goals and quality requirements, into the existing BPAOntoSOA framework (Yousef, 2010). In this phase, an initial framework that abstractly extends the as-is BPAOntoSOA framework is designed to conquer the above two shortcomings resulting the GQ-BPAOntoSOA framework. First, an additional goal-oriented layer that motivates the derivation of the associated quality requirements is introduced. By and large, the GQ-BPAOntoSOA framework is composed of the BPAOntoSOA framework in addition to the new components of goals and quality that are integrated into a Riva-based BPA and its associated business processes.

In the later stages of this research process and particularly in the fifth phase (i.e., the framework conceptual development phase), the initial design of the GQ-BPAOntoSOA framework is elaborated. Consequently, this emphasises the overlap and dependency of the research design phases. Overall, the early theoretical framework design phase is concerned with delivering an initial design of the GQ-BPAOntoSOA framework rather than a solution. The initial design of the GQ-BPAOntoSOA framework embodies the following claim: "for a given BPA model and associated BPMs, there must be a corresponding GO models that influence the derivation of the former and vice versa.". Hence, this phase addresses the third step in DSR method (i.e., design and development).

3.4.3 The Investigation Phase

The goal and quality concepts involved in the early design phase output need to be well allocated to the relevant components that constitute the new layer added to the BPAOntoSOA framework. This phase carries out an early investigation for discovering the relation between the GO models and the corresponding BPA model and associated business processes using a pilot case study. The intended discovery is required in order to initially demonstrate the claim embodied in the early theoretical framework design of the GQ-BPAOntoSOA framework. This phase is necessary in order to direct the construction, design and implementation of the layers and their integral components.

Accordingly, in this phase, three investigation activities are performed respectively to consider the claim and to elaborate the early design later. The three investigation activities are as follow: (1) search for the most viable goal-oriented approaches in relation to the Riva BPA through studying current comparisons and surveys with regard to the GO approaches, (2) collect data (i.e., the GO models as the current BPA already exists) using qualitative research methods such as interviews and (3) integrate the collected data into the corresponding portion of Riva-BPA and re/use this demonstration. The data representations are modelled using the $i*$ framework that has the ability to be internally integrated and designed within the Riva method. In addition, the $i*$ framework involves the hard goal and soft goal concepts, thereby it appears as a candidate comprehensive approach for the investigation.

The output from this phase is the GO models resulting from the pilot study in addition to using the notion of alignment in predicting the relationship between the GO models and the corresponding Riva-based BPA and associated BPMs. With regard to the DSR method, this phase addresses the third and fourth steps (i.e., design & development and demonstration).

3.4.4 The Original BPAOntoSOA Framework Enhancement Phase

According to the previous investigation activities, this phase aims to refine/improve the BPAOntoSOA framework by first reviewing its pre-existing architectural design in order to direct the new complementary detailed design decisions with regard to integrating goals and quality requirements into the reviewed framework (Yousef, 2010). And second by constructing the refined architectural design of the BPAOntoSOA framework. Mainly, the input is the candidate GO approach, its GO models resulted from the pilot study that is the CEMS Faculty of Administration and the notion of the alignment as a proposition for the integration of goals and quality requirements into the BPAOntoSOA framework. Therefore, derived from the above aim, this phase encompasses two main sub-phases illustrated in green colour in Figure 3.4. This phase addresses the third step, which is design and development, in DSR method.

The first sub-phase is for the BPAOntoSOA framework revisiting is concerned with performing independent revisits to the BPAOntoSOA framework layers (Yousef, 2010). The second sub-phase is for the new framework construction that is concerned with constructing the new framework layers independently based on the carried out revisits to the BPAOntoSOA framework. The layer constructions are mainly with respect to integrating goals and quality requirements. Each sub-phase is divided or elaborated into activities that are allocated in order to address their sub-phase. Figure 3.4 clarifies this design by blueprinting this phase, its sub-phases in green and activities in white.

It is worth mentioning that any improvement captured by this phase is a synonym for extending or adding new features, remaining and refining some or deleting unwanted parts of the BPAOntoSOA framework. Moreover, findings in these sub-phases must be demonstrated using the already collected data generated from the early investigation phase. This will assist the revisiting sub-phase activities to derive a corresponding construction activated in the second sub-phase. Next, the two sub-phases and their associated activities are explained with respect to goals and quality requirements.

3.4.4.1 The BPAOntoSOA Framework Revisiting Sub-Phase

Figure 3.4 depicts this sub-phase and it is denoted by the rectangle on the left hand side. The sub-phase involves carrying out two activities, where each is concerned with revisiting a particular layer in the current BPAOntoSOA framework. The two activities are: (i) revisit the original as-is BPAOnt Ontology Instantiation layer and (ii) revisit the original as-is software service identification layer. Each activity must be performed with respect to **goals** or **quality requirements**.

3.4.4.1.1 Revisiting the BPAOnt Ontology Instantiation Layer

This activity aims to investigate the BPAOnt ontology instantiation layer's adaptability for the desired integration of goals and quality requirements. Hence, the input is the current BPAOnt Ontology Instantiation layer extracted from the BPAOntoSOA framework. In particular, the merits and demerits of the Riva method will be considered when investigating its adaptability to explicitly and/or implicitly embody new and foreign concepts such as goals and quality requirements. In another words, in this activity a gap analysis is performed between the GO models and the Riva BPA, and the associated business processes in order to bridge the current gap. Moreover, this review and analysis activity is carried out along with using the findings obtained from the earlier investigation phase, which are the GO models using the CEMS Faculty of Administration BPA. This activity is mandatory prior to heading

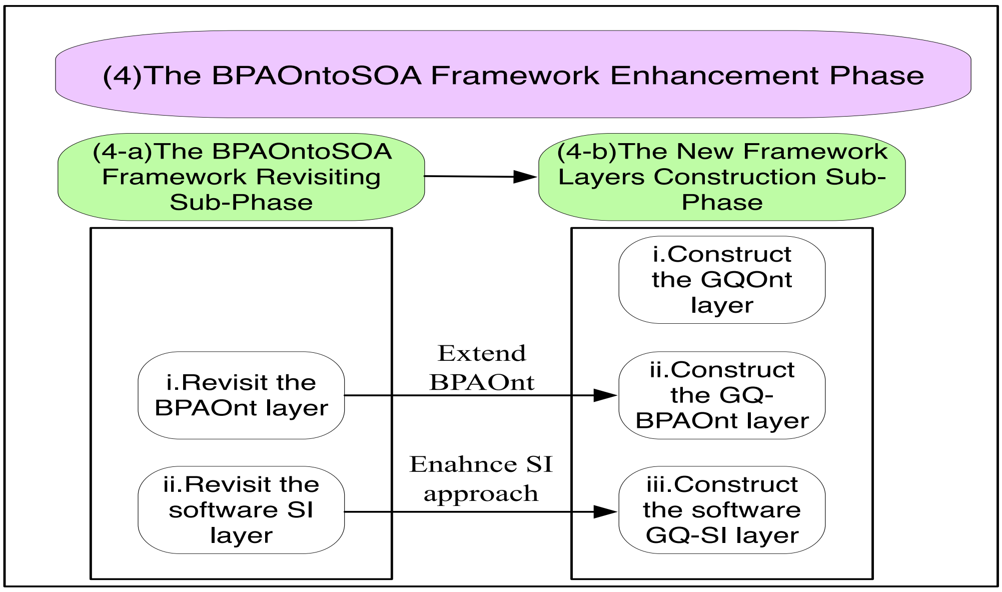


Figure 3.4: The BPAOntoSOA Framework Enhancement Phase.

towards proposing a corresponding construction of the refined BPAOnt Ontology Instantiation layer in activity (4-b-ii).

In short, this activity involves a critical review and analysis of the BPAOntoSOA framework first layer's components, structural design, approaches followed (e.g., Riva BPA method and BPMN modelling for the BPMs) and the OWL-DL ontology implementation. The output of this activity is the design guidelines or algorithms for the construction of the corresponding new/refined layer. Therefore, this output will be the entry for the second activity within the second sub-phase, namely constructing the GQ-BPAOnt Ontology Instantiation layer.

3.4.4.1.2 Revisiting the Software Service Identification Layer

This activity is rich with review and analysis work. The input to this activity is the current software service identification work of the BPAOntoSOA framework (Yousef, 2010). **First**, the aim is to review the as-is service identification layer in the BPAOntoSOA framework and in particular its related algorithms or approaches (e.g., RPA clusters) in order to investigate their adaptability to embody goals and quality requirements of the identified software services. In addition, the SOA principles used to identify candidate software services are reviewed and analysed regarding addressing service goals and quality requirements. **Second**, this activity is aimed at analysing the impact of bridging the gap identified between GO models and the BPA along with processes on the pre-existing candidate services. This is because the SI approach used in the BPAOntoSOA framework is described as a BPA-driven, thereby it is needed to involve the BPA and its associated processes in this analysis part of the activity. The output of this phase is the guidelines for constructing the goal-based software service identification layer that consider the derivation of the QoS requirements. The guidelines are the entry to the third activity within the construction sub-phase (4-b-iii).

3.4.4.2 The New Framework Layers Construction Sub-Phase

After the revisiting sub-phase is carried out, it is possible to operationalise the reviewing and the analysis work that generate the guidelines for the integration of goals and quality requirements into the original framework. Figure 3.4 represents this sub-phase and its associated activities. This sub-phase involves carrying out three activities, where each is concerned with constructing a particular layer in the new framework that addresses the integration of goals and quality requirements. The three activities are as follows: (i) construct the GQOnt Ontology Instantiation layer, (ii) construct the GQ-BPAOnt Ontology Instantiation layer, and (iii) extend the original as-is software service identification layer with regard to goals and quality requirements. Each activity must be performed with respect to **goals** or **quality requirements**. Next, the three activities are explained.

3.4.4.2.1 Constructing the Goal-based and Quality-integrated (GQ)Ont Ontology Instantiation Layer

This activity aims at constructing a new cohesive layer that to be added into the original BPAOntoSOA framework. The input is the early design of this layer generated from the execution of phase 2. In addition, the candidate GO approaches adopted in the investigation phase will assist in structuring the components of the layer. In particular, this activity aims at generating the semantic representation of the BSV for an organisation that uses the BPAOntoSOA framework in generating the software services from the BPA and the BPMs. Before providing a semantic representation, it should have been agreed on the approaches that will be involved in representing the BSV in order to agree on the design decisions for the implementation purposes. The initial structure of this layer design will be reused and refined.

During the construction, it is required to distinguish between the GO model along with its related concepts and the quality related models (i.e., soft goal-oriented models) and concepts. The output of this activity is the GQ-Ont after evaluating it using a pilot case study (e.g., extracted study that is patient reception BP from the entire CCR processes case study).

3.4.4.2.2 Constructing the Goal-based and Quality-integrated (GQ)BPAOnt Ontology Instantiation Layer

This activity reuses the outcome (i.e., guidelines) generated from the corresponding revisiting activity in the first sub-phase (4-a-i) in order to produce a solution or a mechanism to incorporate goals and quality requirements into the BPAOnt. The overall organisation, which comprises of the current related components borrowed from the BPAOntoSOA framework and the components that address or embody the notion of the goals and quality requirements, constitutes the desired construction for the goal-based and quality-linked BPAOnt Ontology Instantiation layer.

The construction is evaluated using the pilot study conducted in the early investigation phase (e.g., CEMS Faculty of Administration). This activity generates a novel approach or an algorithm that would actually bridge the gap between GO models and Riva BPA in an iterative process in order to construct a semi-formal bridge that would pave the way for the formal representation later using ontologies.

3.4.4.2.3 Constructing the Goal-based and quality-linked SI Layer

In particular, this activity attempts to propose a novel or refined approaches that consider goals of the services and their QoS requirements. In addition, the activity implements the design of the approaches into the original layer. The designed approaches and their implementation are encapsulated within the relevant components.

The output of this phase is a goal-based and quality-linked SI layer (GQ-SI) that is concerned with deriving the SOA candidate services, their capabilities and associated QoS requirements.

3.4.5 The Conceptual Framework Design and Development Phase

This phase attempts to organise the linking and the integration of the three constructed goal-based and quality-linked layers via appropriate relations in order to shape the finishing of the entire framework. Also, this phase aims at implementing the three developed layers using the OWL-DL. This phase addresses the third step, which is design and development, in DSR method.

3.4.6 The Application and the Evaluation Phase

In this phase, the final design and construction of the framework is applied using a case study from the health care domain using the CCR case study (AbuRub, 2006). This case study has been used in the evaluation of the BPAOntoSOA framework (Yousef, 2010). Therefore, it is strongly advocated to reuse of the CCR case study since it involves the main concepts used in this research such as goals, quality requirements and process modelling. Applying the resulted framework using a case study has potentiality for emerging shortcomings that may represent some gaps within the proposed framework.

Therefore, the gap-related parts must be resolved within the framework by revisiting the previous related research phases. Finally, the work of this research is evaluated using and evaluation framework that employs the concern-based approach adopted from (Kotonya and Sommerville, 1998), as shown in Chapter 7. The deliverables in this phase are the thesis documentation and the associated publications resulted during the research journey. The conclusion of the work is presented in Chapter 8. Hence, the phase addresses the fourth, fifth and the sixth steps in DSR method.

3.5 The GQ-BPAOntoSOA Framework Characteristics and Associated Requirements to fulfil it

Since the GQ-BPAOntoSOA framework stems from the BPAOntoSOA framework, then it is likely to exploit many of the original framework's characteristics. However, some characteristics are not common between the extended framework and the original BPAOntoSOA framework, which is a BPA-based only where its extension is not but goal-based. This section presents a brief explanation and the requirements needed in order to accomplish this framework.

3.5.1 The GQ-BPAOntoSOA Framework Characteristics

In this section, the GQ-BPAOntoSOA framework characteristics are briefly described, where some of them are driven from the former framework. And some are newly produced by the required extension that was inserted in order to address the research problem.

3.5.1.1 Goal-based Emphasising Business Strategy View Support

The modelling of a business process is mainly carried out based on the objectives or the goals of the business process whether the goals are explicitly represented or not. Either way, goals are embedded and they initiate the business process modelling. The BPAOntoSOA framework contributed to the business/IT environment through automatically deriving candidate software services along with their capabilities that stem both from the understanding of the business the organisation is in using its BPA and associated business processes (Yousef, 2010). Similarly, the absence of the desired quality attributes of the business processes and the services may impact the achievement of the associated business processes and software services.

The candidate software services are driven by the alignment of the business strategy view with the Riva-BPA for an organisation where the former comprises goals, goal holders and the relationships between them (i.e., rationales). In short, the framework is goal-based in order to initiate the desired alignment. This feature has bridged the gap between the goal-oriented models and the BPA including its business process models. Accordingly, this is anticipated to adhere to goal changes with implication on business processes and software services. In the GQ-BPAOntoSOA framework, the goal-based and quality-linked Riva-BPA driven outputs are the identification of candidate software services, their capabilities and the associated quality requirements, namely QoS.

3.5.1.2 Ontology Based

The GQ-BPAOntoSOA framework is based on the GQOnt ontology and the GQ-BPAOnt ontology (i.e., a goal-based and quality-linked refinement of BPAOnt ontology of the BPAOntoSOA framework). The former ontology is a fundamental contribution to this research, where the latter is refined using its original design in order to semantically bridge the gap between the GO models (i.e., strategic view), the BPA and associated business processes. The bridging is anticipated to be another main contribution to this research that will propagate its implication on the identified candidate software services by the end of the instantiation of the newly refined GQ-BPAOntoSOA framework.

The benefits gained from employing ontologies in the new framework are as follow: (1) ontologies can represent wide and complex knowledge used within an organisation, (2) ontologies can assist stakeholders in having an agreed knowledge representation and understanding with the help of meta-models. It was agreed that ontologies assists in providing a common communication language between stakeholders and 'automatic manipulation' (Yousef, 2010) with the traceability support, (3) the used ontologies in the new framework have another benefit that they work as a semantic gateway to other ontologies that are not linked. For example, the GQOnt and the GQ-BPAOnt ontologies are both separately designed and then linked afterwards to obtain traceability and the desired alignment.

3.5.1.3 Domain Independence

This characteristic is common to the original and the extended framework. However, in the new framework, this feature emerges stronger than in the original framework due to the extension proposed based on the integration of goals and quality requirements.

In the new framework, the business strategy view (i.e., GO models for an organisation) of an organisation is generically represented using ontologies. The ontological representation of the strategic view can be used to solely represent the strategic view of the organisation or to be aligned with its corresponding BPA and associated processes.

3.5.1.5 Goal- and Quality-based Enrichment of the Service Identification Process

In this research, it is proposed reusing the service identification approach that was originally developed in (Yousef, 2010), but to enrich it with the integration of goals and quality requirements in order to adjust it with the purpose of the new framework. Although the service identification approach is enriched, still it is anticipated maintaining its simplicity. This is because to address the abstraction of the Riva BPA even after enriching it with goals and quality requirements. The original automation of the service identification approach (Yousef, 2010) is addressed by using the GQ-BPAOnt ontology instead of the original BPAOnt one.

3.5.2 The Essential GQ-BPAOntoSOA Framework Requirements

The GQ-BPAOntoSOA framework is constructed and developed based on addressing three main requirements in relation to addressing the research questions defined in Chapter 1. The three main requirements are as follow:

1) Reuse the BPAOntoSOA framework as the foundation of the GQ-BPAOntoSOA framework

This is a substantial requirement such that without it the research is not initiated. Accordingly, approaches embedded in the BPAOntoSOA framework are used and a gap analysis is performed to address goals and quality requirements. Therefore, the two sub requirements that emerge in addressing this main requirement are:

i) Reuse the Riva BPA modelling approach employed in the BPAOntoSOA framework (i.e., in the first layer) as a gap bridging method between the business and SOA environment in order to explicitly or implicitly allocate the missing concepts of goals and quality requirements within the Riva's context. Since the Riva method is the backbone of the BPAOntoSOA framework, the author is required to carry out a rethinking of the Riva method from the point of view of goals and quality requirements in order to enrich the BPAOntoSOA framework.

ii) Adapt the Riva BPA-driven SI approach developed in the BPAOntoSOA framework (i.e., in the second layer) with regard to the integration of service goals and the QoS requirements. This involves extending the current SI approach, which uses the notion of clusters, employed in the BPAOntoSOA framework considering, but adhering to SOA principles. This guides in answering the first and third research questions (RQ1) & (RQ3) presented in Chapter 1,

"What are the current shortcomings of the BPAOntoSOA framework in relation to the integration of the organisation's business goals and quality requirements?"

"How to utilise the BPAOntoSOA framework to identify QoS requirements for its pre-existing identified candidate services using the goal-based and the quality-linked Riva BPA?"

2) The development of the GQOnt ontology

Although, it was apparent from the state of the art literature review that goals and quality requirements (i.e., or soft goals) are categorised as two different areas, still they overlap and relate to each other in away or another. In simple words, the goals (i.e., by default mean hard goals) embody the concept of the behavioural or the functional requirements from an abstract point of view. Similarly, the quality requirements or soft goals embody the notion of the non-behavioural or the non-functional requirements. The generic relationship between the two is derived from the literature that the latter (i.e., quality requirements) constrains the former (i.e., goals).

In order to integrate goals and quality requirements into the BPAOntoSOA framework, this may take the form of one of the two approaches below. **The first approach** states that the integration of goals and quality requirements concepts are addressed through cementing them into the BPAOntoSOA framework without constructing another layer. **The second approach** states that the integration of the goals and quality requirements into the BPAOntoSOA framework are addressed by constructing them within an additional layer that aims to semantically support their representation. This approach follows the "separation of concerns" design principle (Kotonya and Sommerville, 1998). One could initiate a debate regarding the two approaches and the appropriate design decision. However, it was perceived that the business strategy view for an organisation employing the BPAOntoSOA framework is absent. In short, the strategic view is shaped around the drivers of the organisation that are goals, which are represented using GO modelling approaches that support the decomposition of goals and their traceability through the organisation. The goals of an organisation must be constrained by soft goals that determine the desired attributes (e.g., security). Accordingly, the second approach is followed regarding the integration of goals and quality requirements into the BPAOntoSOA framework. Hence, the need arises to distinguish between the GO model and the related quality-oriented models requiring the development of the semantics of the GO models (siGoal ontology) through conceptualising the elements of the GO approach, and the development of the semantics of the quality-oriented model (sQuality ontology) through conceptualising the elements of the quality-oriented approach.

Thus, achieving these path the way in answering the first and the second research questions (RQ1) & (RQ2):

"What are the current shortcomings of the BPAOntoSOA framework in relation to the integration of the organisation's business goals and quality requirements?"(RQ1)

"How to bridge the gap between the goal-oriented models and the Riva-based business process architecture?"(RQ2)

3) **Bridge the gap between the GQOnt ontology and the BPAOntoSOA framework**
The required bridging between the developed GQOnt ontology (i.e., satisfying the second requirement), which will be presented in Chapter 4, and the original framework generates an enriched BPAOntoSOA framework. This will be mainly achieved by aligning the GQOnt and GQ-BPAOnt ontologies. This also entails linking quality requirements (i.e., soft goals) that are linked with the related GO models in the business strategy view and the Riva BPA model along with its associated business process models. Hence, this contributes to answering the second research question (RQ2):
"How to bridge the gap between the goal-oriented models and the Riva-based business process architecture?"

The resultant GQ-BPAOnt ontology facilitates the alignment between the business strategy view for an organisation and its Riva BPA. Accordingly, this impacts the derivation of candidate software services and hence contributing to answer the fourth research question (RQ4):
"What is the implication of integrating business goals and quality requirements on the pre-existing BPAOntoSOA-driven candidate SOA's services?"

3.6 The Semantically Goal-based and Quality-linked (GQ-) BPAOntoSOA Framework

The GQ-BPAOntoSOA framework is a three-layered model that sets up the required strategic-, business- and software service-oriented components and the relationships between them. The GQ-BPAOntoSOA framework, depicted in Figure 3.4, is designed based on the separation of concerns design principle (Kotonya and Sommerville, 1998). Moreover, a separation of concerns is used between the GO model (identified using one or multiple related GO approaches (Kavakli and Loucopoulos, 2005)) and the linked quality-oriented model (identified and represented using the NFR framework approach (Chung et al, 2000)). Based on this separation of concerns, the siGoal ontology is developed, where the *i** framework is employed and conceptualised, for the former and the sQuality ontology is developed for the latter, where the NFR framework is conceptualised. Both ontologies are original contribution in this research work.

The input to the GQ-BPAOntoSOA framework is the business strategy view for a business organisation. The view is based on business goals that comprise both the hard, which is represented using GO modelling approach, and soft goals, which are represented using soft goal-oriented approach. Then, the strategic view will be aligned with the non-goal-based

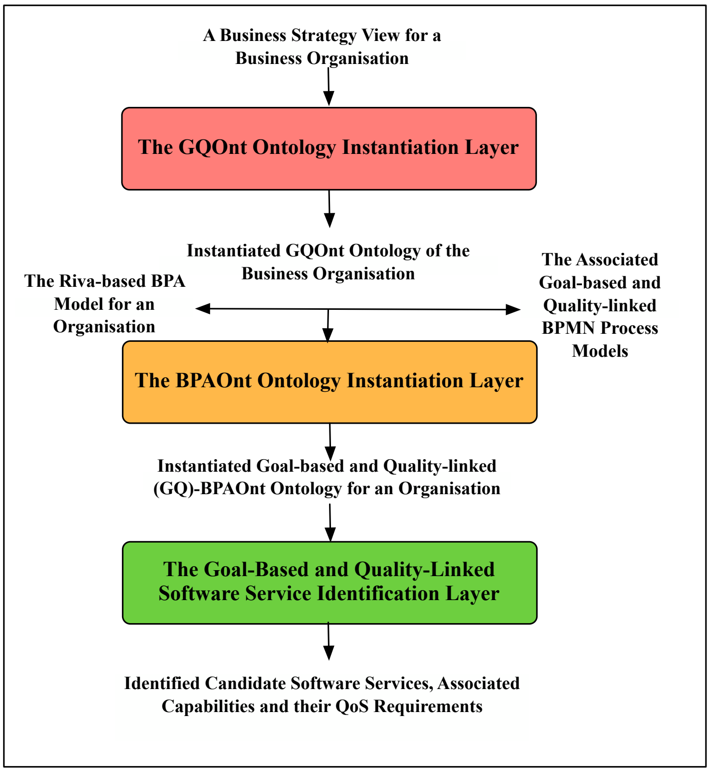


Figure 3.5: The Abstract Architectural Representation of the GQ-BPAOntoSOA Framework.

Riva-BPA. The output is the semantic representation of capabilities and associated QoS requirements of the goal-based and the quality-linked Riva-BPA-driven candidate software services.

Next, the GQ-BPAOntoSOA framework layers are presented and explained along with their components. Some of the new framework components are adopted, and reused from the original BPAOntoSOA framework.

3.6.1 The GQOnt Ontology Instantiation Layer

This layer orients its goals, both hard and soft to the Anton et al., goal definition (Anton et al., 1994) as discussed in Chapter 4. The layer is required to semantically represent a business strategy view for a business organisation, where many of its goal-based business processes are executed as software services. Therefore, the GQOnt ontology is created in order to generate the semantic conceptualisation of hard and soft goals for a particular business organisation. The input is the business strategy view of an organisation that is simply comprised of interrelated GO models that are constrained with the linked soft goals that are represented using soft goal-oriented modelling approach.

The output is the GQOnt ontology of the business organisation. The layer comprises three main components that are the siGoal ontology instantiator, sQuality ontology instantiator and the siGoal and sQuality ontologies linker. They all collaborate in order to generate the GQOnt ontology outcome (i.e., GQ meta-model for an organisation). This strategic-oriented layer along with its components will be discussed in detail and initially evaluated in Chapter 4. This layer and its components are represented in red colour in Figure 3.6.

3.6.1.1 The Semantic Interrelated Goal-Oriented Models (siGoal) Ontology Instantiator

The function of this component is to produce the formal semantic representation of interrelated GO models for an organisation considering that the *i** framework is the backbone for the semantic representation, as justified and discussed in Chapter 4, Section 4.2. Therefore, the letter ***i*** is used to denote to both the notion of <u>**i**</u>nterrelated GO models (Kavakli and Loucopoulos, 2005) and the employment of the <u>***i****</u> framework (Yu, 1995).

The input to this component is the interrelated GO models with the associated *i** framework exploited from the BSV. The output is the instantiation of the abstract siGoal ontology for an organisation. The interrelated GO model elements and their relationships are conceptualised using the abstract siGoal ontology. The classes that represent the GO models are for example goal, soft goal, task resource, actor, SD goal dependency, SD soft goal dependency, SD model, and the SR model as will presented in detail in Chapter 4. The siGoal ontology development using OWL-DL language with the help of the protégé tool, will be presented in Chapter 4 (Section 4.3.3). Using the siGoal ontology instantiator with the support of the Protégé ontology development tool, the siGoal Ontology can be easily and systematically to some extent instantiated for a given situation within the organisation.

3.6.1.2 The Semantic Quality (sQuality) Ontology Instantiator

This component has the ability to formally represent a quality-oriented model such as the NFR framework (Chung et al, 2000) within the business strategy view for an organisation using the OWL-DL ontology language. The input is the quality-oriented or soft goal-oriented models exploited from a business strategy view of an organisation. The abstract NFR framework ontology comprises the conceptual meta-model elements and interrelationships such as NFR type soft goal, an operationalisation, NFR soft goal, AND contribution, OR contribution, SIG diagram, etc. Since the NFR framework will be employed in this layer (for a set of reasons discussed in Chapter 4, Section 4.2), then its conceptualisation, which is the outcome from this component, is considered as one of the rare attempts to delivering the formal representation of an NFR framework.

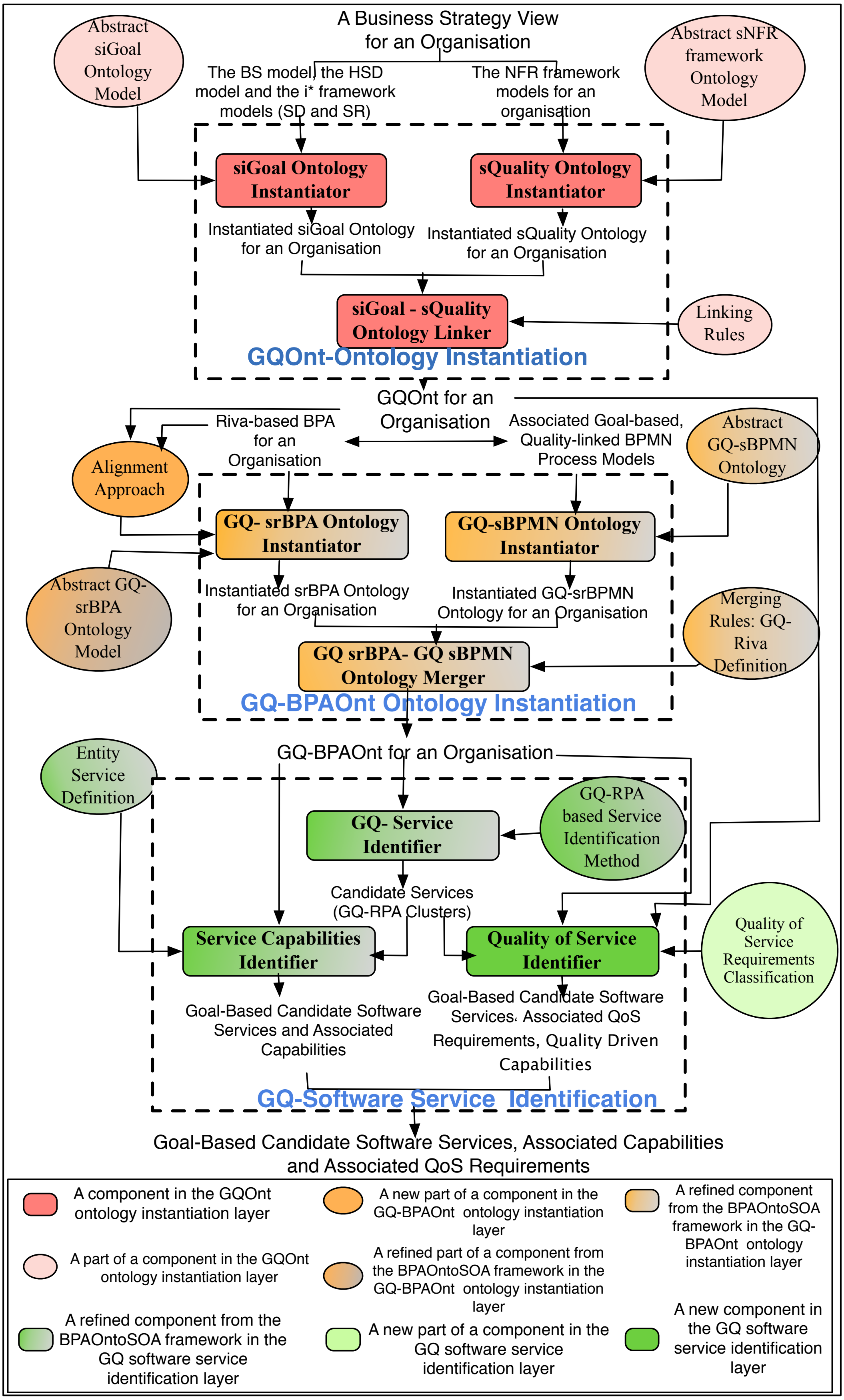


Figure 3.6: The GQ-BPAOntoSOA Architectural Framework.

3.6.1.3 The siGoal-sQuality Ontology Linker
The function of this component is to bridge the input that comprises of the instantiated siGoal Ontology and the instantiated sQuality Ontology for a particular organisation by detecting common logical entities and/or linking rules to operationalise the bridging in the form of merging the two ontologies. The systematic merging rules between the two ontologies are defined after studying the two ontologies and searching for the common logical entities and/or the linking rules, as will be discussed in Chapter 4. The output is the instantiated GQOnt ontology for an organisation.

3.6.2 The GQ-BPAOnt Ontology Instantiation Layer

This layer stems from the original BPAOntoSOA framework. It was established in order to generate the Riva-BPA and associated business processes ontology for an organisation. The input to this layer is the GQOnt ontology for an organisation, the pre-existing not necessarily goal-based and not quality-linked Riva BPA and associated BPMN business processes that are now goal-based and quality-linked. In this framework, this layer is extended using the above GQOnt ontology instantiation layer. The GQ-BPAOnt ontology instantiation layer outputs a goal-based, quality integrated Riva-BPA and associated business processes ontology from the alignment of the GQOnt with the pre-existing not goal-based Riva-BPA and associated not goal-based and quality-linked business processes. This business-oriented layer along with its components will be discussed in detail along with an initial evaluation of the layer's function in the work of Chapter 5. In Figure 3.6, this layer and its components are illustrated in orange colour and the mixed grey-orange colour denotes that component is reused with refinements from the BPAOntoSOA framework.

3.6.2.1 The Goal and Quality-based Semantic Riva-BPA (GQ-srBPA) Ontology Instantiator
This component outputs the formal semantic representation of a goal and quality based srBPA of an organisation using the OWL DL ontology language. Therefore, it extends the srBPA ontology instantiator component in the BPAOntoSOA framework in order to produce the GQ-srBPA ontology from using the notion of alignment with the help of the GQOnt ontology, pre-existing Riva BPA, associated goal-based and quality-linked processes. The original srBPA ontology is extended in order to position itself with the refined Riva method. Thus, the GQ-srBPA is an extended abstract ontology that formally conceptualises the srBPA elements along with the new ones, as will be shown in the work of Chapter 5.

In this component, the alignment approach will be carried out through detecting the EBEs using the GQOnt ontology instead of brainstorming them, as will be discussed in detail in Chapter 5. Hence, this is anticipated to minimise the required effort in deriving the EBEs and to maximizes potential towards the full-automation of the overall framework. This is considered as a further enhancement the original BPAOnt Ontology Instantiation Layer.

3.3.2.2 The Goal-based and Quality-linked Semantic BPMN (GQ-sBPMN) Ontology Instantiator

This component is reused and lightly refined from the BPAOntoSOA framework. In fact the origin of this component is the super project (SUPER, 2008a). In the GQ-BPAOntoSOA framework, this component retains as in its original function, input and output but with respect to integrated goals and quality requirements.

3.3.2.3 The GQ-srBPA-sBPMN Ontology Merger

This component carries out the integration of the two aforementioned ontologies, which are the GQ-srBPA and GQ-sBPMN. The merger uses the opportunity of the ontologies importing which is functionally supported in the Protégé tool. The former merging SWRL rules, which assisted in linking the two ontologies in the original srBPA-sBPMN Ontology Merger (Yousef, 2010), remain and are utilised as they are in the original component even after extending the original abstract srBPA Ontology and sBPMN Ontology. This is because the merger component is also entirely borrowed and reused as it is from the original framework. In addition, few merging rules are added to address the integration of goals and quality requirements in the two components above. Although the input is enriched with goals and quality requirements, still the original merging rules are applicable within the context of the GQ-BPAOntoSOA framework. Hence, the output is the GQ-BPAOnt ontology that semantically represents the enriched merging with respect to goals and quality requirements.

3.6.3 The Goal and Quality based Software Service Identification Layer

This layer derives from the software service identification layer in the original BPAOntoSOA framework. It carries out the generation of the semantic identification of software services from the input GQ-BPAOnt ontology instantiation for an organisation. The software service identification does not solely derive the software service capabilities; it also involves the identification of the quality of these software services. In order to derive the identified software service capabilities and the associated quality requirements, three components are operationalised in this layer: (1) the GQ-service identifier, (2) the service capabilities identifier and (3) the QoS identifier. It is clear that the first component is adapted from the BPAOntoSOA framework (Yousef, 2010) in order to be refined with regard to the goals and the quality requirements integration in the GQ-BPAOnt ontology. It is represented in grey-green colour component in Figure 3.6. The second component is borrowed with very light refinements on its function from the BPAOntoSOA layer (Yousef, 2010) therefore, it is represented in grey-green colour in Figure 3.6. The third component is newly established in order to address the missing module of the driven software services, which is the desired quality requirements and it is represented in green colour. This service-oriented layer along with its components will be discussed in detail and initially evaluated in the work of Chapter 6.

3.6.1 The Goal-based and Quality-linked (GQ) Service Identifier

This component inputs the GQ-BPAOnt ontology instantiation for an organisation and outputs the corresponding goal-based and quality-linked -RPA clusters. In particular, this component has been modified in order to incorporate the integration of goals and quality requirements into the original BPAOnt that resulted in the GQ-BPAOnt ontology. Therefore, this component has been illustrated in mixed green-grey colour in Figure 3.6. With the reference to the service identifier component in the BPAOntoSOA framework (Yousef, 2010), the component utilises the proposed novel approach of identifying the software services from the BPAOnt ontology using the RPA clusters (Yousef, 2010).

3.6.2 The Service Capabilities Identifier

The purpose of this component is the same as in the original BPAOntoSOA framework, but taking into account the integration of goals and quality requirements (Yousef, 2010). The software service capabilities are extracted from the GQ-sBPMN ontology embodied in the GQ-BPAOnt and the goal-based and quality-linked RPA clusters where quality requirements that constrain the capabilities fulfilment are addressed in the next component below.

3.6.3 The Quality of Service Requirements (QoS) Identifier

The purpose of this component is to identify the QoS requirements that are associated with the identified software services capabilities. The QoS requirements refer to the desired of attributes or constraints on the software service capabilities. This component uses the GQ-RPA generated from the GQ-BPAOnt. This is because quality integration is explicit in the GQ-srBPA ontology using the GQOnt ontology, as discussed in Chapters 5 and 6.

3.7 Comparing the BPAOntoSOA and the GQ-BPAOntoSOA Frameworks

The BPAOntoSOA framework was developed to semantically derive Riva-based BPA candidate services that adhere to SOA principles (Yousef, 2010). This framework had functionally contributed to BIA, particularly in a dynamic environment through using the BPA and associated process models. However, the derived services do not address the QoS requirements. Moreover, the BPAOntoSOA stems solely from what business an organisation is involved in rather than the rationale(s) of this business. Consequently, the BPAOntoSOA framework services are neither constrained with quality requirements nor stemming from business goals. Hence, this emphasises on the absence of a BSV that must exist according to OMG (OMG, 2013). These two main flaws motivated the requirement for a framework that integrates both business goals and the quality requirements.

From the above, this section aims to compare the original and the new framework by manifesting their differences and similarities in order to highlight modifications and acquired features.

3.7.1 Input/Requirements and Output/Results

The original framework required a Riva-based BPA and its associated business processes in order to derive candidate software services along with their behavioural capabilities (i.e., the software service functional requirements) that adhere to the SOA principles in a dynamic environment.

However, the new framework requires the BSV for an organisation and the pre-existing BPA (i.e., neither goal-based nor quality-linked pre-existing BPA) used in the BPAOntoSOA framework. The former input encompasses interrelated GO models linked with their related soft goal-oriented models and the second is the as-is Riva-based BPA and associated goal-based and quality-linked business processes. These two inputs are aligned using the new framework in order to output candidate software services that are refined in terms of their identification, capabilities and QoS integration.

3.7.2 Framework Architectural Design

The earlier framework was designed in a top-down layered architecture. Two major layers structure the framework, where the top layer is the BPAOnt Ontology Instantiation Layer and the bottom is the Software Service Identification Layer.

The new framework is designed to extend and enrich the old one depicted in Figure 2.29. In particular, three cohesive layers are created to establish the new framework in the form of top-down layout architecture. The first layer is the GQOnt Ontology Instantiation Layer, which is the newly established goal-oriented layer. The second and the third are borrowed, refined and then utilised from the old framework in order to extend its functionality. The BPAOnt Instantiation Layer occupies the second layer in the new framework where the SI Layer is the third. The first layer is concerned with generating the business strategy view ontological representation for an individual organisation, namely the GQOnt ontology. This ontology embodies hard and soft business goals and the relationships between them. The hard goals are realised with goal-oriented modelling (kavakli and loucopoulos, 2005) in the siGoal ontology and the soft goals are realised with quality requirements modelling approaches (Chung et al ,2000) in the sQuality ontology. This GQOnt ontology aligns with the two refined layers to form an extended top-down architectural framework, namely the GQ-BPAOntoSOA framework. In other words, the GQOnt ontology is established to operationalise the two layers optimisation.

The new framework extends and enriches the current dynamism of the Riva methodology (Ould, 2005) by allowing the ability to change in relation to business goals and quality requirements. Therefore, the GQ-BPAOntoSOA framework operates in an environment where not only business processes are continuously changing but also business goals and quality requirements in order to meet changing business needs. This multidimensional business-oriented accommodation in one framework is anticipated to semi-automate the generated candidate SOA services that are not solely aligned to business functional needs, but also by the business goals and desired quality requirements.

3.8 Summary

This chapter presented the DSR paradigm that aims at solving a problem. The research process was presented in detail and shown how the BPAOntoSOA framework is reused in order to carry out the development of the new framework, namely the GQ-BPAOntoSOA framework.

The new and the old frameworks share some common characteristics and differ in some others. For example, they are both ontology-based. However, the earlier is BPA ontology driven and the new is a goal driven framework. The main requirements needed to realise the new framework were discussed and were associated with the research questions.

Furthermore, the architectural representation of the GQ-BPAOntoSOA framework is addressed. Three main layers establish the GQ-BPAOntoSOA framework: (1) the GQOnt Ontology Instantiation Layer, (2) the GQ-BPAOnt Ontology Instantiation Layer and (3) the GQ-Software Service Identification Layer. The first layer is purely new. However, the two others stem from the BPAOntoSOA framework including the refinements to adjust them in order to address the research work objectives.

Finally, The original BPAOntoSOA framework was compared to the new GQ-BPAOntoSOA framework. The comparison is anticipated to guide the reader as to when to apply each framework. The work of this chapter assisted in answering the research questions show in Figure 3.7 and addressing research phases in Figure 3.8.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1			✓						
RQ2			✓						
RQ3									
RQ4									

Figure 3.7: Answering Research Questions for the Work of Chapter 3.

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									**Addressed**
RP2			✓						**Active**
RP3									
RP4(ai)									
RP4(aii)									
RP4(bi)									
RP4(bii)									
RP4(biii)									
RP5									
RP6									

Figure 3.8: Research Phases Activated within the Work of Chapter 3.

Chapter Four: The Goal and Quality Ontology (GQOnt): A Semantic Business Strategy View

4.1 Introduction

This chapter presents the body of work wrapped within the first layer of the GQOnt-BPAOntoSOA framework, namely the GQOnt Ontology Instantiation Layer. It has been already mentioned, in Chapter 3, that this layer is newly established in order to extend the BPAOntoSOA framework unlike the rest of the layers, which will be refined as part of this research (in Chapters 5 and 6) in order to adjust the integration of goals and quality requirements in the GQOnt Ontology Instantation Layer. The GQOnt ontology development is an essential novel contribution of this research representing the second requirement needed to attain the proposed GQ-BPAOntoSOA framework. In addition, the GQOnt ontology is an input requirement for the alignment proposed with the BPAOnt Ontology Instantiation Layer from the original BPAOntoSOA framework in (Yousef, 2010), as will be discussed in Chapter 5. The input into the GQOnt Ontology Instantiation Layer is the BSV, which represents any interrelated goal- and soft-goal models of an organisation. The soft goal models represent the constraints on the associated goal models. The BSV has been already defined in Chapter 2 following OMG (OMG, 2013); however, the work of this chapter aims at rethinking this defintion. The generated output from the GQOnt Ontology Instantiation Layer is the semantic representation of a business strategy view using the OWL-DL language.

The GQOnt ontology instantiation layer conceptualises the goal and the soft goal elements of the BSV models for an organisation and the relationships between them, resulting in the GQ Meta Model ontology. In particular, the GQOnt ontology consists of two sub ontologies, the **siGoal ontology** represents the relevant goal-oriented models' following the *i** framework, and the **sQuality ontology** represents the NFR framework elements of the soft goal-oriented models. The next section presents a detailed justification regarding the selection of the *i** framework GO approach and the NFR framework soft goal approach among other alternatives. Hence, the strategic models of an organisation are goal and soft goal- oriented due to the combined employment of both the *i** framework (Yu, 1995) and the NFR framework (Chung et al, 2000). It is necessary to denote that in this chapter, the formal representation of the business strategy view models does not take into account the business process architecture of an organisation or reusing the original BPAOntoSOA framework (Yousef, 2010).

The work in this chapter takes place in the three research phases, namely that are the early theoretical framework design phase, the investigation phase and finally the original BPAOntoSOA framework enhancement phase. In addition, this chapter addresses the first research objective identified in Section 1.3. The chapter is structured as follows. Section 4.2 presents in detail the rationales beyond the GQOnt ontology instantiation layer construction. Both the framework of the GQOnt ontology instantiation layer along with a detailed explanation of the business strategy view definition are presented in Section 4.3. Also, this section presents the OWL-DL implementation of the major components of this layer, namely the siGoal ontology instantiator, sQuality ontology instantiator and the siGoal-sQuality ontology linker. Section 4.4 presents the demonstration of the GQOnt ontology instantiation using the patient reception process pilot study in (Abu rub, 2006). Finally, the work of this chapter is discussed and concluded in Section 4.5 that involves answering the relevant research questions.

4.2 The Rationales beyond the GQOnt Development

The GQOnt ontology instantiation layer occupies a strategic role, which is earlier and higher than the BPAOnt ontology instantiation layer and the service identification layer in BPAOntoSOA framework (Yousef, 2010). This is because the GQOnt will rationally derive the integration of goals and quality requirements into the BPAOnt using the alignment notion, as discussed in Chapter 5. In addition, the GQOnt ontology development is regarded as the second requirement in order to realise the GQ-BPAOntoSOA framework, as discussed in Section 3.5.2. The GQOnt ontology is a standalone, rich and reusable resource that reserves all relevant knowledge with regard to the organisation's business strategic view, as will be shown in this chapter.

The significance and the benefits of the GQOnt ontology originate from its role and from the strategic view. The benefits gained from the GQOnt ontology are as follows:

First, the GQOnt ontology is designed to elicit and preserve the identified business strategy elements such as the hard and the soft business goals, their holders and dependency relations using the GQOnt ontological Meta Model. The goal- and quality-oriented concepts (i.e., business strategy view elements) are reserved in one repository and have the potential to transfer their means to the associated business process models afterwards due to their richness of the relevant goal- and business-oriented concepts (e.g., actor, role, goal, soft goal, task, resource, etc.). Also, the GQOnt repository can be reused to review the organisation's strategies and tactics in addressing the identified business hard and associated soft goals. Therefore, the GQOnt ontology instantiation layer is BPAOntoSOA-independent. The GQOnt ontology will be necessary to bridge the gap between the strategic models, that are the goal and the soft goal models, and the BPA and associated BP models, as discussed in Chapter 5.

Second, the main reason behind developing the GQOnt ontology is the second and the third requirements that are presented in Chapter 3 Section 3.5.2, needed to achieve the GQ-BPAOntoSOA framework.

Third, the GQOnt ontology development fills the current gap that is the absent business strategy view for an organisation. This problem was previously mentioned in Chapter 1 Section 1.2. It is necessary to pay the effort in presenting unambiguous business strategy view, as it embodies the drivers (i.e., goals) of the business for an organisation. The conceptualisation of those drivers is anticipated to facilitate carrying out a gap analysis through tracing the current business activities to their original drivers (i.e., goals). In addition, the use of ontologies assists stakeholders in sharing unambiguous information about the business drivers, and hence the associated business activities.

Fourth, the GQOnt ontology will support the current and the future research work that contributes to leveraging IT environments based on the alignment with the business domain's strategic view, as will be shown in the work of Chapter 6. In this research, the GQOnt ontology will engage in bridging the gap between the GO models and BPA. For example, the GQOnt ontology will semantically process the alignment with a Riva-BPA, as discussed in Chapter 5.

Recalling that the GQOnt ontology is comprised primarily of the siGoal ontology that conceptualises the GO models and the sQuality ontology that conceptualises the quality-oriented models. The *i** framework (Yu, 1995) forms the backbone for the former ontology, where the NFR framework (Chung et al, 2000) is conceptualised for the quality-oriented models in the latter ontology. The reasons beyond employing these two approaches in this research project are presented next.

The GQOnt ontology employs the *i** framework as a goal-oriented approach (Yu, 1995). This approach was chosen as the most appropriate from other alternatives owing to many reasons, as have been previously briefly discussed in Chapter 2. First, it is the only goal-oriented approach found in the literature that is subjected to formally understand and characterise a business organisation with rich business-related concepts to this research (Kavakli and

Loucopoulos, 2005). Second, it is a simple approach to comprehend. Third, it provides two strategic models that are the SD and SR as coarse-grain and fine-grain strategic models, respectively. Fourth, it pays attention to actors as dependers and depndees for the goals and/or soft goals. It is very significant to highlight actors in goal-oriented models owing to the fact that goals achievement cannot be carried out in the absence of actors. The explicit notation of an actor, that is assigned as depender and/or dependee, credits the *i** framework among other GO approaches (Kavakli and Loucopoulos, 2005). The reasons do not end here as the *i** framework was chosen for further benefits gained from its SD and SR models.

With regard to the *SD model* benefits, **first**, it blueprints the exterior structure of an organisation with respect to goals and their holders (i.e., actors) in the form of external dependency networks that are in coarse-grain manner. The networks assist in a way or another in agreeing on organisation's boundary. Moreover, the SD model is represented without showing how the dependencies are addressed or fulfilled. **Second,** since the SD model dependency networks do not provide any information about the rationales, then the dependencies with their ends facilitate an external traceability mechanism. Finally, the SD model is described as a flexible model that allows freedom for actors to discover their broader implications (Yu et al, 2010).

With regard to the *SR model* benefits, it elaborates the SD model and thereby it is described as a fine-grain strategic model. **First**, it '*describes the actors internal structure they participated in the SD model through representing thier intentional strategic entities and relationships'* (Yu, 1995)**. Second**, on the contrary of SD model, the SR model is anticipated to assist this research work through presenting an ***internal traceability mechanism***. The SR model provides rationales about the backend business processes through its rationale relationships. This is also a credit for the SR-model and to be added to the i* framework. Hence, this research work will utilise these external and internal traceability mechanisms, where both are joined in SD and SR models, in order to address the research objectives.

As the *i** framework lacks the higher GO representation of the SD and the SR models, the GO designer is not aware of the drivers behind the *i** framework models. This shortcoming will be resolved in Section 4.3.1.

The NFR framework conceptualisation represents the quality ontology in the GQOnt ontology, and it was selected due to many reasons, as briefly mentioned in Chapter 2. Essentially, the NFR framework is the only candidate quality-oriented approach that is concerned with soft goals for software systems development (Kavakli and Loucopoulos, 2005). The NFR framework is a process-oriented approach, and this is anticipated to facilitate the conceptualisation and the linkage with relevant approaches. Moreover, the NFR framework is very rich of relevant concepts that elegantly map with the *i** framework's concepts (Chung et al, 2000). This harmony, between the *i** framework and the NFR framework, motivated the development of the GQOnt ontology that attempts to generate a comprehensive semantic representation of the organisational goals and soft goals, as will be shown with pilot study example in Section 4.4. Accordingly, the conceptualisation of the NFR framework establishes a repository of quality-oriented knowledge that paves the way to stakeholders to share it with its common syntactic and semantics. Since the NFR framework is process-oriented approach, then this encourages stakeholders to deal with NFRs conceptualisation systematically.

4.3 The Framework of the GQOnt Ontology Instantiation Layer

In this section, the GQOnt ontology instantiation layer structure is shown in Figure 4.1.This layer is evaluated using a pilot study, which is the patient reception process, extracted from the Jordan CCR case study in the next section.

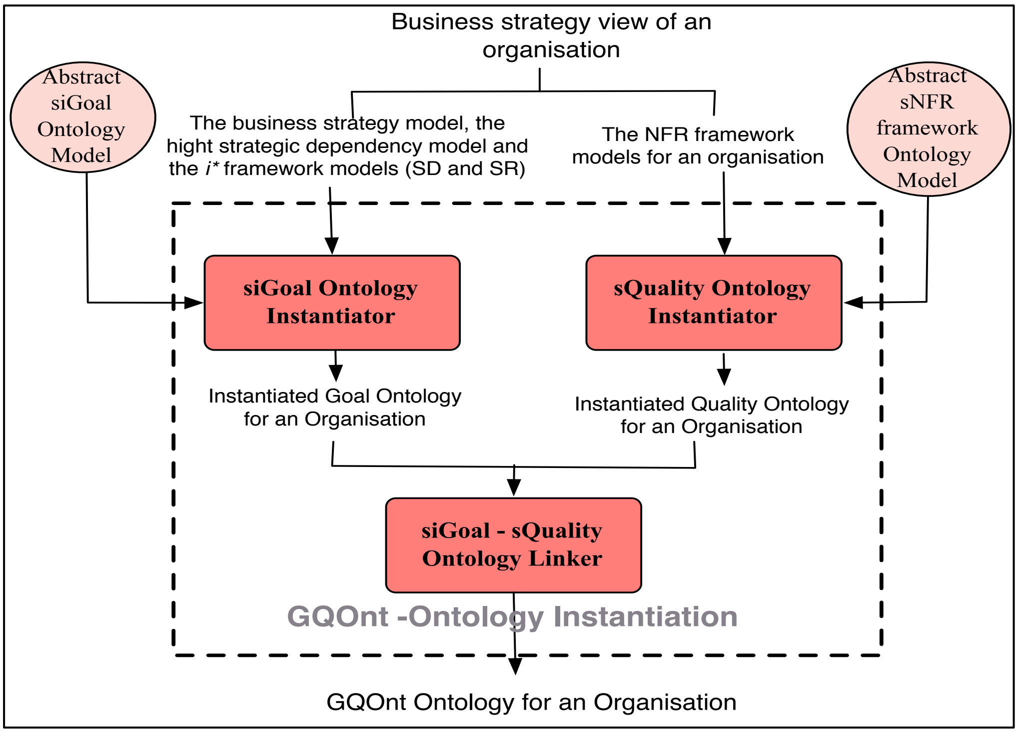


Figure 4.1: The Structure of the GQ-Ontology Instantiation Layer in the GQ-BPAOntoSOA Framework.

In the GQOnt ontology, the conceptualisation of the *i** framework (Yu, 1995), including its SD and SR models is represented in the siGoal ontology. This is because another two goal-oriented models proposed in this research, that are higher in abstraction than the SD model, will be connected to the *i** framework, as will be explained in Section 4.3.1. Accordingly, the siGoal ontology must provide the basic goal elements of the organisation that form the soft/goals, actors and the relationships between them. In addition, the conceptualisation of the NFR framework (Chung et al, 2000); including its SIG and its elements are semantically represented on behalf of the soft goal-oriented models in the sQuality ontology. The GQOnt ontology development entails linking the siGoal ontology and the sQuality ontology, using the siGoal-sQuality ontology linker, in order to produce the GQOnt ontology for an organisation. This is because the *i** framework models in the siGoal ontology encapsulates with harmony the formal representation of the NFR framework using the sQuality ontology. By and large, the basic skeleton of the GQOnt is comprised of the interrelated GO models, where the *i** framework is one of them, along with the relevant NFR framework models.

The siGoal and the sQuality ontologies are developed to generate the GQOnt ontology as an independent product in this research. Thus, the GQOnt ontology product is used when the business strategy view of an organisation is required for instantiation in order to get its actual strategic elements for some reason or another (e.g., aligning the organisation's systems with the recent business strategy, reuse the strategic view for a business process redesign, etc). In the siGoal ontology, the GO concepts and relationships are represented using the OWL-DL and hence the GO concepts include elements such as the actors, goals, soft goals, tasks, etc. Similarly, the sQuality ontology does formally specify the NFR framework elements, such as NFR type soft goal, NFR soft goal, operationalization, decomposition, etc., and sets the relationships between the elements.

In the next sub-section, a rethinking of the OMG's (OMG, 2013) definition or the meaning of a BSV for an organisation is presented and explained. In short, the next sub-section aims at making the business strategy view comprising of the goal-oriented models (one of them is the *i** framework) and the associated soft goal-oriented models (the NFR framework). The integration of the NFR framework into the *i** framework, as part of the goal-oriented models, is carried out using any of the four linking cases in Section 4.3.1.5.

It is necessary to denote that the GQOnt development is not fully automated. This is because a human intervention is required somewhere in order to obtain the instantiation. For example, the human has the responsibility to decide which goal is a specialisation of another. Another example, the human has the responsibility to identify actor abilities and intentions in order to automatically then determine dependency relations.

This section is structured as follows. Section 4.3.1 proposes a rethinking of the current OMG's definition and meaning of the business strategic view of an organisation. In addition, it presents a method for constructing the business strategy view for this research including the goal- and soft goal-oriented models employed in this view. Section 4.3.2 presents the ontology development tool used to specify the first layer's components in the GQ-BPAOntoSOA framework, as shown in Figure 4.1. The siGoal ontology design and development are discussed in Section 4.3.3. Section 4.3.4 presents the sQuality ontology design and development. Finally, the linking of the two ontologies is illustrated in Section 4.3.5.

4.3.1 The Business Strategy View For An Organisation

This sub-section aims at presenting a rethinking of the current definition of the BSV concept for an individual business organisation. The most comprehensive definition for the BSV was previously mentioned in Section 2.2 by the OMG in (OMG, 2013). This sub-section complements the OMG's definition or meaning and reconsiders the Business Strategy View (BSV) for an organisation as follows:

"*The strategic view for an individual organisation is a set of interrelated GO models that elaborate each others, starting from the highest goal until getting the lowest goal in* *abstraction within the organisation, in order to facilitate the early understanding of the business organisation from the goals' point of view. The elaboration is represented in the form of network of goals that drive the organisation forward. The soft goals associate GO component in the GO models as their constraints or desired quality requirements and represented using their relevant methods if needed. Both hard and soft goals, that are strategical elements, are decomposed within their models into tactical elements or approaches where their run would fulfil any/both of the strategic goals such as tasks, hard resources (i.e., physical) and soft resources (i.e., data and information). The representation of strategical goals is aided with traceability in both directions (i.e., from the highest soft/goal to the lowest soft/goal in abstraction and vice versa) and with mapping to appropriate metrics (i.e., qualitative or quantitative) 'that provide ongoing evaluation of how successfully the organisation is achieving its goals' (OMG, 2013). The GO models must consider the participation of the hard and soft goals' holders and their interaction within the models as required active entities.*"

From the above definition, the BSV for a business organisation must provide traceable interrelated GO models that represent their goal holders, strategical elements (hard goals and soft goals) and the tactical elements (tasks, hard and soft resources) considering that both the strategical and the tactical elements are measurable. The OMG's definition necessitates the existence of the BSV, as it is an integral part along with it sibling views within the business architecture part of the enterprise architecture (OMG, 2013).

In this research, the *i** framework has already been utilised to represent part of the GO models within the BSV for an organisation. However, in this research, the *i** framework approach represents its goals without addressing the ultimate goal (i.e., of the organisation). Therefore, this research aims at refining the *i** framework by extending its representation of the GO models through engaging higher GO models in abstraction that are the Business Strategy (BS) model and the High Strategic Dependency (HSD) model in order to meet the aforementioned BSV definition. The BS and the HSD models are originally proposed in this research in order to enrich and fill the current gaps of the *i** framework.

As shown in Figure 4.1, the BSV is divided into two views. The first view is the goal view, the input to the siGoal ontology instantiator component. The second view is the soft goal view, the input to the sQuality ontology instantiator component. In this research, the proposed BSV for an organisation is in the form of a sequence of interrelated GO models linked with the relevant soft goal models according to the aforementioned definition as shown in Figure

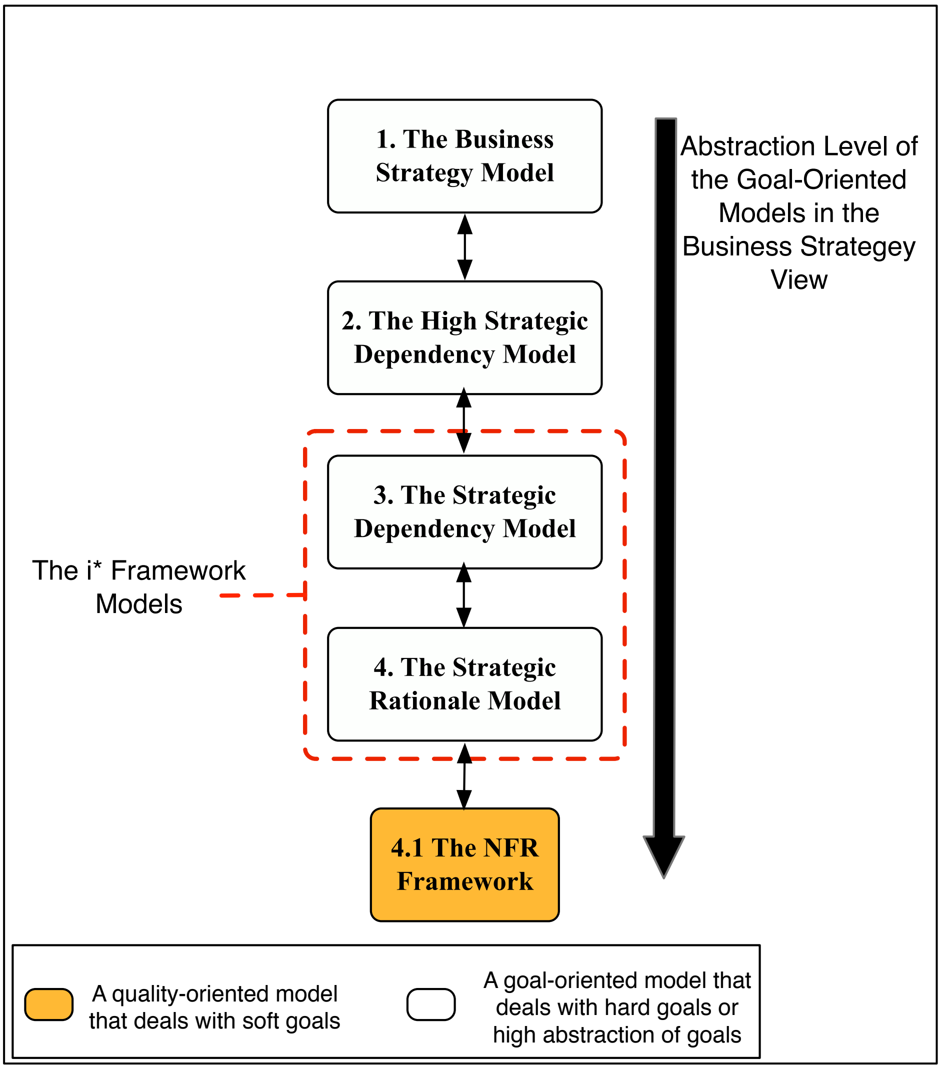


Figure 4.2: The Business Strategy View Proposed for an Organisation

4.2. The models that set up the proposed BSV are: (1) the BS model, (2) the HSD model, (3) the SD model and (4) the SR model. The NFR framework, which represents the quality models (i.e., soft goal models), is linked within the SR model as illustrated in Figure 4.2. In short, the coarse-grain representation of the interrelated GO illustrated in Figure 4.2 fits well the refined definition of the BSV.

It is apparent that the extension proposed for the *i** framework is carried out from two levels as the outer boxes shown outside the dotted red rectangle in Figure 4.2. The first-level is the higher representation of the *i** framework that consists of the BS model and the HSD models respectively. The second-level is the quality-linking representation within the SR model using the NFR framework. The BSV's information for an organisation is obtained through eliciting it from stakeholders using a data collection method(s) (e.g., interviews) or through deriving them from relevant models (e.g., recent Goal-based, Quality-linked and Role-driven Business Process Models (GQR-BPMs)). It is necessary to note that this research is concerned with representing the BSV for an individual organisation that depends on itself rather than on another organisation for addressing its goals. Figure 4.3 Part (a) illustrates the self-dependent organisation whereas part (b) is not the case for this research. Moreover, it is necessary to note that this research does not consider the task and the resource dependencies between the actors. This research is also not concerned with dealing with the risk management (Yu, 1995). Therefore, the refined *i** framework that represent the GO view within the BSV does not consider the vulnerability of the dependencies between the actors. The work of deriving the BSV is a novel part in this research.

4.3.1.1 The Business Strategy Model in the Business Strategy View

This model appears as the first model within the first-level of the refined *i** framework modelling stages as in Figure 4.2. In this stage, the boundary of an organisation and its associated Highest Business Goals (HBGs) must be agreed. A HBG is the ultimate goal of the

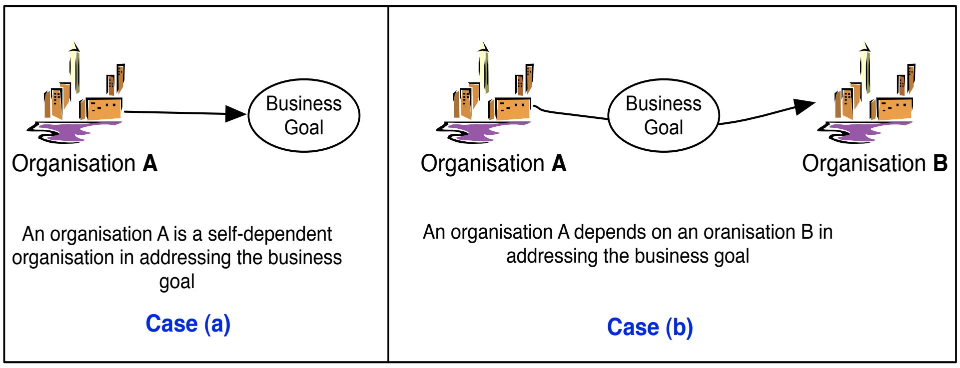


Figure 4.3: Two Dependency Manners in Addressing a Business Goal for an Organisation.

agreed organisation. The canonical list in (Clements and Bass, 2010), which was presented in Section 2.2 assists in selecting at least one HBG for the organisation in this model. A business organisation, which is aimed at deriving its strategic view, refers to an enterprise, a business process, a department or even a group of individuals that are collaborating in order to achieve at least one HBG.

Figure 4.4 illustrates how the BS model is represented to pave the way to design the remained models in Figure 4.2. In Figure 4.4, the building notation on the left hand side represents the agreed organisation where the associated label names it. On the right hand side, at least one HBG must be selected from the canonical list. The "Aims To" relationship must be directed from the business organisation notation (source) to the associated HBG oval (destination). Using the BS model, the stakeholder can early comprehend the highest business goals of the organisation that are explicitly represented without the need to refer to an informal representation (e.g., long texts). Moreover, the BS model triggers the derivation of the further GO models that will embody and represent the proposed tactics in order to meet the identified HBG. Therefore, the BS model is strategical and does not occupy any tactical role in the BSV.

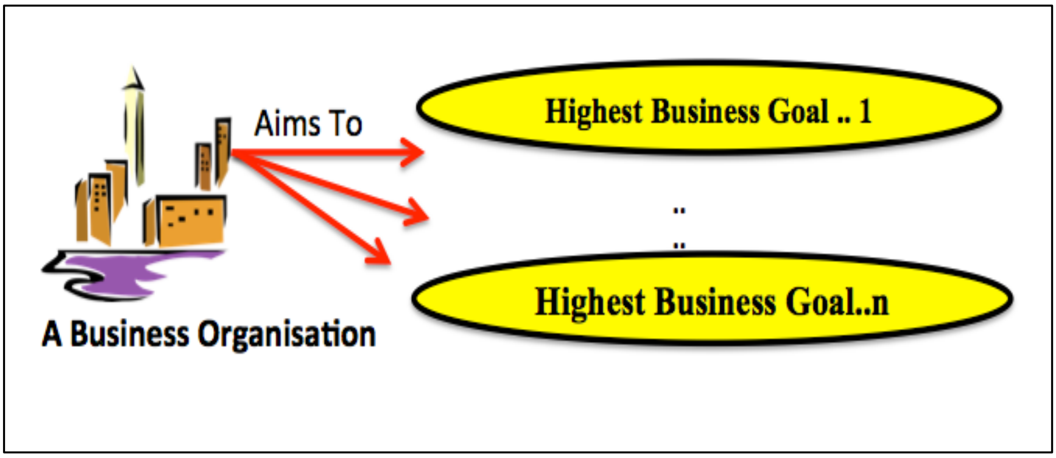


Figure 4.4: The Business Strategy Model for an Individual Self-Dependant Organisation in the Business Strategy View.

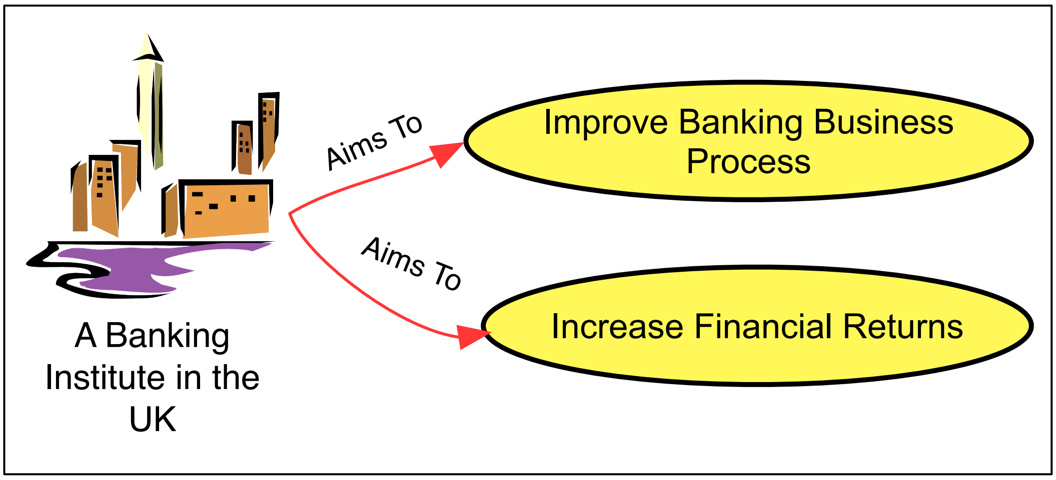


Figure 4.5: An Example of the Business Strategy Model for Banking Institute.

In Figure 4.5, a simple example of the BS for unreal banking institute is illustrated and it is apparent that at least one of the HBGs is borrowed from the canonical list (Clements and Bass, 2010).

4.3.1.2 The High Strategic Dependency Model in the Business Strategy View

This is the second model within the first-level in the refined *i** framework design that results the first goal-oriented dependency model, namely the HSD. Each HBG in the BS model is decomposed using the goal decomposition link into associated sub goals, namely Immediate Highest sub Goals (IH-Gs). The relation between the HBG and the HSD model is defined as, that is, for each HBG in the BS model, there must be at least two IH-Gs that fulfil that HBG, as depicted in Figure 4.6. The resulted IH-Gs must generate at least one HSD model. As from the HBG perspective, the IH-G is defined as the set of the immediate decomposed goals that make up the HBG parent. And from a GBP perspective, the IH-G is defined as the main

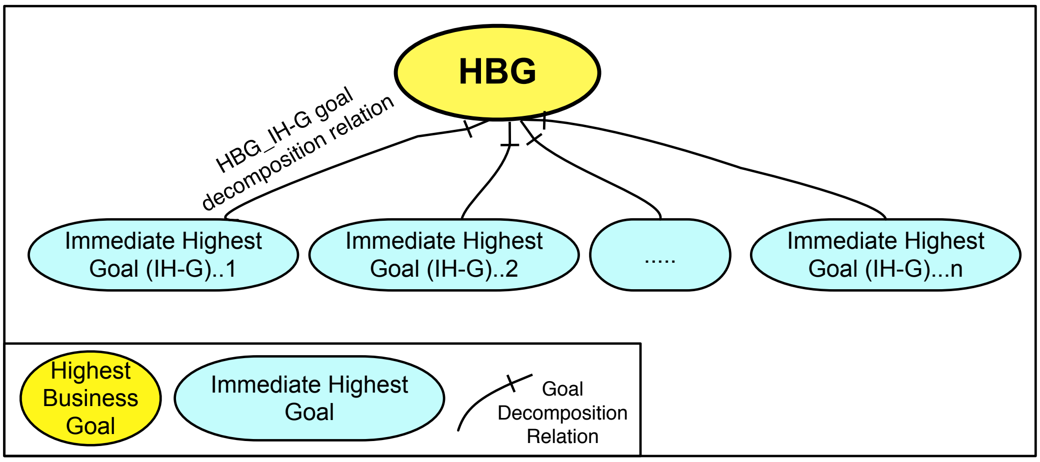


Figure 4.6: Representing The First-Level of the Goal Network in the Business Strategy View.

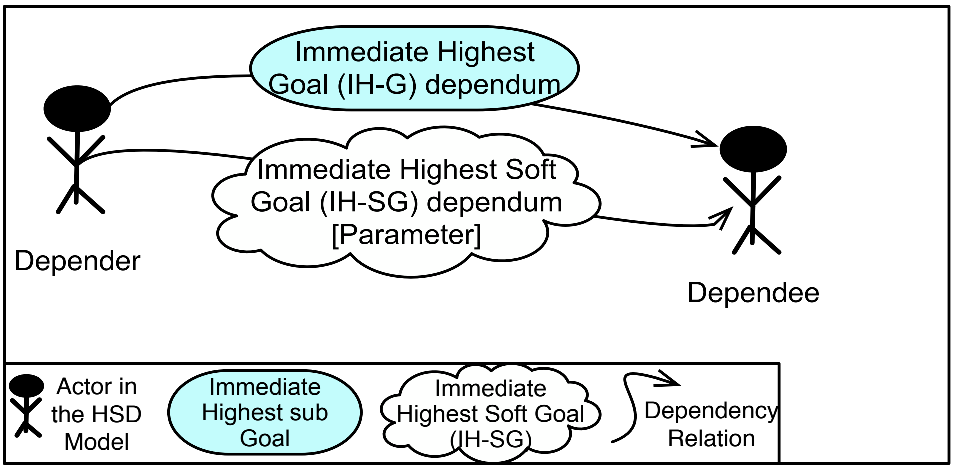


Figure 4.7: The High Strategic Dependency Model in the Refined *i Framework Approach.**

objective for a number of collaborating GBPs that aim to meet the IH-G parent. The IH-Gs are constrained with an associated Immediate Highest Soft Goals (IH-SGs) if needed. The HSD model occupies two different roles in the BSV. The first role is tactical regarding addressing the strategy of the parent BS model considering that the HBG as the end target. The second role is a strategic where the IH-Gs in the HSD model becomes the strategy for further GO models and particularly the *i**framework, which is designed in order to address the IH-Gs.

The notion of the HSD model is inspired from the SD model in the current *i** framework (Yu, 1995). However, in the HSD model; the actors and dependums are higher in abstraction than the actors and the dependums in the SD model (Yu, 1995). Since the HSD model elements are high in abstraction, then the corresponding strategic rationale will be difficult to attain. It is necessary to note that an actor in the HSD model is either a key actor or a main actor for sub actors. The actor in the HSD model is represented using the stick man notation. The IH-G and the IH-SG notations are borrowed from the *i** framework and the NFR framework (Chung et al, 2000). The actor in the HSD must have intentions and/or abilities that overall differ from another actor in this stage.

Figure 4.7 illustrates how the HSD is designed and Figure 4.8 illustrates an example of the HSD from the banking business domain. In Figure 4.7, a directed IH-G dependency link generates from the depender to the dependee, who both are designed in the actor stick notation. An IH-SG is designed in the same way but using the soft goal notation. In the HSD model of the banking example, the client depends on the bank in improving her/his bank account services IH-G. Also, the client depends on the bank for IH-SG of providing security and privacy on her/his account details. In this organisation, the client is a key actor where the bank is a main actor for sub actors (e.g., bank teller, bank manager, etc.).

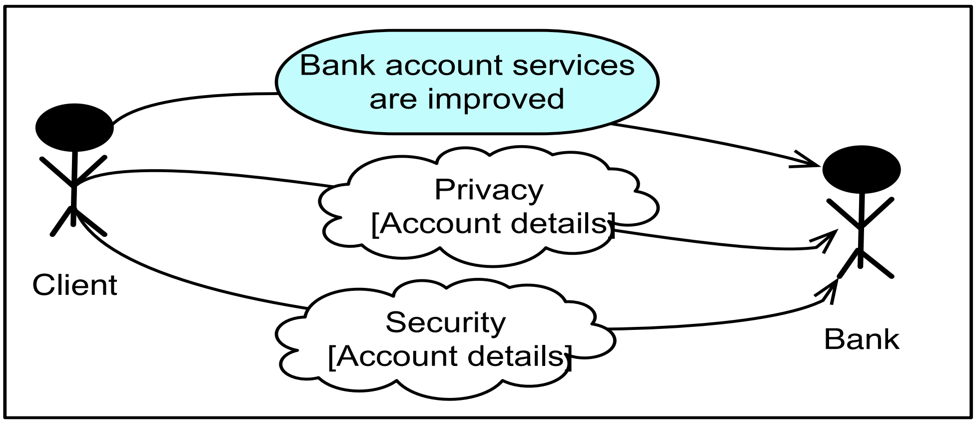


Figure 4.8: The High Strategic Dependency Model for a Banking Organisation

4.3.1.3 The Strategic Dependency Model in the Business Strategy View

In this stage, the *i** framework gradually starts to emerge by elaborating the BS and then the HSD models. This stage aims at looking for the interrelated goal-based BPs that collaborate in order to fulfil an IH-G parent. The GBP is considered as a goal itself in order to adjust the desired alignment with the BPA, as will be discussed in Chapter 5. Therefore, the IH-G is decomposed into a set of GBPs that are collaborating with each other in order to fulfil the IH-G parent. The GBP generated from the decomposition is called the Goal of SD and it is represented using the green oval as depicted in Figure 4.9.

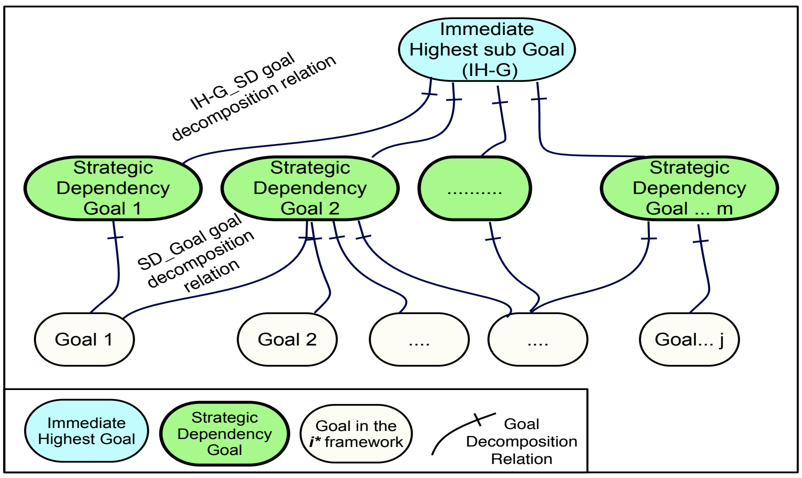


Figure 4.9: Representing the Second- and the Third- Levels in the Goal Network of the Business Strategy View

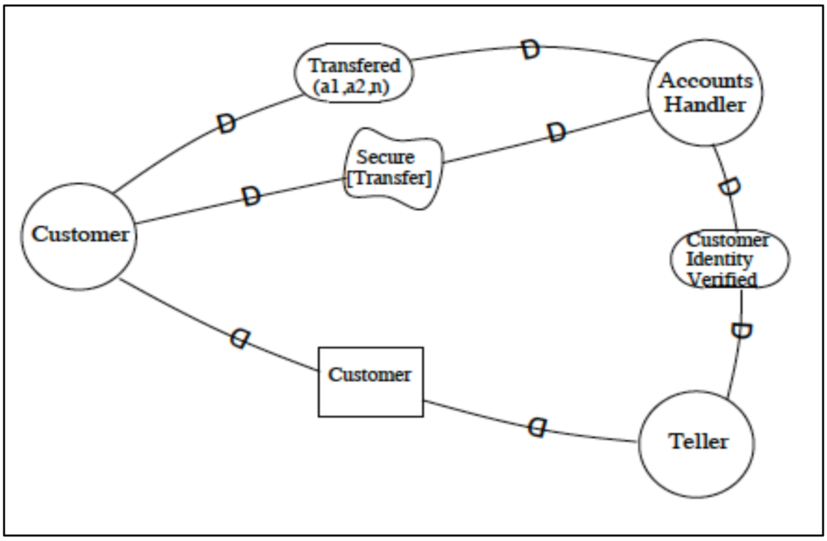


Figure 4.10: The Banking-by-Teller Strategic Dependency Model for a Banking Organisation [Source: (Yu et al, 1995), Used with the author's permission].

Once the Goals of SD are identified for each IH-G, then it is claimed that there is a one-to-one relation between the Goals of SD and SD models (i.e., the goal of SD is the name of the corresponding SD model). That is for each Goal of SD there is a corresponding SD model, where each SD model derives the SR model that is the most refined model in the GO view and in the entire BSV. In particular, it is claimed that one SD model derives at least one SR model. Figure 4.9 depicts the relation between the IH-G, Goal of SD (i.e., SD model) and the goals within the SD and the SR models that represent the actors' intentions and abilities. Finally, it is necessary to note the strategic and the tactical roles of the SD model. The SD model occupies a strategic role with respect to the SR models. The tactical role of the SD model emerges within the design of the SD model in order to address an IH-G as the end target.

In order to follow the example of the banking organisation BSV, the SD model for the banking-by-teller found related and thereby it is adopted from (Yu et al, 1995) shown in Figure 4.10 in order to assist in representing the interrelated GO models in the BSV. Recalling the illustration in Figure 4.9, the banking-by-teller fills the place of the SD goal in green colour, where as the "transferred" goal (i.e., in Figure 4.10) occupies the place as the *i** framework goal. In addition, any goal within the corresponding SR models is also considered as a goal within the *i** framework.

4.3.1.4 The Strategic Rationale Model in the Business Strategy View

The SR model is the second strategic model within the *i** framework (Yu, 1995). It has already been learned from reviewed literature in Chapter 2 (Section 2.2.2.1) that each aforementioned SD model is elaborated into at least one corresponding SR model. With regard to the BSV (in this stage), the SR model integrates with the relevant NFR framework models. In particular, the SR model is the actual bridge between the GO models and the soft goal-oriented models within the BSV as was illustrated in Figure 4.2. In addition, the SR model is the finest goal-oriented model in the proposed BSV. The SR model establishes a strong bridge for the desired future alignment with a BPA, as discussed in Chapter 5. Therefore, the SR model is a critical model in the BSV.

In order to end the representation of the GO models within the BSV for the banking organisation example, the SR model of the corresponding SD model, which was shown in Figure 4.10, is illustrated in Figure 4.11. In Figure 4.11, the bank actor boundary addresses the "profitable bank" soft goal using the *i** framework notations (Yu et al, 1995). However, in the BSV this will be replaced with the NFR framework notations (Chung et al, 2000). Since the SD model must be elaborated into at least one SR model, then another SR model (i.e., banking-by-phone) as an alternative for the banking-by-teller is designed in (Yu et al, 1995) and addresses the same SD model in Figure 4.10.

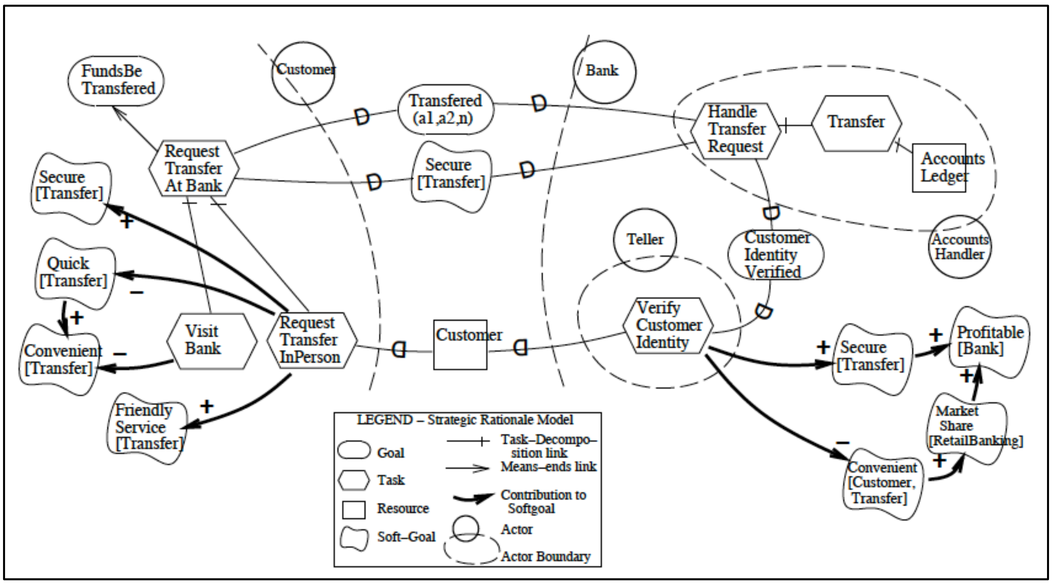


Figure 4.11: The Banking-by-teller Strategic Rationale Model for a Banking Organisation [Source: (Yu et al, 1995), Used with the author's permission].

4.3.1.5 The Non-Functional Requirement Framework in the Business Strategy View

The NFR framework is a standalone soft goal-oriented approach following Chung et al approach discussed in Section 2.2.2.2 (Chung et al, 2000). In this research, the NFR framework models are the SIGs that are independently generated and organised in the form of catalogues that displays the soft goal-oriented models of the organisation for stakeholders. The SIGs are then linked into the GO models of the BSV using the appropriate linking rules. Although there are a lot of successful attempts by researchers regarding the integration of the NFR framework into the *i** framework models (Chung et al, 2000), still their proposed linking is not well-defined through using explicit and structured linking rules. The current integration of the NFR framework into the *i** framework is carried out roughly through an analysis work in (Chung et al, 2000).

Accordingly, structured explicit linking rules have been proposed in this research that connect the NFR framework SIGs with the corresponding SR models. In this research, four structured and explicit linking rules have been identified in order to carry out the integration as illustrated in Figure 4.12.

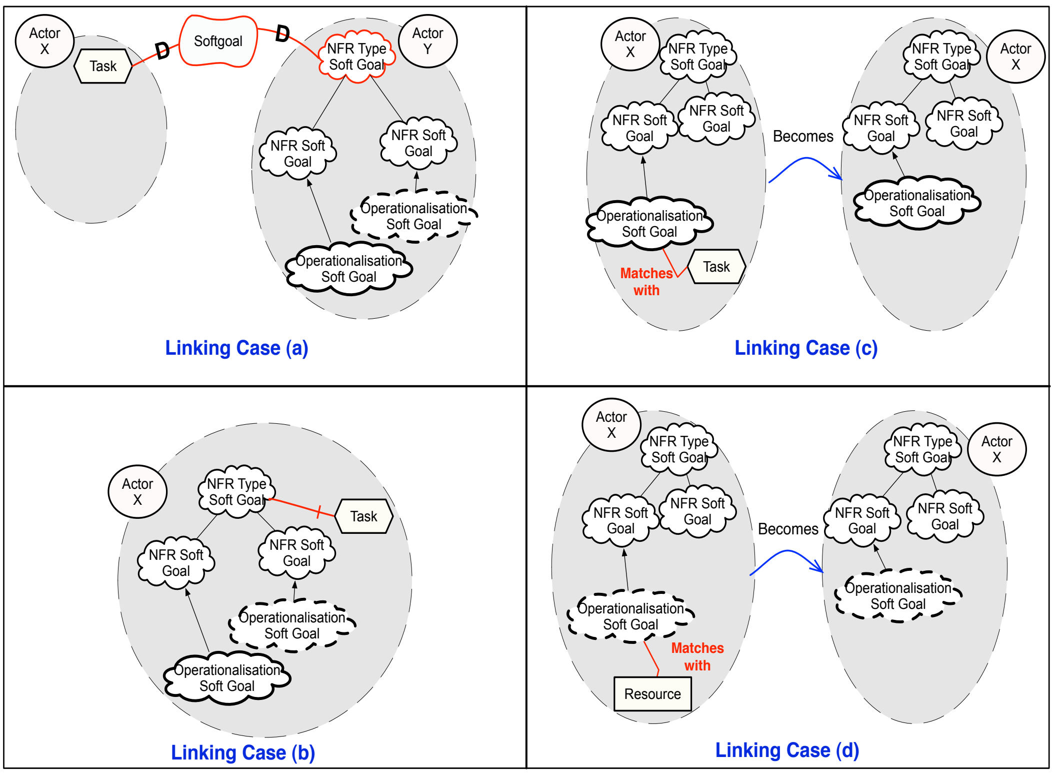


Figure 4.12: Four Linking Cases that integrates the NFR Framework into the *i Framework.**

In Figure 4.12, the illustration in red colour denotes that the analyst is activating the linking. The third and the fourth linking situations are similar with a small difference where both are shown in Figure 4.12 cases (c) and (d), respectively.

1. The first linking rule is activated when the entire NFR framework SIG addresses a soft goal dependum in a dependency relation within the *i** framework. This situation is depicted in the linking case in Figure 4.12 (a). In this kind of situations, it is required to integrate the entire NFR framework and link the root to the soft goal dependency relation. It is apparent that the NFR framework SIG lives within the actor boundary. Therefore, it is required to consider the actor or soft goal holder during the NFR framework design. The current NFR framework approach does not consider this point (Chung et al, 2000); however, few research attempts paid the attention to involve the actors or roles during the design of the SIG (Aburub, 2006) (Cysneiros et al, 2002).

2. The second linking rule is depicted in Figure 4.12 (b). This rule is activated when a main NFR (i.e., main soft goal) is considered as sub part of a task or it is constraining the task that exists in the actor's boundary. In this situation, the entire NFR framework SIG is connected from the task; which exists in the *i** framework to the root of the NFR diagram using the task decomposition relation.

3. The third linking rule shown in Figure 4.12 (c) is activated when a dynamic opreationalisation within a relevant SIG matches exactly with a pre-existing task within the *i** framework. If the analyst decided to keep both of them in the BSV, then this causes a duplicate of one of the BSV elements (i.e., task). In this situation, the analyst must give the priority to the NFR framework representation and thereby omitting the task that belongs to the *i** framework. This omission of the task may be attributed to the necessity of possessing a fully linked *i** framework with its relevant NFR framework graphs. However, the omission of the task does not weaken the representation of the *i** framework, yet it credits the *i** framework in its ability to map its concepts to concepts in another models taking into account the original objective in relation to the early understanding of the organisation's business.

4. The fourth linking rule, shown in Figure 4.12 (d), is similar to the third linking rule, but with respect to the static operationlisation within the NFR framework and with respect to resource instead of the task within the *i** framework.

In the BSV, the NFR framework occupies a strategic role and tactical one as well. The strategic role is manifested in the representation of a strategic soft goal in the BSV that needs the entire representation of its SIG (i.e., using the first and the second linking rules). Its tactical role is manifested when at least one of its leaf soft goal operationalisation matches with an actual task or resource in the SR model (i.e., using the third and the fourth linking rules).

In Figure 4.11, assume that there is a security of accounts soft goal dependency relation sketched from the customer to the bank. And assume that the security of accounts NFR framework SIG exists in the catalogues as shown in Figure 4.13, then this activates the first linking rule within the bank actor boundary in Figure 4.11. The security NFR framework SIG is borrowed from (Chung et al, 2000) and linked it with the banking organisation example in order to illustrate the integration within the BSV.

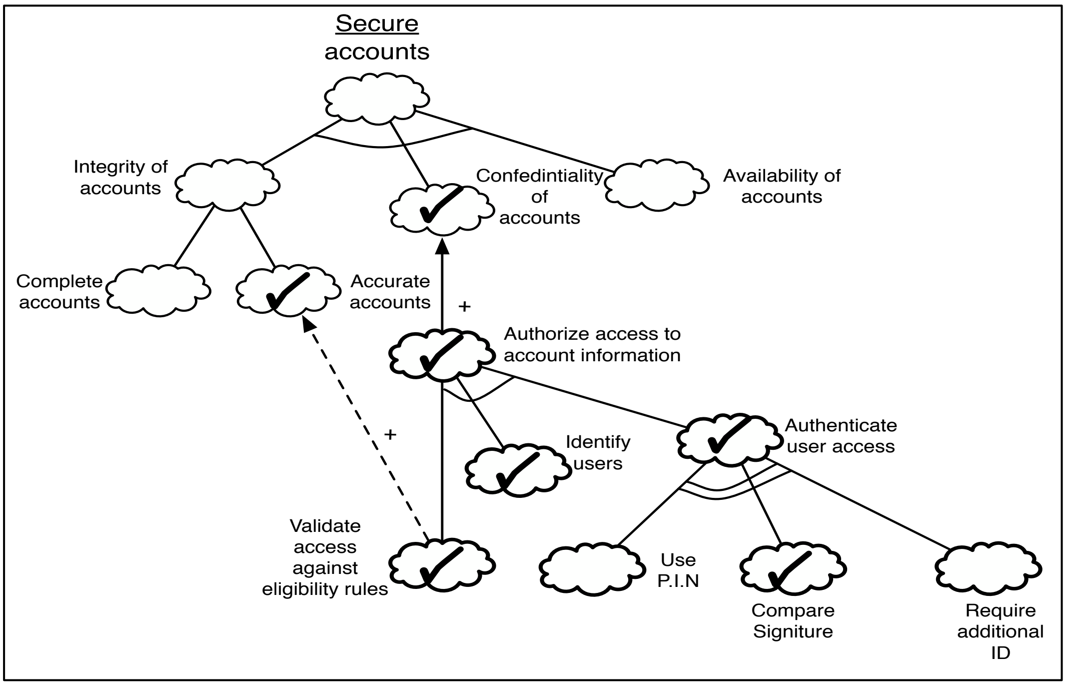


Figure 4.13: Partial Security NFR SIG for Bank Accounts [Source: (Chung et al, 2000), Adapted with the permission of the author and publisher © 2000 by Springer Science+Business Media New York and Kluwer Academic, Non-functional Requirements in Software Engineering, The NFR Framework in Action, 2000, page 38, Lawrence Chung, Brain Nixon, Eric Yu and John Mylopoulos, © 2000 All rights reserved].

4.3.2 The GQOnt Ontology Development Tool and Language

The GQOnt ontology is implemented in OWL-DL using the Protégé ontology development environment (Protégé, 2011). Protégé is an open source tool developed at Stanford Centre for

Biomedical Informatics Research. The BPAOntoSOA framework was developed using Protégé 3.4.1 (Yousef, 2010). Therefore, in this research, it has been chosen not to migrate to Protégé 4.x, which is a later version of Protégé, yet to remain on protégé 3.4.1. This is because the Protégé 4.x does not provide accessibility to the BPAOntoSOA's OWL-DL ontologies that were implemented in Protégé 3.4.1. Moreover, Protégé 4.x does not support full OWL meta-modelling, where a Protégé 3.x does. Consequently, the GQOnt ontology has been developed using OWL-DL Protégé 3.4.1.

Protégé provide the feature of implementing Semantic Web Rule Language (SWRL) rules that are if-then conditional statements. SWRL rules are needed when instances can be generated or linked automatically. Jess engine automates the execution of SWRL rules. Jess engine is not an open source plug-in downloaded separately in order to plug it into Protégé. Jess engine is required, as it assists the user in not only executing SWRL rules quickly, but also in creating clasess and classifiying instances quickly.

4.3.3 The siGoal Ontology Development

The siGoal instantiator component conceptualises the goal-oriented models of the BSV. This component requires the GO models, (i.e., the BS, the HSD and the *i** framework models), in addition to the abstract siGoal ontology as input in order to generate an instantiated siGoal ontology for an organisation as was previously shown in Figure 4.1. In fact, GO models are derived from a set of conducted interviews or from the Goal-based, Quality-linked and Role-driven Business Process Models (GQR-BPMs). For some reason, if the analyst preferred and decided to obtain GQR-BPM-driven GO models, then in this case pseudo code-based algorithms have been introduced to guide how the GO models (i.e., GO view) within the BSV are generated from the GQR-BPMs as will be shown next. In the siGoal ontology, the GO models' main concepts, associated properties and principles permit to perform an automatic consistency check on the generated GO models, as this is a feature in the Protégé environment.

4.3.3.1 Algorithms for Deriving the Strategic View GO Models from the Role-Oriented, Goal-based and Quality-linked (GQR) -BPMs

This section aims at presenting how the GO view for the BSV excluding the soft goal part are systematically derived from the GQR-BPMs of an organisation using pseudo code-based algorithms. Figure 4.14 depicts abstractly the connection between the algorithms where their detailed representations are shown in Figures 4.15-4.26.

Figure 4.14 illustrates the six main algorithms designed in order to obtain the GO models of the BSV. The first algorithm, which is shown in Figures 4.15- 4.16, represents the main algorithm that acts as the parent of the rest algorithms that calls and controls the generation of its three child algorithms. For example, the main algorithm can control the number of the HSD generated models based on the number of the HBGs in the BS model. Each of the child algorithms deals with a particular GO model. The parent algorithm reserves the returned results from each of its children in order to unite them and obtain the required GO view.

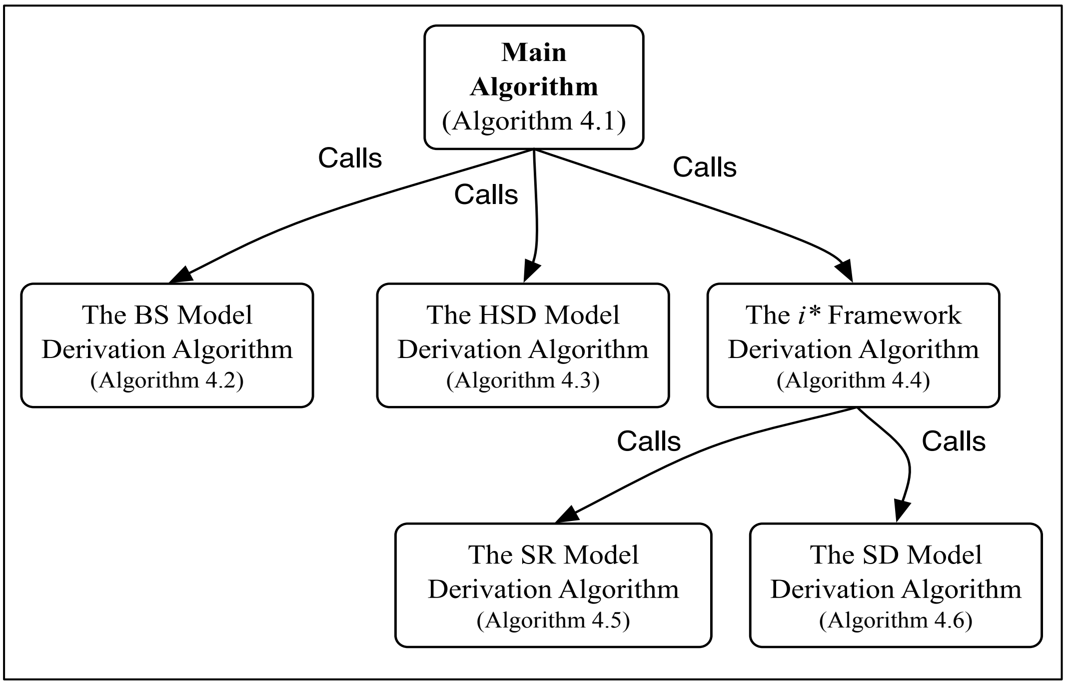
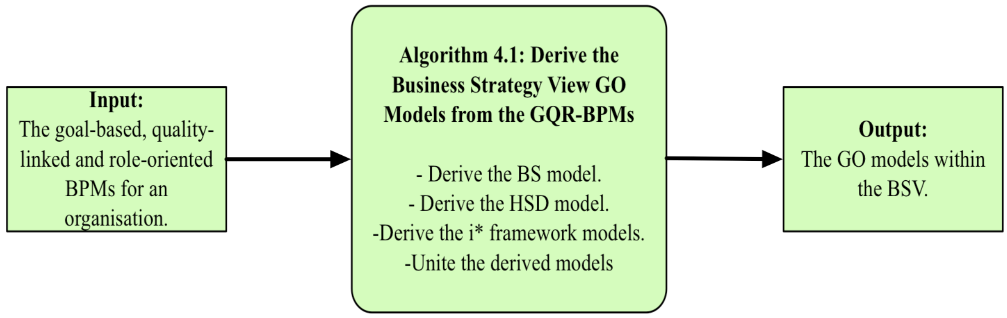


Figure 4.14: The Six Main Algorithms for Deriving the GO Models in the Business Strategy View.



Algorithm 4.1: Deriving the Strategic View Goal-Oriented Models From the GQR-BPMs

Description: This is the main algorithm for generating the goal-oriented models within the BSV for an organisation. The required input is the goal-based, quality-linked and role-oriented BPMs for the organisation. The output is GO models that consist of the business strategy model, the high strategic dependency model and the i* framework models (the SD and the SR models). The algorithm is basically divided into three-child algorithms, where each is concerned with generating a particular GO model.

Input: *The set of main goal-based, quality-linked and role-oriented Business Process Models, GQR-BPM= {gqrbpm$_0$, ...qrgbpm$_n$}.*

Output: *The Business Strategy View's GO Models that comprise of:*

1- *The Business Strategy Model, BS_Model, with all its components: agreed business organisation, ORGANISATION, highest business goals, HBG={hbg$_0$, ..., hbg$_e$} and aims to relation between the agreed organisation and the associated highest business goal(s), Aims_To_REL ={at_rel$_0$, ..., at_rel$_d$}.*

2- *The high strategic dependency model, HSD_Model, with all its components: actors, HSD_Actor={hsd_actor$_0$, ...,,hsd_actor$_u$} immediate highest sub goals IH-G= {ihg$_0$, ..., ihg$_v$}, immediate highest soft goals, IH-SG={ihsg0, ..., ihsg$_t$}, the immediate highest sub goal and soft goal dependency relations between the actors, IHG_DREL={ihg_drel$_0$, ..., ihg_drel$_p$} and IHSG_DREL={ihsg_drel$_0$, ..., ihsg_drel$_f$}.*

3- *The strategic dependency models, SD_model={sdm$_0$, ..., sdm$_r$}with all its components: actors, Actor={actor$_0$, ..., actor$_x$}, goal dependums, Goal_Dependum= {gd$_0$, ..., gd$_o$}, soft goal dependums, Soft_Goal_Dependum= {sgd$_0$, ..., sgd$_y$}, goal dependency relations between Actors, Goal__Dependecy_REL={gd_rel$_0$, ..., gd_rel$_j$}, and soft goal dependency relations between Actors, Soft_Goal__Dependency_REL={sgd_rel$_0$, ..., sgd_rel$_w$}.*

Figure 4.15: The Main Algorithm for Deriving the GO Models that exist within the Business Strategy View from the GQR-BPMs: Part 1 of 2.

4- *Strategic rationale models that contains (3), SR_model={srm_0, ..., srm_s}, actor boundary, Actor_Boundary={ bdr_0, ..., bdr_l}, goals within an actor boundary, Goal={g_0, ..., g_i}, soft goals within an actor boundary, Soft_Goal={sg_0, ..., sg_k}, tasks within an actor boundary, Task={$task_0$, ..., $task_c$}, resources within an actor boundary, Resource={$resource_0$, ..., $resource_g$}, task decomposition relations between task and task, Task_Decomposition_subTask={tdt_rel_0, ..., tdt_rel_b}, task decomposition relations between task and goal, Task_Decomposition_subGoal={tdg_rel_0, ..., tdg_rel_a}, task decomposition relations between task and resource, Task_Decomposition_ResourceFor={tdr_rel_0, ..., tdr_rel_h}, task decomposition relations between task and soft goal, Task_Decomposition_SoftGoal={$tdsg\_rel_0$, ..., $tdsg\_rel_q$}, relations between a mean task and an end goal, GT_MeanEnd={mgt_rel_0, ..., mgt_rel_m} and finally, relations between a mean task and an end task, TT_MeanEnd={mtt_rel_0, ..., mtt_rel_{a1}},*

Begin
Let org be an instance of ORGANISATION.
st_view_go is the goal-oriented view within the strategic view for org;

Identity the set of all goal-based, quality-linked and role-oriented main business process models, GQR-BPM={$gqrbpm_0$, $gqrbpm_1$, ..., $gqrbpm_i$..., $gqrbpm_n$}, $0 \le i \le n$;
// Deriving the BS Model
bs is an instance of the BS_Model class;
bs = Derive_BS_Model (GQR-BPM);// Calls the algorithm 4.2.

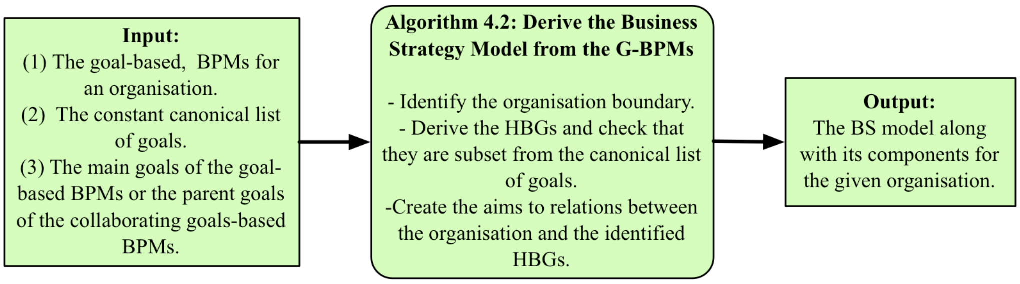
//Deriving the HSD model
max= number of returned HBGs in bs.
Counter= 0.
Create a (max) number of HSD instances.
Let the set HSD_Model be the set of required high strategic dependency models in order to address a particular hbg, *HSD_Model={$hsdm_0$, $hsdm_1$, ..., $hsdm_{q1}$, ..., $hsdm_{q2}$}*, $0 \le q1 \le q2$.
For each hbg in bs
    If (counter<= max)

    $hsd_{counter}$= Derive_HSD_Model(GQR-BPM); // Calls the algorithm 4.3
    counter= counter+1;
    else break for each hbg loop;
    End if else
End for each hbg
Identify the set of all immediate highest sub goals in hsd_{q1}, *IH-G= {ihg_0, ihg_1,..., ihg_{v1}, ...,ihg_v }, $0 \le v_1 \le v$;*

//Deriving the i* framework models
Let i^*_{new}=null, i^*_{old}= i^*_{new};
If the i* framework models are absent
    iList is a list in order to insert a set of GQR-BPMs;
    For each ihg_{v1} in the HSD_Model
        i^*_{old}= i^*_{new};
        Find the set of GQR-BPMs that address the ihg_{v1};
        Add the founded GQR-BPMs to the iList;
        Let i^*_{in}=Derive *i*Framework (iList); //Calls algorithm 4.4*
    Let i^*_{new}= i^*_{in} U i^*_{old};
    End for each ihg_{v1}
End if
// Deriving the GO view within the BSV.
st_view_go= bs U hsd U i^*_{new};

Figure 4.16: The Main Algorithm for Deriving GO Models that exist within the Business Strategy View from the GQR-BPMs: Part 2 of 2.

The first child algorithm presented in Figures 4.17 and 4.18 handles the derivation of the BS model from the G-BPMs for an individual organisation. In this algorithm, the first GO model is obtained for the BSV and returned to its main algorithm. The required input to this algorithm is the G-BPMs, the constant canonical list of goals proposed in (Clements and Bass, 2010) and the ultimate goals of the G-BPMs or the parent goals for the collaborating G-BPMs. The output is the BS model for the given G-BPMs of an organisation.



Algorithm 4.2: Derive_BS_Model From the G-BPMs.

Description: This algorithm is the first child algorithm that is concerned with generating the business strategy model of a given business organisation from its goal-based business process models.

***Input*:** (1) The constant canonical list of goals, CLG={growth and continuity of organisation, meeting financial objectives, meeting personal objectives, meeting responsibility to employees, meeting responsibility to society, meeting responsibility to country, meeting responsibility to shareholders, managing market position, improving business processes and managing quality and reputation of products}. (2) *The set of main role-oriented and goal-based Business Process Models, GBPM= {$gbpm_0$, ...$gbpm_n$} with their associated main goals Main_G= {mg_0, ..., mg_v} or parent goals of each set of the collaborating GBPMS, PG={pg_0, ..., pg_h}.*

Output*: The Business Strategy Model, with all its components: agreed business organisation, ORGANISATION, highest business goals, HBG={hbg_0, ..., hbg_e} and aims to relation between the agreed organisation and the associated highest business goal(s), Aims_To_REL ={at_rel_0, ..., at_rel_d}.*

Figure 4.17: Algorithm 4.2, Derive the BS Model from the G-BPMs: Part 1 of 2.

Begin

Let org be the name of the given organisation, ORGANISATION;

Let bs_model be an instance of the BS_Model;

Add org to bs_model;

If the main goals are identified for the given G-BPMs

Let the Main_G be the set of the identified optimal goals for the given G-BPMs, Main_G= $\{mg_0, ..., mg_u, ..., mg_v\}$. $0 \leq u \leq v$;

If at least one of the $optg_u$ does not match with one element in the CLG then //checking

Re-elicit the optimal goals in Main_G and start the algorithm again.

Else // if they matched with CLG.

<u>Label: draw</u>

Identify the set of the highest business goals of the given organisation, $HBG=\{hbg_0, ..., hbg_r, ..., hbg_e\}$, $0 \leq r \leq e$;.

Identify the set of the aims-to relations Aims_To_REL=$\{at\_rel_0, ..., at\_rel_a, ..., at\_rel_d\}$, $0 \leq a \leq d$.

For each mg_u in OPT_G for the G-BPMs do the following

Allocate a hbg_r for a corresponding mg_u.

Create aims-to relation, at_rel_a, from the source org to the destination hbg_r.

Add hbg_r and at_rel_a to bs_model;

End for each optimal goal

Exit algorithm;

End if else //check at least one $optg_u$ matches with CLG.

Else // if optg is not identified

Label parent: Identify the set of the parent goals for each set of collaborating GBPMs, , $PG=\{pg_0, ..., pg_w, ..., pg_h\}$, $0 \leq w \leq h$.

Combine at least two different pg_w in PG;

Assume that the resulted goals from the combination are the highest business goals or the given org, $HBG=\{hbg_0, ..., hbg_r, ..., hbg_e\}$, $0 \leq r \leq e$.

If at least one of the hbg_r does not match with one element in the CLG then //checking

Go to label: parent.

Else Go to label: draw.

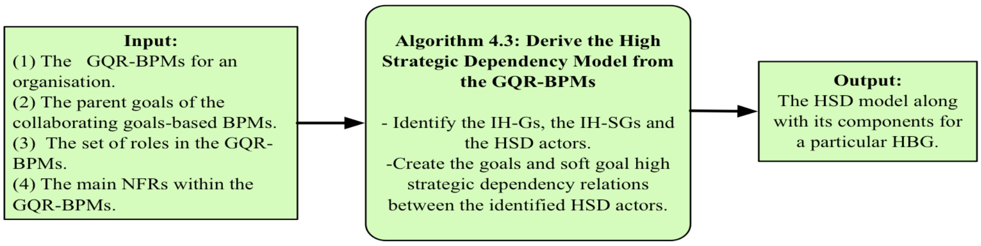
End if else at least one hbg_r

End if else

End

Figure 4.18: Algorithm 4.2, Derive a BS Model from the G-BPMs: Part 2 of 2.

Figures 4.19 and 4.20 show the second child algorithm that is designed for deriving the HSD model through elaborating each HBG returned to the main algorithm. In particular, for each HBG there is a corresponding HSD model that will gradually assist in generating the *i** framework models. In this algorithm, the input is the GQR-BPMs along with their roles and main NFRs. The output is the HSD model with its components.



Algorithm 4.3: Derive_HSD_Model From the GQR-BPMs

Description: This algorithm is the second child algorithm that is concerned with generating the high strategic dependency model as the first dependency model in the strategic view of a given business organisation from its goal-based business process models.

Input: (1) *The set of main goal-based, quality-linked and role-oriented Business Process Models, GQR-BPM= {*$gqrbpm_0$*, ...*$gqrbpm_n$*} with their associated parent goals (if exist) of each set of the collaborating GQR-BPMS, PG={*pg_0*, ...,* pg_h*} and roles, R={*r_0*, ...,* r_y*}*.(2) The main NFRs of GBPMs, NFR_Type={nfr_type_0, …, nfr_type_c}

Output: *The high strategic dependency model, with all its components: actors, HSD_Actor={*hsd_actor_0*, ...,*hsd_actor_u*} immediate highest sub goals IH-G= {*ihg_0*, ...,* ihg_v*}, immediate highest soft goals, IH-SG={*$ihsg_0$*, ...,* $ihsg_t$*}, the immediate highest sub goal and soft goal dependency relations between the actors, IHG_DREL={*ihg_drel_0*, ...,* ihg_drel_z*} and IHSG_DREL={*$ihsg\_drel_0$*, ...,* $ihsg\_drel_e$*}*

Begin

Identify the set of the goal-based, quality-linked and role-oriented business process models, GQR-BPMs=*{*$gqrbpm_0$*, ...,*$gqrbpm_q$*, ...,* $gqrbpm_n$*}*. $0 \leq q \leq n$.

For each $gqrbpm_q$ *do the following*

Identify the set of the NFRs that constrain the $gqrbpm_q$*, NFR_Type={*nfr_type_0*, ...,* nfr_type_b*, ...,* nfr_type_c*}* $0 \leq b \leq c$.

Identify the set of roles within the $gqrbpm_q$*, R={*r_0*, ...,* r_j*, ...,* r_y*}* $0 \leq j \leq y$

End for each gqrbpm

Figure 4.19: Algorithm 4.3, Derive a HSD Model from GQR-BPMs: Part 1 of 2.

If the parent goal(s) for the given GQR-BPMs are already identified (exist) then
Replace/Rename the identified parent goals, PG={pg_0, …, pg_f, …, pg_h}, $0 \leq f \leq h$, with the set of the immediate highest sub goals, IH-G={ihg_0, …, ihg_p, …, ihg_v} $0 \leq p \leq v$;
Else
Identify the parent goals for each set of collaborating GQR-BPMs, PG={pg_0, …, pg_f, …, pg_h}, $0 \leq f \leq h$;
Rename the PG with the set of the immediate highest sub goals, IH-G={ihg_0, …, ihg_p, …, ihg_v} $0 \leq p \leq v$;
End if else
Make the given NFR_Type as the immediate highest soft goals, IH-SG={$ihsg_0$, …, $ihsg_x$, …, $ihsg_t$} $0 \leq x \leq t$;.
Identify the set of actors in the HSD model, HSD_Actor={ hsd_actor_0, …, hsd_actor_i, …, hsd_actor_u }, $0 \leq i \leq u$;.

For each role, r_j, in GQR-BPM do the following
If r_j is a key actor then
Add r_j to HSD_Actor;
Else
Combine the rj with another role in R that both address a particular IH-G or IH-SG;
Add the resulted combination of roles as one actor to HSD_Actor;
End if else
End for each role

For each identified actor, hsd_actor_i,do the following
Create intention_list of hsd_actor_i;
Create ability_list of hsd_actor_i;
For each ihg_p in IH-G do the following
If ihg_p is an intention for hsd_actor_i,
Add ihg_p into the intention_list of hsd_actor_i,
Else If ihg_p is an ability for hsd_actor_i
Add ihg_p into the ability_list of hsd_actor_i,
End for each ihg

For each $ihsg_x$ in IH-SG do the following
If $ihsg_x$ is an intention for hsd_actor_i,
Add $ihsg_x$ into the intention_list of hsd_actor_i,
Else If $ihsg_x$ is an ability for hsd_actor_i
Add $ihsg_x$ into the ability_list of hsd_actor_i,
End for each ihsg
End for each HSD actor

Identify the set of the immediate highest sub goal dependency relation, IHG_DREL={ihg_drel_0, …, ihg_drel_w, …., ihg_drel_z}, $0 \leq w \leq z$

Identify the set of the immediate highest soft goal dependency relation, IHSG_DREL={$ihsg\_drel_0$, …, $ihsg\_drel_s$, …., $ihsg\_drel_e$}, $0 \leq s \leq e$

For each identified actor, hsd_actor_i,do the following
For each intention in the intention list of hsd_actor_i,do the following
Find an actor in HSD_ACTOR that address the intention in its ability_list;
If the intention is goal then
Create a ihg_drel_w relation from , hsd_actor_i, to the actor found
Add ihg_drel_w to IHG_DREL;
Else
Create a $ihsg\_drel_s$ relation from , hsd_actor_i, to the actor found
Add $ihsg\_drel_s$ to IHSG_DREL;
End for each intention
End for each HSD actor
Add the HSD_Actor, IHG_DREL, IHSG_DREL, IH-G and IH-SG into the hsd_model.
End

Figure 4.20: Algorithm 4.3, Derive a HSD Model from GQR-BPMs: Part 2 of 2.

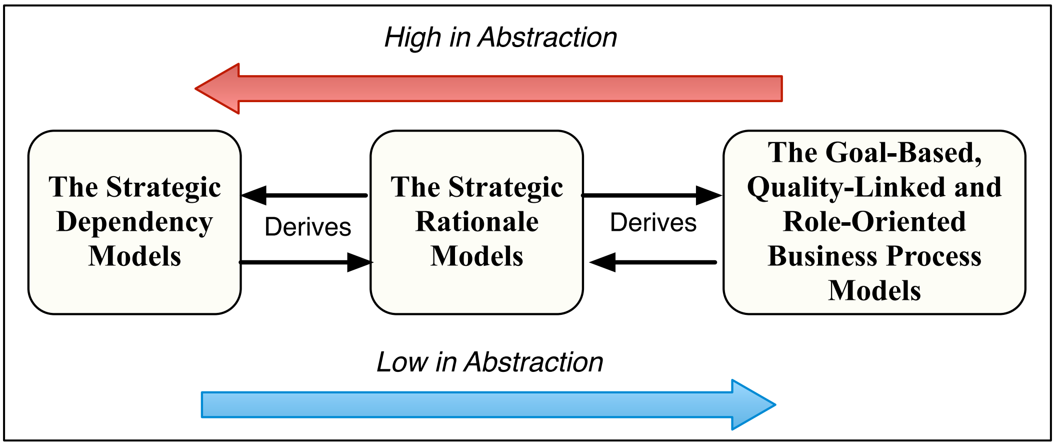
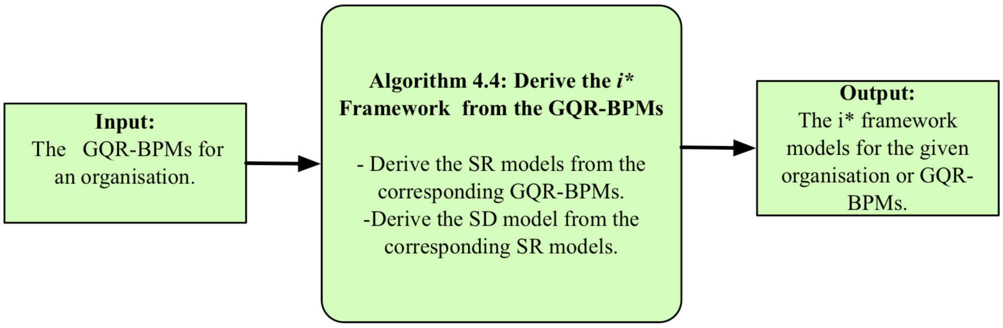


Figure 4.21: The Abstraction Relation between the SD, the SR and the GQR-BP Models.

The third child algorithm is for generating the *i** framework strategic models (i.e., the SD and the SR models) as was illustrated in Figure 4.14. The generation sequence of the *i** framework models starts with the SR model and then the SD model afterwards. This sequence is used in deriving the *i** framework due to the relevant abstraction level between the GQR-BPMs and the corresponding SR models as shown in Figure 4.21. In particular, the GQR-BPM model and the SR model are fine-grained models; and therefore they possess some relevant abstraction level. However, this does not assert that deriving an SD model from GQR-BPM is unattainable. The SR and the SD models set up the *i** framework; thereby, they are already relevant in this level of abstraction taking into account that the SR model is finer than the SD model as shown in Figure 4.21.

The algorithms related to the derivation of the *i** framework are shown from Figures 4.22 - 4.26. In the body of the *i** framework algorithm (Algorithm 4.4, that is shown in Figure 4.22 calls the *i** framework models). In this algorithm, the SR model is firstly derived from a corresponding GQR-BPM in the form of a one-to-one relation. The SD model will afterward result from the derived SR model. The algorithm uses the goal-based, role-oriented and quality linked BPMs as the input requirement. Each of these BPMs generates one SR model and one SD model as well. It is potential to derive several SR models from their corresponding GQR-BPMs for one SD model. This is because a one SD model generates at least one SR model, which elaborates the SD model (i.e., the strategy of the SR model(s)) with several tactics if needed.

Accordingly, it is deduced that a one SD model is a common target for several SR models corresponding to related GQR-BPMs.



Algorithm 4.4: Deriving the *i** Framework from a Set of GQR-BPMs

Description: *This is the third child algorithm that is designed to generate the i* framework models involved in the proposed strategic view's GO models. The algorithm invokes its sub algorithms that generate the SR and then the SD models respectively from the relevant GQR-BPMs.*

Input: *The set of main role-oriented and goal-based Business Process Models, GQR-BPM= {$gqrbpm_0$, ...$gqrbpm_n$}.*

Output: *The i* Framework Goal-Oriented Models, with all its elements of both strategic models:*

(1) *Strategic dependency models, SD_model={sdm_0, ..., sdm_r}, actors, Actor={$actor_0$, ..., $actor_x$}, goal dependums, Goal_Dependum= {gd_0, ..., gd_o}, soft goal dependums, Soft_Goal_Dependum= {sgd_0, ..., sgd_y}, goal dependency relations between Actors, Goal__Dependecy_REL={GD_rel_0, ..., GD_rel_j}, and soft goal dependency relations between Actors, Soft_Goal__Dependency_REL={SGD_rel_0, ..., SGD_rel_w}.*

(2) *Strategic rationale models that contains (1), SR_model={srm_0, ..., srm_s}, actor boundary, Actor_Boundary={ bdr_0, ..., bdr_l}, goals within an actor boundary, Goal={g_0, ..., g_z}, soft goals within an actor boundary, Soft_Goal={sg_0, ..., sg_k}, tasks within an actor boundary, Task={$task_0$, ..., $task_c$}, resources within an actor boundary, Resource={$resource_0$, ..., $resource_g$}, task decomposition relations between task and task, Task_Decomposition_subTask={tdt_rel_0, ..., tdt_rel_u}, task decomposition relations between task and goal, Task_Decomposition_subGoal={tdg_rel_0, ..., tdg_rel_a}, task decomposition relations between task and resource, Task_Decomposition_ResourceFor={tdr_rel_0, ..., tdr_rel_d}, task decomposition relations between task and soft goal, Task_Decomposition_SoftGoal={$tdsg\_rel_0$, ..., $tdsg\_rel_v$}, relations between a mean task and an end goal, GT_MeanEnd={mgt_rel_0, ..., mgt_rel_m}, relations between a mean task and an end task, TT_MeanEnd={mtt_rel_0, ..., mtt_rel_{a1}},*

Begin

Identity the set of all goal-based, quality-linked and role-oriented main business process models, GQR-BPM={$gqrbpm_0$, $gqrbpm_1$, ..., $gqrbpm_i$..., $gqrbpm_n$}, $0\leq i\leq n$;

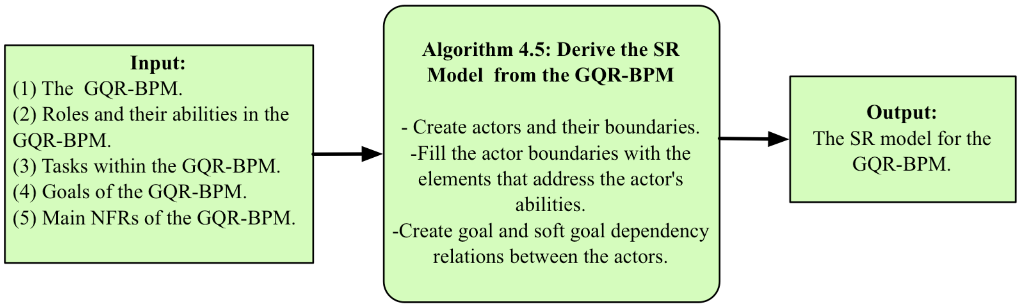
Identify the corresponding set of strategic rationale models of the identified GQR-BPMs, SR_Model={srm_0, srm_1, ... srm_b, ..., srm_s}, $0\leq b\leq s$;

Identify the corresponding set of strategic dependency models of the identified SR_Model, SD_Model={sdm_0, sdm_1, ... sdm_p, ..., sdm_r}, $0\leq p\leq r$;

For each goal-based business process model, $gqrbpm_i$ in GQR-BPM do the following
  srm_b = Derive_SR_Model ($gqrbpm_i$); //Calls algorithm 4.5
  sdm_p = Derive_SD_Model (srm_b); //Calls algorithm 4.6
End for each goal-based, quality-linked and role-oriented business process model;
End

Figure 4.22: Algorithm 4.4, Deriving the *i Framework from GQR-BPMs**

Algorithm 4.5, which is shown in Figures 4.23 - 4.25, is concerned with generating one SR model from one goal-based, role-driven and quality linked BPM. Therefore, the required input is the GQR-BPM and the output is the corresponding SR model of this GQR-BPM. It is necessary to recall that the BSV is interested in designing the goal and the soft goal dependencies.



Algorithm 4.5: Derive the SR Model

Description: This algorithm is designed in order to generate the SR model from a role-oriented, goal-based and quality linked BPM. The algorithm works first by generating the actor boundaries separately along with elements and relations. Then the algorithm works to generate the dependency relations between the actor boundaries in order to link all the actors together.

Input: *A goal-based, quality-linked and role-oriented business process model,* $gqrbpm_{i,}$ *where it is a collection of roles,* $R = \{r_0, \ldots, r_m\}$ *consisting of a number of tasks* $T = \{t_0, \ldots, t_{l'}\}$ *and non/behavioural abilities or responsibilities,* $Resp=\{resp_0, \ldots, resp_y\}$*, in order to fulfil BPM's goals,* $GBP=\{gbp_0, \ldots, gbp_{f'}\}$.

Output: *A SR model,* srm_b*, within it elements and the relationships between the elements, as was mentioned in the second output in Algorithm 4.4.*

Begin

G_List *is a global variable of goals and soft goals (main NFRs) in SR and associated SD models of the* $gqrbpm_i$*;*
Identify the set of goals of the $gqrbpm_{i,}$*,* $GBP=\{gbp_0, gbp_1, \ldots, gbp_f, \ldots, gbp_{f'}\}$, $0 \leq f \leq f'$;
Add all goals and soft goals in GBP of $gqrbpm_i$ *in the G_List;*
Name the srm_b *with the name of the* $gbpm_i$*;*
//Make G_List as the list of goals that must exist in the $srm_{b;}$

Identity the set of roles in the business process model, $gqrbpm_i$*;* $R=\{ r_0, r_1, \ldots, r_j, \ldots, r_m\}$, $0 \leq j \leq m$;
Identify the R's corresponding set of actors for the srm_b*;* $Actor=\{actor_0, actor_1, \ldots, actor_a, \ldots, actor_x\}$, $0 \leq a \leq x$;
Identify the boundary for each identified actor, $Actor\_Boundary=\{ bdr_0, bdr_1, \ldots, bdr_u, \ldots, bdr_l\}$, $0 \leq u \leq l$;

Figure 4.23: Algorithm 4.5, Derive the SR Model from GQR-BPM (Part 1 of 3).

[redacted]	Actor abilities	Actor intentions
Goals		
Soft goals		

For each role r_j in R do the following

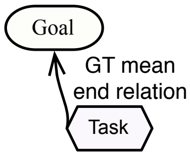
Create a corresponding $actor_a$ of r_j;
Name $actor_a$ with the r_j role name;
Create a corresponding boundary, bdr_u, for $actor_a$;

Identify the role's r_j set of behavioural and non-behavioural responsibilities or abilities, Ability =$\{ability_0, ability_1, \ldots, ability_d, \ldots, ability_y\}$, $0\leq d\leq y$;
*Let **Ability_List** be a variable that stores the goal and soft goal abilities of $actor_a$;*
Turn the identified behavioural and non-behavioural abilities into goals and soft goals;
Add the identified ability goals and soft goals in the Ability_List for $actor_a$;

*Let **Intention_List** be a variable that stores the goal and soft goal intentions of $actor_a$;*
Elicit $actor_a$'s intentions from relevant business documents of $gqrbpm_i$, Intention=$\{in_0, in_1, \ldots, in_e, \ldots, in_{e'}\}$,$0\leq e\leq e'$;
Turn the elicited behavioural and non-behavioural intentions of $actor_a$ into goals and soft goals;
Add the elicited non/behavioural intention goals in the Intention_List for $actor_a$;

Identity the set of tasks in r_j, T=$\{t_0, t_1, \ldots, t_k, \ldots, t_l\}$, $0\leq k\leq l'$;
*Let **Task_List** be the list of all tasks for an actor.*
Identify the corresponding set of tasks for T in $actor_a$, Task=$\{task_0, task_1, \ldots, task_t, \ldots, task_c\}$, $0\leq t\leq c$, and add them into Task_List of $actor_a$;
// Tasks for the SR model where T is for the GQR-BPM.
Add all tasks in Task_List into bdr_u of $actor_a$;

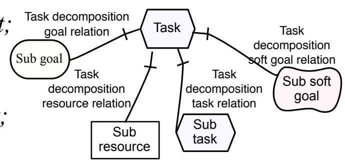
While (Ability_List is not empty **AND** *the G_list is not empty* **AND** *Intention_List is not empty)*
*For each **goal** $ability_d$ of $actor_a$*
If there is at least one $task_t$ in Task of $actor_a$ that fufills $ability_d$ of type goal
Add $task_t$ into bdr_u of $actor_a$;
Else
Create a $task_t$ that fulfil $ability_d$;
Add $task_t$ to Task list of $actor_a$ and bdr_u;
End if else



Identify the set of goal-end with task-mean relations, GT_MeanEnd=$\{mgt\_rel_0, mgt\_rel_1, \ldots, mgt\_rel_w, \ldots, mgt\_rel_m\}$,$0\leq w\leq m$;
Add goal $ability_d$ into bdr_u of $actor_a$;
Create GT_meanend relation, mgt_rel_w, from $task_t$ to $ability_d$
Add mgt_rel_w into the bdr_u of $actor_a$;

For each decomposable $task_t$ in bdr_u
Classify the mean $task_t$ "parent task" into sub tasks, sub resources, and if possible to sub ability, that is goal or soft goal in Ability of $actor_a$,

//Create task decomposition relation
Add the classified entities as a sub part of the identified parent task in bdr_u;
Delete the added elements from Task.
Add a task decomposition relation from parent task to its sub part;

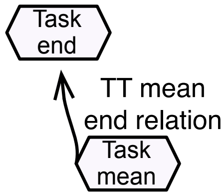


***If** the sub part is task*
Add a tdt_rel in bdr_u from parent task to the task sub part;
Else if the sub part is resource
Add a tdr_rel in bdr_u from parent task to the resource sub part;
Else if the sub part is soft goal
Add a tds_rel in bdr_u from parent task to the soft goal sub part;
Delete this soft goal from the Ability_List and G_List;//it may be intention
Else if the sub part is goal
Add a tdg_rel in bdr_u from parent task to the goal sub part;
//here the goal may be ability or intention
End if sub part
***End for each** decomposable task*

Figure 4.24: Algorithm 4.5, Derive SR Model from GQR-BPM (Part 2 of 3).

Identify the set of end task with mean task relations, TT_MeanEnd={mtt_rel_0, mtt_rel_1..., mtt_rel_{a1}, ..., mtt_rel_{a2}}, $0 \le a1 \le a2$;
Find the bdr_u all mean tasks in Task_List that address an end $task_t$;
Create TT_meanend relation, mtt_rel_{a1}, from each mean task to $task_t$;
Add all mtt_rel_{a1} relations into the bdr_u of $actor_a$;

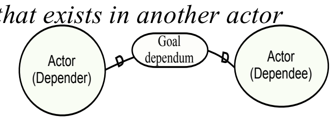
***If goal** $ability_d$ matches with a goal in G_list then*
Delete goal $ability_d$ from Ability_List of $actor_a$;
Delete matched goal in G_list with $ability_d$;
***End if goal** $ability_d$*



*End for each **goal** ability*
End while

End for each role

// Now all actors are created with their boundaries that are filled with the SR model elements.
//However, the actors are still not connected with the dependency relations.
//Work on goal dependency relations
Identify the set of the goal dependency relations, Goal__Dependecy_REL={GD_rel_0, GD_rel_1, ..., GD_rel_{n1}, ..., GD_rel_{n2}}, $0 \le n1 \le n2$;
***For each leaf goal**, in_e,(intention) in bdr_u of $actor_a$, in Actor do the following*
If the in_e goal is achieved through depending on a $task_t$ that exists in another actor bdr then

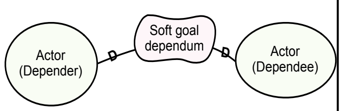


Create a goal dependency relation GD_rel_{n1};
Add in_e goal dependum in GD_rel_{n1};
Make the GD_rel_{n1} source $actor_a$ as depender;
Make the GD_rel_{n1} destination $actor_a$ as dependee;

***If** intention in_e, match with a goal in G_list then*
Delete in_e from Intention_List of $actor_a$ and from G_List;
***End if** intention*
End if
End for each leaf goal

//Work on soft goal dependency relations

Identify the set of the soft goal dependency relations, Soft_Goal__Dependecy_REL={SGD_rel_0, SGD_rel_1, ..., SGD_rel_{w1}, ..., SGD_rel_{w2}}, $0 \le w1 \le w2$;
For each leaf soft goal, in_e,(intention) in bdr_u of $actor_a$, in Actor do the following
If the in_e soft goal is achieved through depending on a NFR or a task that exists in another actor bdr then
Create a soft goal dependency relation SGD_rel_{w1};
Add in_e soft as goal dependum in SGD_rel_{w1};
Make the SGD_rel_{w1} source $actor_a$ as depender;
Make the SGD_rel_{w1} destination $actor_a$ as dependee;
***If** intention in_e, matches with a soft goal in G_list then*
Delete in_e from Intention_List of $actor_a$ and from G_List;
***End if** intention*
End if



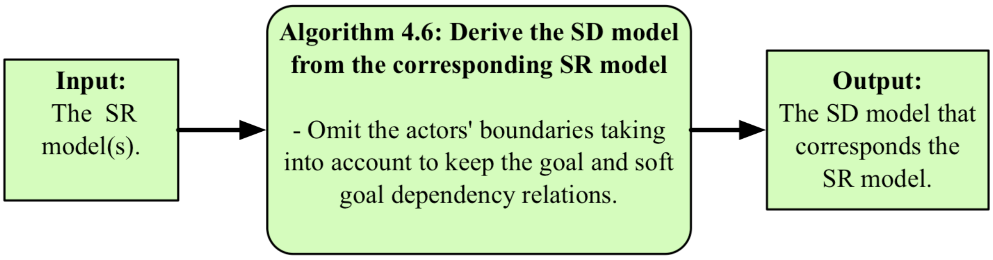
End for each leaf soft goal

// If G_List is not empty then Error→ Gap analysis.

Figure 4.25: Algorithm 4.5: Derive the SR Model from the GQR-BPM (Part 3 of 3).

Algorithm 4.6 does only delete the unrequired parts of the SR model in order to have the SD model that are the actors' boundaries along with their internal elements and the relationships between them, yet remaining the actors, the goal dependencies and the soft goal dependencies.

4.3.3.2 The siGoal Ontology Design Decisions



Algorithm 4.6: Deriving the SD Model from a Corresponding SR Model

Description: The algorithm derives the SD model from the corresponding SR model, which is generated from a goal-based, quality-linked and role-oriented BPM using the Algorithm 4.5.

Input: *An SR model within its elements and the relationships between the elements.*

Output: *The SD model with its elements and the relationships between them, as was previously mentioned in the first output of algorithm 4.4.*

Begin

Dependency_Rel is a global variable that reserves a list of the identified goal and soft dependency relations with their dependers, dependees and depndums in Algorithm 4.5;
Create SD Model sdm_r,that belongs to *SD_Model={sdm_0, sdm_1, ... sdm_p, ..., sdm_r}, $0 \leq p \leq r$;*
For each actor $actor_a$ *in* srm_b
Delete the bdr_u of $actor_a$ keeping on the $actor_a$. dependency relationships of *Goal__Dependecy_REL* and *Soft_Goal__Dependecy_REL* ;
Add $actor_a$ *as a depender/dependee and associated in/outgoing dependency relations with their dependums into* ***Dependency_Rel****.*
End for each actor
// Checking the completeness of the dependency relationships in the SD model
For each dependency relation entry in the Dependency_Rel list
If *the dependency relation has unidentified depender/depndee then*
Mark this dependency relation in the Dependecy_Rel as uncomplete;
Else
Mark this dependency relation in the Dependecy_Rel as complete;
Add the dependency relation, its depdenser, dependee and dependum to the sdm_r;
End if

End for dependency relation entry

Figure 4.26: Algorithm 4.6, Deriving an SD Model from a Corresponding SR Model

In this section, the aim is to explicitly determine the design decisions of the siGoal ontology that semantically captures the goal-oriented models of the proposed strategic view (i.e., the BS model, the HSD model and the *i** framework models). Therefore, the aforementioned pseudo code-based algorithms (i.e., in Section 4.3.3.1) are utilised in order to exploit the required classes and their associated properties. Based on the algorithms, the key concepts of the GO models in the strategic view such as the BS model, the HBG, the HSD model, the SD model, SR model, actor, goal dependency, soft goal dependency, goal of SD, and soft goal, etc. reveal the key classes in the siGoal ontology. Each concept in the GO models is a class along with its attributes, as shown next in Section 4.3.3.3.

The siGoal ontology instantiation component establishes a strategic goal-oriented view instance for a business organisation, as it has been discussed in Chapter 3. The word strategic is borrowed from the common perspective of the BS, the HSD, the SD and the SR models, that are all strategic models with different abstraction levels. Each concept representation in the generated strategic GO view is an instance from its class in the designed ontology. For example, “appointments arranged” is an instance of the class goal and “patient” is an instance of the class actor. It is necessary to obtain the siGoal ontology in order to assist the systematic elicitation of the fundamental blocks of the Riva BPA, namely the EBEs as will be shown in Chapter 5. Moreover, the siGoal ontology design paves the way to link with the design of the sQuality ontology; thereby, in order to obtain the GQOnt for an organisation.

The GQR-BPM is a very rich model with relevant business GO concepts, for example actor, goal, task, resource, etc. Therefore, process models establish a start point in order to generate the proposed GO models as was previously shown in Figure 4.21. The abstraction level of elements and the information that derive designing the process models is very much similar to the one in the GO models. The aforementioned algorithms manifested the harmony between the GQR-BPM and the GO models for the BSV. For example, a GQR-BPM consists of role, goal, NFR, activity, etc. These elements are similar to the ones in the GO models that are actor, goal, soft goal, task, etc. The GQR-BPMs systematically allow deriving the BS and the HSD model considering that they are coarse grain GO models. The GQR-BPM concepts found easy to map with the two models. This is because the GQR-BPM comprises of the required information such as its parent goals and their associated quality requirements and constraints. Another example, a GQR-BPM derives a corresponding *i** framework, where the *i** framework is finer than BS and HSD models. Here, the GQR-BPM has been found easy to map its fine concepts with the *i** framework concepts. This is because the GQR-BPM employs the required knowledge for the *i** framework design such as actor, goal, task, etc. Moreover, the SD-model abstraction level is higher than the SR-model. Therefore, it is easier to map the concepts between the GQR-BPM and the SR-model, thereby deriving the SR-model from the former. Consequently, deriving the SD-model from the SR-model. Based on the mapping found between concepts, the *i** framework algorithms do firstly derive the SR-model from the GQR-BPMs and then generate the SD-models from the derived SR-model, as have been presented in Section 4.3.3.1.

The siGoal ontology starts to instantiate the BS model, the HSD models, the SD models and the SR models of the organisation where the models emerge from GQR-BPMs. Since the GQR-BPM is implicitly employed in order to derive the proposed sequence of GO models, then the siGoal ontology implicitly bridges BPMs and GO models within the BSV. Moreover, the relation between the GO models and the corresponding GQR-BPMs reveals a great harmony of concepts mapping in order to represent new knowledge. For example, in the SD model the goals are derived from the goals of the corresponding GQR-BPM. However, the goals may not be explicitly designed in the GQR-BPM.

For example, if x is true then y happens. This feature fits well with many conditional statements designed in the aforementioned algorithms, as has been shown in Section 4.3.3.1. SWRL rules are needed along with the classes and attributes in order to automate to some extent the generation of the GO models. The use of variables in SWRL rules facilitates the execution of a rule for more than one instance. This is a desired feature for this research work as the GQOnt instantiates the BSV of an organisation that consists of many GO models such as the SD and SR models with their elements. In particular, an organisation is likely to possess more than one goal model due to the presence of more than one collaborating business processes. Hence, each of the goal models and their components are represented using variables that SWRL rules provide. In addition, SWRL rules are executed using logical expressions.

Challenges

Most of the current approaches of modelling BPs do not explicitly include the goal and the soft goal concepts or at least their notations are absent in the GQR-BPMs (Yousef, 2010) (Aburub, 2006) (Ould, 2005). The reason behind this is that the BP model represents the workflow of activities in order to fulfil a set of goals. Therefore, deriving goals and soft goals from the GQR-BPMs is anticipated to be a challenge, as their derivation requires an elicitation and analysis work in the associated business documents. However, an effort has been put in the aforementioned algorithms (as presented in Section 4.3.3.1 in order to conquer some challenges to some extent.

4.3.3.3 The siGoal Ontology Classes and Properties

This section is responsible for showing classes and associated restrictions agreed for the siGoal ontology. The work of this section is attached in Appendix B. Figure 4.27 depicts a partial Meta-Model of the siGoal ontology.

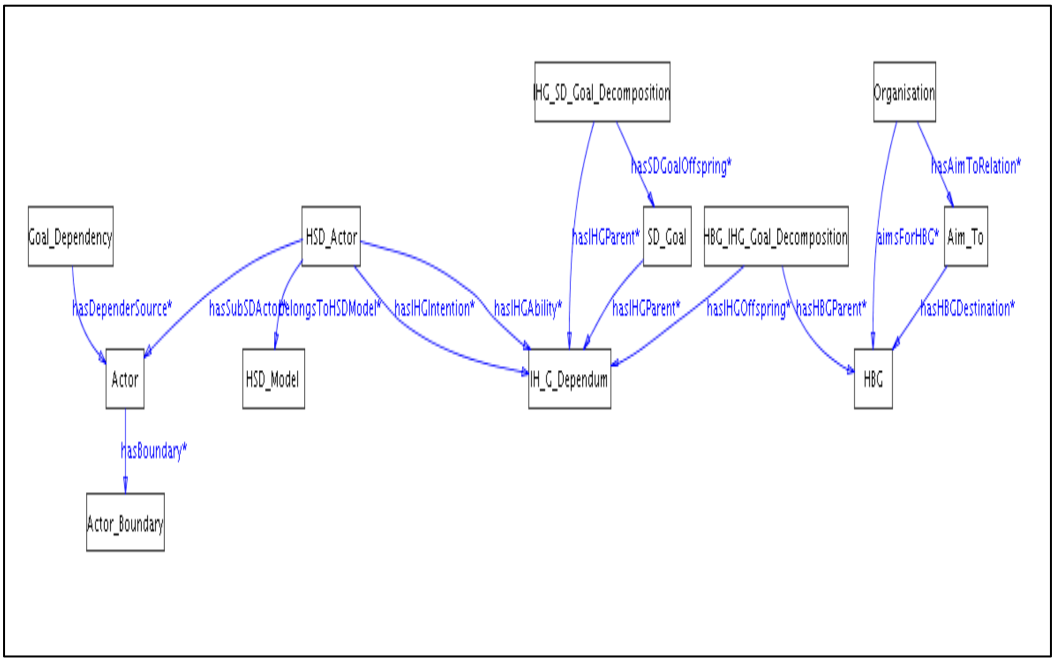


Figure 4.27: A Partial Metamodel Extracted from the siGoal Ontology.

4.3.3.4 Ontologising The GO Modelling Steps Using SWRL Rules and OWL-DL

In this section, the GO view modelling steps (i.e., for the BS, the HSD and the *i** framework models) within the BSV are recalled from Section 4.3.1. The GO view modelling steps are designed using about 28 SWRL rules, as attached in Appendix C.

4.3.3.5 The siGoal Ontology Instantiator

The siGoal ontology instantiator component is constructed in order to carry out the function of the formal representation of the GO view, which is extracted from the BSV, for an organisation using OWL-DL. It has already been mentioned in Chapter 3 Section 3.6.1 and in the introduction of this chapter that the input to this component is the interrelated GO models or view extracted from the original BSV. The output from this component is the instantiated siGoal ontology for the given organisation's BSV.

In the abstract siGoal ontology, the SWRL rules tab contains about 28 rules that implement the GO view modelling steps. In addition, these SWRL rules relate as OWL classes (i.e., about 42 classes within siGoal ontology) and the properties that bind the class instances with each other's or with a value. For example, if there is an SD model that exists in the GO view, then it should have a corresponding SR model that elaborates it. This example is simply implemented in the SWRL rule: SD_Model(?sdm) ^ hasCorrespondingSRModel(?sdm, ?srm) → SR_Model(?srm). Enabling this rule in the siGoal ontology and running the JESS engine plugged (Section 4.3.2) into Protégé will generate a number of instances (i.e., individuals) along with the related classes and properties.

The protégé ontology editor supports the visualisation of the metamodelling for the embedded ontology. A partial Meta-Model of the siGoal ontology was shown in Figure 4.27. Finally, the consistency of the siGoal ontology is checked using the built-in reasoner, namely Pellet 1.5.2, and the Racer Pro.

4.3.4 The sQuality Ontology Development

The sQuality ontology complements the function of the siGoal ontology through semantically representing the missing quality part, which is a required part in the overall GQOnt ontology in order to generate the entire ontologised picture of the desired BSV for a given organisation. In particular, it is needed to understand how the behavioural elements presented in the GO models are constrained with the desired quality requirements (i.e., soft goals and NFRs). The full representation of the quality requirements along with their satisfaction values should be linked into the GO view using the SR GO model, which will be shown in the next Section. Some progress and uncompleted on-going research work has been observed with regard to proposing a semantic representation of quality requirements using ontologies (Sancho et al, 2007) (Dobson et al, 2007). The slow progress may be attributed to few comprehensive and structured approaches proposed for designing quality requirements taking into account that quality requirements by nature involve complex knowledge especially that their validation can be at run-time. A comprehensive and recent quality-related representation offered by the NFR framework (Chung et al, 2000). Therefore, the NFR framework approach was advocated for representing the quality-related models within the BSV owing to that the NFR framework is a structured quality-oriented approach that fits well with the main role and place of the SR GO model, which is the gateway employed for the link with the NFR framework. In addition, the SR basic elements are found easy to link with the NFR framework notations and vice versa, as will be discussed in Section 4.3.5.

In this section, the aim is to introduce the semantic quality-oriented model. In this regard, the NFR framework in (Chung et al, 2000) is designed using OWL-DL. Also, the work of (Abu rub, 2006) is further employed in designing the NFR framework.

The next sub-sections are structured as follows. The work of the sQuality ontology through setting the design decisions that involve the NFR framework design is presented in Section 4.3.4.1. In Section 4.3.4.2, the design decisions are utilised in order to guide the identification of the classes and their associated attributes within the sQuality ontology. SWRL rules are designed for constructing the NFR framework approach are presented in Section 4.3.4.3.

4.3.4.1 The NFR Framework Design Decisions

The NFR framework is considered appropriate candidate employed on behalf of the quality-oriented modelling representation for the quality view within the GQOnt ontology design. In fact, the NFR framework approach proposed by Chung in (Chung et al, 2000) generates the soft goal interdependency graph for a particular main quality requirement (e.g., security or safety) considering that the quality-related concepts in the approach are denoted by the soft goals. The NFR framework has the credit among other similar approaches that involve complex quality concepts (Lamsweerde, 2009). In particular, the NFR framework approach is structured and found simple in representing the different perspectives of soft goals (i.e., quality perspectives such as ones related to their elaboration, measurement and evaluation).

In the design of the NFR framework approach, there is an explicit differentiating between the NFR type soft goal (i.e., root of SIG), NFR soft goal (i.e., sub soft goal of the NFR type soft goal) and an operationalisation (i.e., the leaf soft goal in the SIG). Consequently, the refinement relationships are also differentiated. The key concepts of the NFR framework are revealed as classes in the ontology design such as NFR type, NFR soft goal, operationalisation soft goal, AND decomposition, etc. The actual example of each class is an instance of the class, for example, "information availability" is an example of NFR type and "information related to patient" is an example of NFR soft goal that is sub part of the NFR type. The NFR soft goal and the operationalisation soft goal appear in the SIG diagram as ordinary cloud and thick/dotted cloud, respectively.

Figure 4.28 illustrates the common refinements cases within the NFR framework and Figure 4.29 illustrates the rare ones that consist of different soft goal types. In Figure 4.28, the cases (a), (b) and (d) represent common decomposition cases, as they were originally proposed in (Chung et al, 2000). Whereas in Figure 4.29, they are rare cases as they were discovered by Aburrub who employed the NFR framework in his work in (Aburub, 2006). The NFR framework approach uses the term decomposition when the parent and the off springs are from the same kind of soft goals (Chung et al, 2000). Cases (a) and (b) show the decomposition relation between NFR soft goals, considering that the NFR soft goal is referred as a type if it represents the root of the SIG. Case (d) depicts the decomposition of the operationalisation soft goals. In Figure 4.28, cases (c) and (e) represent the common explicit contribution from operationalisation soft goals to NFR soft goals

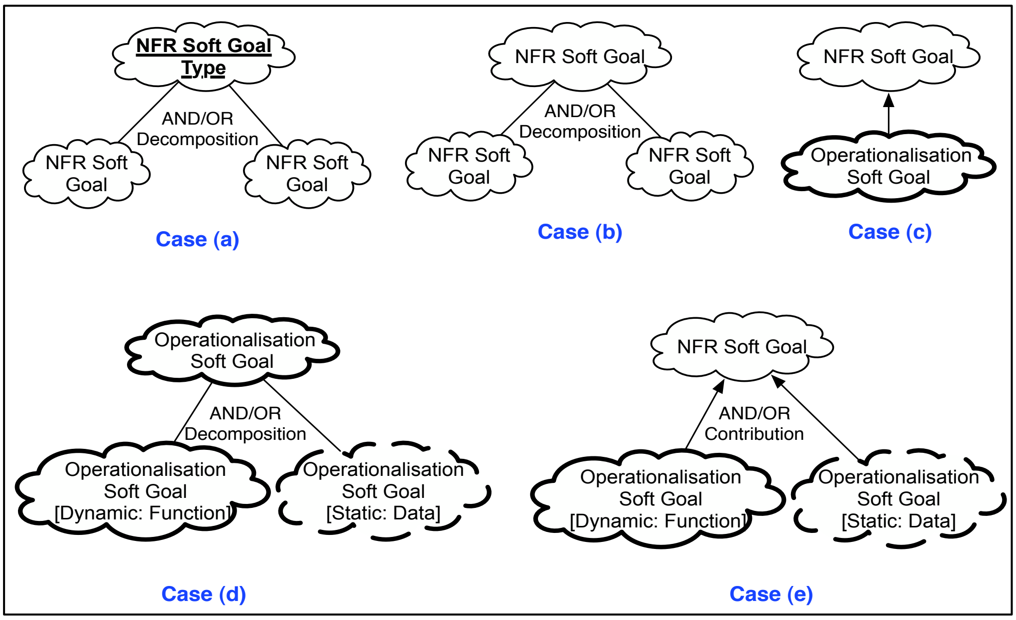


Figure 4.28: The Common Refinement Cases in the NFR Framework.

In Figure 4.29, rare contribution cases were found in (AbuRub, 2006), yet they addressed the NFR framework. Case (a) and (b) show an explicit contribution from the operationalisation to the **NFR type soft goal** immediately without contributing to NFR soft goal. In addition, the operationalisation joins the decomposition relation from the NFR type to its sub soft goal. This joining is addressed as well in the cases (c) and (d) in Figure 4.29 but with an NFR soft goal parent.

The cases shown in Figures 4.28 and 4.29 are represented in the OWL-DL sQuality ontology design. These were merged as can be seen in Table 4.1. In Table 4.1, the shaded cells imply that this case is not permitted at all. In this table, the NFR soft goal (row) is sub part of NFR type (column) and sub part of NFR soft goal (column); however, the NFR soft goal (row) is

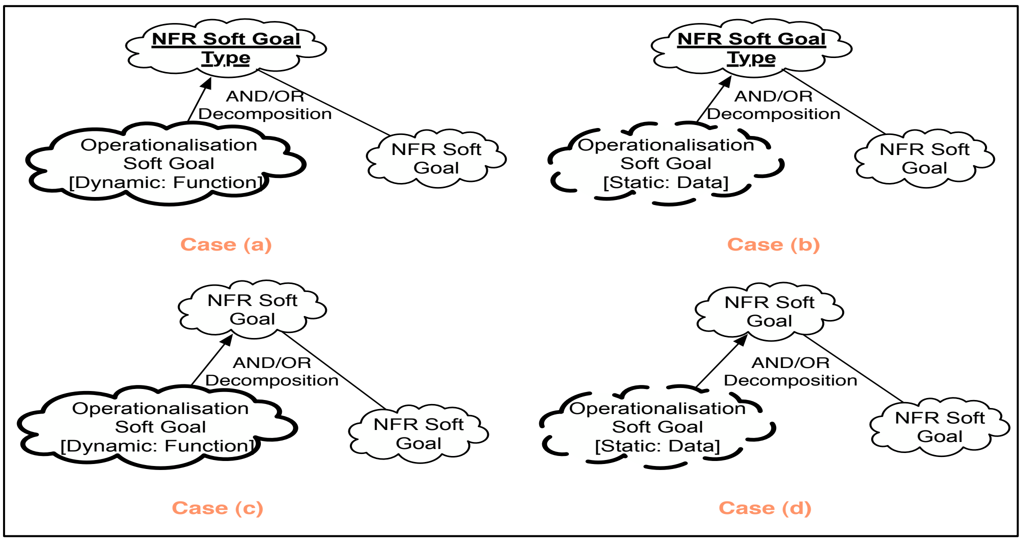


Figure 4.29: The Rare Decomposition Cases in the NFR Framework.

not allowed to be a sub part of an operationalisation soft goal (column). This table will be used with an example in (Section 4.4), but with additional column in order to link the NFR framework with corresponding *i** framework.

The process of designing the NFR framework is iterative. The requirement engineer, analyst, designer and/or developer has the control over the refinement of the soft goals and their extent. Moreover, the algorithm's steps iterate to generate the optimal SIG for a main soft goal. Thus, this may require a lot of human intervention in order to obtain the required NFR framework. It is not easy to determine a measuring value (i.e., make, help, hurt or break) for each soft goal. Yet, all types of soft goals must be measured for verification

Table 4.1: The Permitted Decomposition and Contribution Cases in the NFR Framework.

[Optional] Sub of/ Contributes ↗	**NFR type**	**NFR soft goal**	**Operationalization soft goal**
NFR type			
NFR soft goal			
Operationalization soft goal (<u>S</u>tatic: data /<u>D</u>ynamic: function)			

4.3.4.2 The NFR Framework Algorithm

The NFR framework is a structured quality-oriented approach (Chung et al, 2000) as has already been presented in Chapter 2 (Section 2.2.2.2). However, the priority and claim soft goal are not part of this research interest.

Since the GO view was derived from the GQR-BPMs, then it is normal to reuse them with a particular attention to the quality part (i.e., the NFR framework SIGs) in order to pave the way for the desired linking with the GO view.

Recall that the NFR framework was proposed in order to link the NFRs into the software systems (Chung et al, 2000); however, the NFR framework has proved its capability in linking the NFRs into a BPM. A related research work demonstrated the integration of the quality requirements into the R-BPMs and produced the QR-BPMs using the NFR framework (Abu rub, 2006). In the work of (Aburub ,2006), a quality-integrated BP, using the NFR framework, associated models that are the NFR frameworks, where each framework is concerned with addressing a particular desired main quality requirement that is represented in the form of soft goal concepts. Hence, this section does not provide any new algorithm for generating the NFR framework's SIGs using the BP models, but utilises the work of (Chung et al, 2000) (Aburub, 2006) that have already been presented in Chapter 2.

4.3.4.3 The sQuality Ontology Classes and Properties

This section aims to show the required classes and properties in order to represent the conceptualisation of the NFR framework in the sQuality ontology. The work of this section is attached in Appendix D. Figure 4.30 depicts a snapshot from the Protégé window that displays a partial Meta model of the sQuality ontology.

4.3.4.4 Ontologising the NFR Framework Steps and Rules

In this section, the NFR framework steps are recalled from Chapter 2 (Section 2.2.2.2) taking into account not considering the relevant steps for the identification of the claim soft goals and the soft goal priorities. Appendix E shows how those steps are ontologised using the rich ontology language that is OWL-DL.

4.3.4.5 The sQuality Ontology Instantiator

The sQuality instantiator component encapsulates the development of the NFR framework soft goal-oriented models. Accordingly, this component requires the NFR framework models and the abstract sNFR framework ontology as input in order to generate the instantiated sQuality ontology (i.e., an instantiation of the formal semantic representation of the NFR framework) for an organisation as was shown in Figure 4.1.

The development language and tool used in the sQuality ontology are presented in Section 4.3.1. The sQuality ontology instantiation is carried out similarly to the siGoal ontology instantiation explained in Section 4.3.3.5 using the SWRL rules tab editing and execution features.

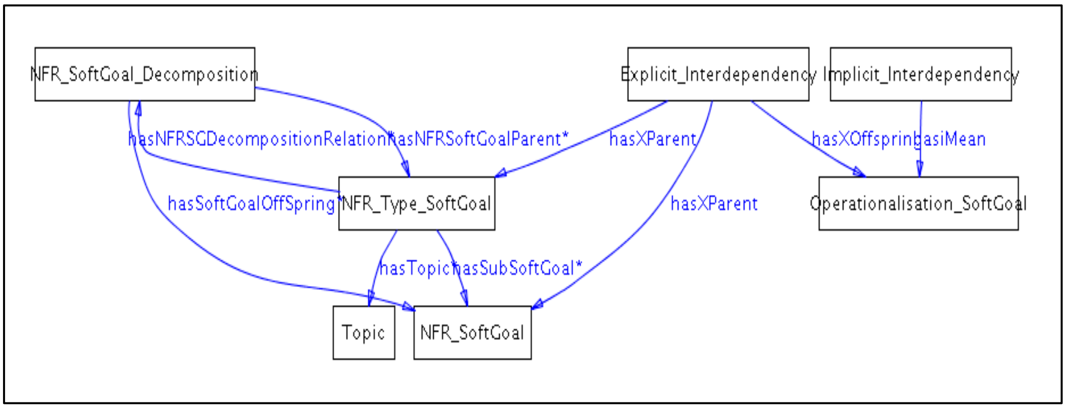


Figure 4.30: A partial NFR Framework Meta-Model Extracted from the sQuality Ontology.

4.3.5 The Linker Ontology Development

The task of this componenet is to link siGoal and the sQuality ontologies to form the GQOnt, using the notion of ontology repositories provided in Protégé. The siGoal ontology employed the refined *i** framework as a goal-oriented approach in order to formally represent the goal-oriented view within the BSV of the organisation's processes. The sQuality ontology employed the NFR framework approach in order to semantically instantiate the SIGs that record the design of how the ogrnaisation's NFRs are and analysed, designed and addressed. The siGoal and the sQuality ontologies are independently designed, implemented and instantiated. In this Section, the proposed linking cases between the GO models and the quality-oriented models are recalled from Section 4.3.1.5 due to the absence of an explicit linking method. Therefore, the linking cases are deduced rather than being borrowed from a relevant research work. In this Section, the four linking cases; which were previously presented in Figure 4.12 are developed into four linking rules in order to set up the link between the elements of both the siGoal and the sQuality ontologies. Once the link is detected and established, the NFR framework can be extracted from the sQuality ontology as shown in Figure 4.1.

According to the linking definition proposed within the BSV, the quality-oriented model (i.e., the NFR framework) can be linked with the *i** framework models using any of the four situations once required. It is necessary to highlight that it is not required to address the entire situations, as linking is carried out when a designer faces a situation that matches with any of the four cases depicted in Figure 4.12. If none of the situations have faced the designer, then a redesign activity must be carried out.

The first linking case relation is when a soft goal dependeum in a dependency relation is addressed in a corresponding NFR SIG in the dependee's boundary. In this situation, the entire SIG should belong to the dependee's boundary taking into account to the link the soft goal depdenum to the NFR soft goal within the SIG. **The second linking case relation** is when a particular NFR type soft goal is a sub of task within the actor boundary. In another words, the NFR soft goal constrains the task that the NFR is sub-part of. Accordingly, the entire NFR SIG should be integrated within the actor boundary. **The third linking relation** is when a task in the *i** framework (i.e., in the SR model) matches with a corresponding operationalisation soft goal or a dynamic operationalization soft goal that is involved in the design of a SIG for a particular NFR. In this situation, the task is replaced with the dynamic operationalization considering the integration of the entire SIG. Finally, **the fourth linking relation** is when a resource within the actor boundary in the SR model matches with a corresponding static operationalisation soft goal that is involved in the design of the relevant SIG diagram. In this situation, the resource is replaced by the static operationalisation taking into account integrating its entire SIG diagram that it belongs to in case it does not currently exist in the actor's boundary.

In the light of the above, Table 4.2 extends the original Table 4.1, using the additional yellow column, in order to reference the linking rule used with the NFR framework SIG integrated into the relevant *i** framework models. Therefore, Table 4.2 can be used for two purposes. The first purpose is regarding the NFR framework SIG representation similar to the one in Table 4.1. The second purpose is to explicitly manifest the linking between the NFR framework with the *i** framework. Hence, Table 4.2 occupies the role of bridging between the relevant instantiation of the siGoal ontology and the sQuality ontology. Thus, Table 4.2 is provides traceability support between the elements in the sQuality ontology and the siGoal ontology.

Table 4.2: Linking the *i Framework and the NFR Framework.**

Optional] Sub of/ Contributes ➚	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
NFR type				
NFR soft goal				
Operationalization soft goal (<u>S</u>tatic: data **/<u>D</u>ynamic:** function**)**				

In the GQOnt ontology, it is required to formally represent the organisational BSV that is divided into GO models and quality-oriented models taking into account to distinguish between the two types of models. Therefore, the GO elements that are linkable with their relevant quality model (i.e., the NFR framework SIG) have been defined semantically using OWL-DL. It is important to recall that not all the GO elements (e.g., IH-G, HSD actors, and goals) in the GO models accommodate linking with the quality-oriented models. In particular, only the soft goal depndeum, the sub soft goal for a task using the task decomposition relation and the task or resource that matches with a corresponding operatioanlisation within the actor boundary have the merit to integrate with the NFR framework SIG. Therefore, the formal representation of the NFR framework SIG is needed in order to instantiate the quality-oriented model using the sQuality and accordingly linking the two ontologies (i.e., the siGoal and the sQuality). The four linking rules that have been proposed permits that GO linkable elements the connection with a soft goal in the relevant SIG diagram.

Since the four linking rules have been identified, then four corresponding functional object properties along with their domains and ranges are implemented as follows: (1) matchesWithNFRTypeSIG whose domain is either a Soft_Goal_Dependum or Soft_Goal (from the siGoal ontology) and whose range is the class SIG_Diagram. This is a 2 in1 rule, which covers two linking cases that are the first and the second linking cases. (2) correspondsAnOperationalisationSoftGoal whose domain is a Task (from siGoal ontology) and whose range is the class Operationalization_SoftGoal or Dynamic_Operationalisation (from the sQuality ontology) this rule covers the third linking case. (3) correspondsAStaticOperationalisationSoftGoal whose domain is the class Resource (from the siGoal ontology) and whose range is the class Static_Operationalisation (from the sQuality ontology). The SR model in the GO models embodies the four linking cases and thereby the model is very important to emphasise the linking.

Accordingly, the OWL restrictions are constructed for the classes Soft_Goal_Dependum, Soft_Goal, Task and Resource as follows:

- Soft_Goal_Dependum: ∀ matchesWithNFRTypeSIG only SIG_Diagram.
- Soft_Goal: ∀ matchesWithNFRTypeSIG only SIG_Diagram.
- Task: ∀ correspondsAnOperationalisationSoftGoal only (Operationalisation_SoftGoal or Dynamic_Operationalisation).
- Resource: ∀ correspondsAStaticOperationalisationSoftGoal only Static_Operationalisation.

Figure 4.31 depicts a partial illustration of the GQOnt that links between the siGoal and the sQuality ontologies that both are distinguished using the label “G” and “SG”, respectively.

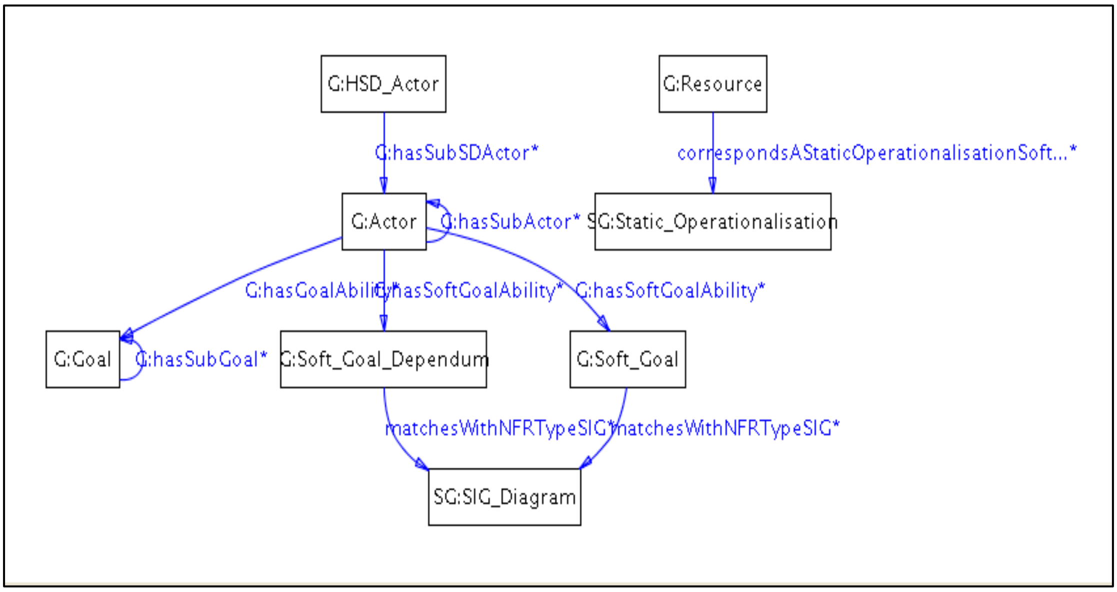


Figure 4.31: Part of the GQ Ontology, showing how the siGoal and the sQuality ontologies are linked together.

4.4 Instantiating of the GQ Ontology using the Patient Reception Business Process within the CCR Process in the Kingdom of Jordan

This section aims at presenting the instantiation work of GQOnt using a pilot study from the health care domain located in the Kingdom of Jordan, namely the patient reception business process. The pilot case study is extracted from the main CCR case study conducted within the King Hussein Cancer Care Centre (KHCC) in (Aburub, 2006).

The patient reception process is a GQR-BPM, as attached in Appendix F (Figure F.1), that is designed for a set of particular objectives. It " formally registers patients in the hospital, books appointments, organises patients' appointments, and checks if the patient has been diagnosed" (Aburub, 2006). Three main roles collaborate in order to address the aforementioned objectives, which are the outpatient receptionist, the patient and the medical records clerk. The aforementioned objectives are derived from the main objective of the CCR that is "to improve the administration of cancer treatment and the collection of information about cancer cases" (Aburub, 2006). The outpatient receptionist is responsible "to arrange appointments with specialists for diagnosed patients" where the medical records clerk is responsible for the "essential administrative tasks in the CCR process, including managing patients' manual files, performing hospital cancer registration, and sending required information to the Jordan Cancer Registry (JCR)" (Aburub, 2006). Finally, the patient (i.e., non/diagnosed) appear as a highly depending role on the others where s/he "aims to get the suitable treatment and follow up" (Aburub, 2006) through collaborating with the other roles in the domain. Nine main quality requirements constrain this process, which are: empathy, responsiveness, assurance, reliability (i.e., regarding the prompt execution of the business process), the information domain, the information availability NFR, the system user satisfaction NFR and confidentiality NFR (Aburub, 2006).

Although the CCR process embodies goals, yet the goal-oriented models of the CCR business process are absent. However, the NFR framework models were designed for each corresponding goal-based business process within the CCR (Aburub, 2006). Therefore, the six algorithms presented in Section 4.3.3.1 are reused to create the GO models for the patient reception process.

This Section is structured as follows. Section 4.4.1 presents the instantiation of the siGoal ontology for the patient reception process, where its instantiation of the sQuality ontology is presented in Section 4.4.2. Finally, the linking that produces the desired GQOnt ontology (i.e., the formal semantic representation of the BSV for the patient reception process) is shown in Section 4.4.3.

4.4.1 The siGoal Ontology Instantiation Using the Patient Reception Process Pilot Study

In this section, the instantiation of the siGoal ontology using the patient reception process example is presented in order to formally create the semantic representation of the GO view within the BSV for the process. The six algorithms presented in Section 4.3.3.1 are used in order to generate the BS model, the HSD models and the *i** framework models for the patient reception process that are in Appendix G (Section G.1).

4.4.2 The sQuality Ontology Instantiation Using the Patient Reception Process Pilot Study

In this section, the sQuality ontology instantiation for the patient reception process is presented as attached in Appendix G (Section G.2).

4.4.3 The siGoal-sQuality Ontology Linker Instantiation Using the Patient Reception Process Pilot Study

In this section, the linking of the sQuality ontology within the siGoal ontology of the patient reception process is represented. The work of this section is attached in Appendix G (Section G.3).

4.5 Discussion and Conclusion

This chapter is concerned with constructing and developing the first layer of the GQ-BPAOntoSOA framework, namely the GQ ontology layer. The layer is quiet new in relation to the layers in the BPAOntoSOA framework. The function of this layer is to semantically represent the BSV for an organisation taking into account addressing the separation of concerns principle through distinguishing explicitly between the GO models and the soft goal-oriented models. The deliverable from this layer is the instantiation of the GQ ontology for an organisation that comprises of the instantiation of the siGoal ontology, where the i* framework (Yu, 1995) concepts form the backbone of the siGoal ontology, and the instantiation of the sQuality ontology, where the NFR framework (Chung et al, 2000) concepts form the backbone of the later ontology. The work of this layer has been initially evaluated using a pilot study (i.e., patient reception process) extracted from the main case study of this research that is the CCR case study (AbuRub, 2006).

In this section, the strength, weakness and limitation aspects rose while generating the three ontologies are discussed. The characteristics of the ontologies are also considered.

4.5.1 The Overall GQ Ontology

In Appendix G, the figures G.1, G.5, G.7, G.11 and G.15 represent the BSV for the patient reception process as a pilot study exploited from the CCR main case study in this research. It is observed that the GO models and associated soft goal-oriented models within the BSV are very rich of relevant business-oriented concepts (e.g., actor, goal, soft goal, task, resource, operationalization, NFR type, etc) mapped to the corresponding GQR-BPM. Although the GQOnt meta-model is rich in terms of the quantity of concepts, they still all live in harmony in one model (i.e., the GQOnt conceptual model) that conceptualises the BSV in order to understand the organisation in the early stages of the requirements engineering process. The BSV conceptual model (i.e., the GQOnt ontology) bridges the gap between the goal-oriented model, soft goal-oriented models and the GQR-BPMs of an organisation. Therefore, the GQOnt conceptual model initiates a further bridging with a BPA discussed in Chapter 5.

The GQ ontology provides **traceability** in both directions, which is from the elements in the siGoal ontology to the elements in the sQuality ontology, and vice versa. This traceability feature assists in detecting the implications if any changes or improvements occur in the proposed BSV of an organisation. In addition, this traceability is supported with information about the required dependencies and rationales that are between and within the actors in the GQOnt ontology. The provided traceability informs the requirement engineers that all the GQOnt ontology elements are interrelated and thereby must keep this traceability consistent with respect to any required changes. In addition, this traceability notion is anticipated to support answering queries regarding any element that live in the GQOnt ontology.

The instantiation of the GQOnt of an organisation is **automated**, using Protégé, but with limitations. For example, for each SD model there is a corresponding SR model that is created automatically. However, the creation of goals and tasks in the SR model is carried out manually.

The GQOnt **bridges the gap** between the goal-oriented, soft goal-oriented and GQR-BPMs. A GQR-BPM has a corresponding BSV represented using the GQOnt ontology that is not concerned with the detailed workflow, yet with understanding the dependencies and rationales beyond the BP.

The GQOnt provides a **comprehensive business strategic model** owing to its embodying to strategic (e.g., the hard goals and soft goals) and tactical elements (e.g., tasks, resources and operationalisations). Although there are new elements that have been identified in the GQOnt stemming from the GQR-BPM (e.g., goals and main NFR), this does not add any new knowledge to the corresponding GQR-BPM. In other words, the derivation of the GQOnt from the GQR-BPM does not amend the process model representation and/or content. However, the identification of new elements in GQ ontology alerts the requirements engineer about the missing but required elements (e.g., goals and main soft goals) that do not exist explicitly in the GQR-BPMs. Similarly, if any of the GQR-BPM elements is not found in the GQOnt ontology or vice versa, then this alerts the requirements engineer for a gap within the analysis and the implementation. By and large, the GQR-BPM elements are subset from its corresponding GQOnt ontology.

In this research, the BSV is proposed as a way to addresses **clearly** the oragnisation's strategical elements (e.g., HBG, IH-G, goals and main soft goals), and tactical elements (e.g., tasks, resources and operationalisations) along with their holders (e.g., HSD actor and actors in the i* framework models). This clarity encourages more unformed communication between related stakeholders in managing the present and the future business strategy of the organisation. In addition, a clear BSV is anticipated to assist in validating the correctness, completeness and consistency of the strategical elements and their associated tactics along with the holders. In short, this clarity is manifested in the BSV classification in relation to the notion of change as shown in Figure 4.32 The static and dynamic categories are anticipated to encourage the reusability of its instances. For example, the BS and the HSD models are static models as well their components, which they rarely change. The main NFR in the SIG is another instance from the static category. The SR model is considered a dynamic model because it embodies most of the tactical elements (e.g., tasks) adjusted to satisfy the associated strategical elements (e.g., goals). Both categories encourage the **reusability** of their instances based on the notion of change. In particular, if the organisation is moving for radical changes, then the static instances are reused and reviewed. Similarly, if the organisation plans to amend its tactics to improve the satisfaction of its strategies, then the instances of the dynamic category are reused, reviewed and adjusted as per the need for change. Therefore, the entire BSV models and components are in **evolution** with respect to change.

In the proposed BSV, the elegant integration using the elaboration notion between the GO models produced the overall goal network as shown in Appendix G (Section G.1). The goal network is conceptualised in the GQOnt that is based on the direction from the generalisation to the specialisation of goals without considering their holders and their change.

Overall, the GQOnt embodies the functional (i.e., stemming from the goals) and the NFRs (i.e., stemming from the soft goals) and it includes both the coarse-grain models (e.g., the BS,

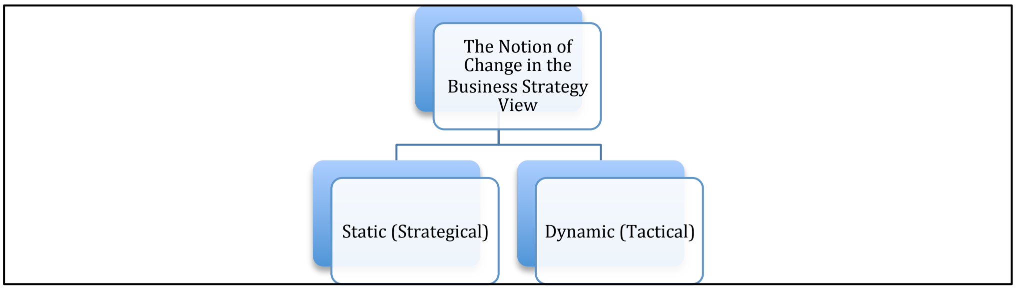


Figure 4.32: The Notion of Change Classification in the proposed BSV.

the HSD and the SD models) and fine-grain models (e.g., the SR model). As it has been shown in Section 4.3.4 that the linker component links the siGoal ontology with sQuality ontology using any of the three identified linking relations.

4.5.2 The siGoal Ontology

Since the GO models (i.e., the BS, the HSD and the *i** framework models) employ the dependency notion, then the siGoal ontology is rich in conceptualising the dependency notion between actors in the organisation in order to fulfil the associated parent goal. Moreover, the siGoal is rich in conceptualising information about rationales within the internal representation of an actor. However, the siGoal ontology does not provide rich quality-oriented concepts, yet it only provides soft goal-related concepts with very limited relationships. This is because the role of the siGoal ontology is in conceptualising only the GO concepts and the relations between them.

Furthermore, the siGoal ontology **semantically bridges the gap** between GO models that are rich of strategic elements (e.g., the BS, the HSD and the SD models) and GO models that are rich of tactics (e.g., the SR models). The semantic goal network, that comprises the entire goals regardless of their notion of change (i.e., strategical or tactical), manifests the semantic bridge.

It is not easy to automate the instantiation of the GO models and their components for adjusting the goals and the relationships between them. However, the siGoal ontology instantiation component in the GQOnt ontology instantiation layer addresses the automation to some great extent with the help of SWRL rules.

The siGoal ontology is a repository that assists the stakeholders in acquisitioning the goal-related knowledge within the BSV without paying effort to distinguishing between goal and soft goal perspectives addressed in the sQuality ontology. In addition, the siGoal ontology assists in tracing the goal holders, which are interested in achieving the goals using the dependency notion which is very rare in the current GO approaches (Kavakli and Loucopoulos, 2005). This perspective is considered as a credit for the *i** framework among other alternatives (Kavakli and Loucopoulos, 2005).

4.5.3 The sQuality Ontology

The sQuality ontology forms a repository of quality-related knowledge that assists stakeholders in acquisitioning the relevant knowledge without the need to trace it back to the associated GO models within the GQOnt.

The instantiation of the sQuality ontology provides advanced information about the desired quality requirements (i.e., NFRs) that do not exist in the siGoal ontology. Such advanced information consists of the decomposition of soft goals, mechanisms that fulfil the soft goals, etc. However, neither the network of dependencies nor their strategic rationales are conceptualised in the sQuality ontology. Addressing a soft goal is not a straightforward process and requires a particular approach in order to design the related concepts and relationships between them. Therefore, the siGoal ontology is only concerned with conceptualising the network of dependencies and the rationales of actors where the sQuality ontology is a soft goal-oriented conceptualisation. Hence, linking the two ontologies produce a comprehensive strategic knowledge. The work of the sQuality ontology joins the recent research work attempts in producing a quality-oriented Meta-Model and particularly the rare ones to conceptualise the NFR framework (Chung et al, 2000).

The work of this chapter (i.e., the GQOnt ontology development) has few limitations such as **ones** related to the dependency notion of the organisation that is considered as individual self-dependent organisation in addressing its business goals. In particular, the GQOnt ontology is not targeted to an organisation that is dependant on another organisation in addressing business goals. The **second** limitation is related to the time factor of goals that is explicitly involved neither in the BSV proposed in Section 4.3.1, nor in the GQOnt ontology. The **third** limitation is related to the risk management. Although the BSV proposed here does not address explicitly the risks and their implication, the *i** framework is still employed in the view to support addressing the risk facets and their mitigation (Yu et al, 2010). The **fourth** limitation is the absence of the cost prediction. This is because it is difficult to estimate the cost of addressing the business goals due to their strategy notion. The cost prediction is anticpated to be addressed in a bottom-up manner (i.e., from the tactical elements toward the strategical elements upward) rather than the top-down (i.e., from the strategical elements toward the tactical elements downward). For the BSV, it is anticpated that it will be easier to determine the cost of the finest-grain elements (i.e., tasks and operationalisations), as they are easy to measure and manage in order to estimate the cost of related coarse grain elements (i.e., the strategical elements such as goals and soft goals). The **fifth** limitation is related to the need for a comprehensive evaluation using case studies from different domains in order to strength the claim of the GQOnt ontology domain independency. The **sixth** limitation is concerned with priorities to goals.

Finally, this chapter answers the RQ2 and part of the third RQ3 as shown in Figure 4.33. Also, this work occupies a space in the activated research phases shown in Figure 4.34.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1				✓					
RQ2				✓					
RQ3									
RQ4									

Figure 4.33: Answered Research Questions in the Work of Chapter 4

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									**Activated**
RP2				✓					**Active**
RP3				✓					**Active**
RP4(ai)									
RP4(aii)									
RP4(bi)				✓					**Active**
RP4(bii)									
RP4(biii)									
RP5									
RP6									

Figure 4.34: Active Research Process Phases in the Work of Chapter 4

Chapter Five: A Semantic-Based Alignment of a Riva-based Business Process Architecture to the Business Strategy View of An Enterprise

5.1 Introduction

This chapter presents the work carried out within the second layer of the GQ-BPAOntoSOA framework, namely the GQ-BPAOnt ontology instantiation layer. This layer is originally borrowed from the BPAOntoSOA framework and particularly the BPAOnt ontology instantiation layer (Yousef, 2010), as shown in Figure 5.1, but extended through the proposed integration of goals and quality requirements. The original BPAOnt ontology, which represents an OWL-DL semantic representation of a Riva-BPA and its associated BPs, in the BPAOntoSOA framework lacks the integration of goals and quality requirements. The employed Riva method (Ould, 2006) in the original BPAOnt suffers from those two shortcomings and thus resulted an instantiated BPAOnt without addressing goals and quality requirements.

The refined layer inputs the pre-existing Riva BPA, associated BPMs along with the BSV for a given organisation in order to output the semantic representation of the goal-based and quality-linked Riva BPA and associated GQ-BPMs. The GQ-BPAOnt ontology instantiation layer refines the original conceptualisation of the Riva BPA and the associated BPMs along with merging rules as shown in Figure 5.2. Accordingly, the extended layer consists of two main ontologies and there are the GQ-srBPA and the GQ-sBPMN ontologies. The first ontology is reused from (Yousef, 2010) with refinements in the conceptualisation of the Riva BPA and associated relevant components using the influence of the BSV. The second ontology sBPMN (SUPER, 2008) (Yousef, 2010) refines the conceptualisation of the associated BPMs designed using BPMN using the influence of the enriched Riva BPA. Those two ontologies are merged using the original and new rules, as will be shown in this chapter. The entire work of this chapter represents work conducted three research phases of the research process discussed in Chapter 3. There are the early theoretical framework design phase, the investigation phase and the original BPAOntoSOA framework enhancement phase (i.e., particularly in the activities related to the BPAOnt in revisiting and reconstructing).

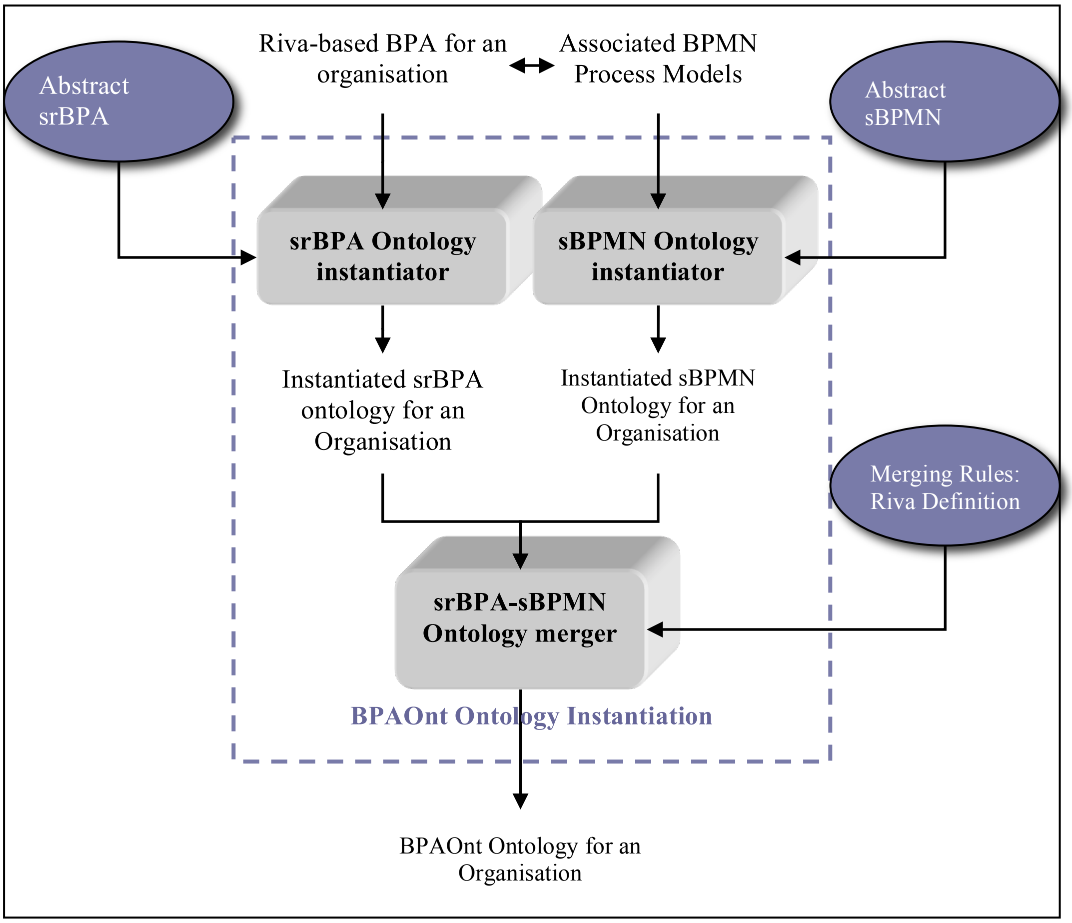


Figure 5.1: The Original BPAOnt Instantiation Layer from the BPAOntoSOA Framework [Source: (Yousef, 2010), Used with the author's permission]

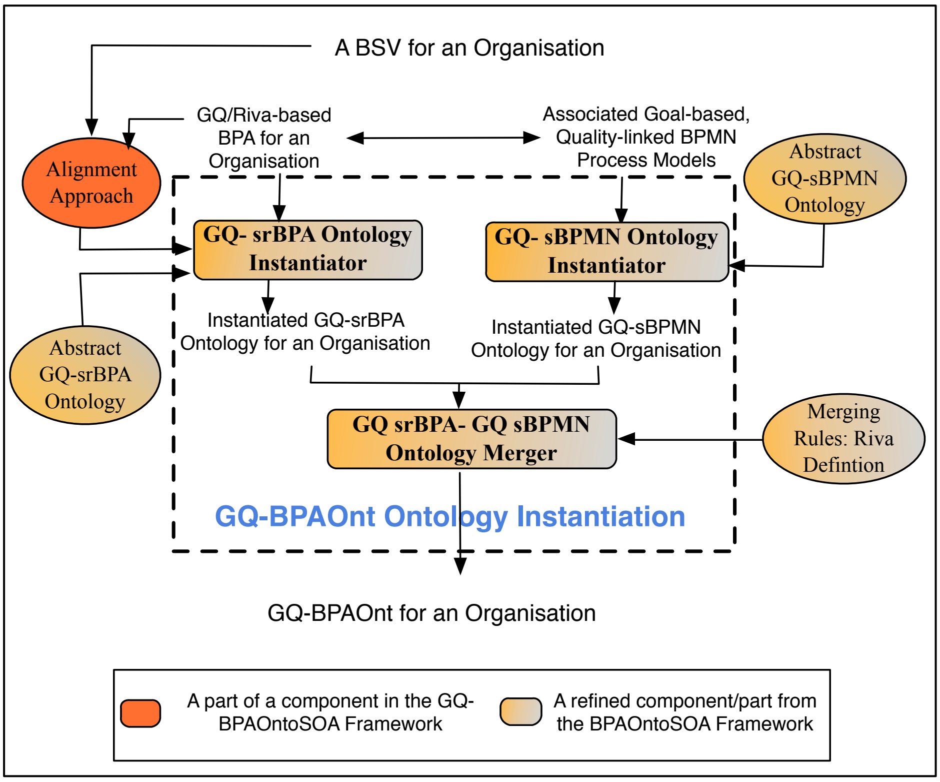


Figure 5.2: The GQ-BPAOnt Ontology Instantiation Layer in the New Framework

This chapter is structured as follows; Section 5.2 presents a brief contrast of the original layer to the new layer. Section 5.3 presents the reasons behind integrating goals into the original BPAOnt ontology instantiation layer using an alignment approach. Section 5.4 provides the reasons behind the need of integrating quality requirements into the Riva method and thereby the BPAOnt. Also, suggestions for refining and enriching the original Riva method with goals and quality requirements without referring to the BSV are discussed in Section 5.5. Similarly, suggestions to represent goals and quality requirements in BPMN are introduced in Section 5.6. The specification of GQ-BPAOnt ontology in this layer is introduced in Section 5.7. In Section 5.8, two alignment approaches are proposed to align the current Riva BPA to the BSV. Those two alignment approaches are ontologised in Section 5.9. In Section 5.10, the CEMS Faculty of Administration is employed as a pilot study in order to evaluate the work of this chapter. Finally, the work of this chapter is closed with a discussion and conclusion in Section 5.11.

5.2 Comparing the Original BPAOnt and the GQ-BPAOnt Ontology Instantiation Layers

In this section, the original work of the BPAOnt ontology instantiation layer is recalled (i.e., presented in Chapter 2 and shown in Figure 5.1). This recalled is needed in order to briefly highlight the major differences between the original layer and its proposed extension. In particular, the original BPAOnt is extended into the GQ-BPAOnt ontology instantiation layer as shown in Figure 5.2 in order to utilise the notion of goals and quality. Accordingly, the extended structure of the BPAOnt ontology instantiation layer requires as inputs the Goal-based and Quality-linked Riva BPA (GQ-Riva BPA) and its associated Goal-based and Quality-linked BPMNs (GQ-BPMNs) (i.e., whether they resulted from an alignment with pre-existing Riva BPA or not). Those refined inputs are processed within the same 3 components; however, they are functionally extending the function of the corresponding original ones in the BPAOnt ontology instantiation layer in order to generate the desired output, namely the instantiation for the GQ-BPAOnt for the given organisation.

5.3 Rationales Behind the Integration of Goals into the BPAOnt Ontology Instantiation Layer Using An Alignment Notion

In this section, the difference between the meaning of deriving a Riva-based BPA from business goals and the alignment of business goals with a Riva-based BPA is discussed. In addition to the motivations and gained benefits from the chosen notion, which is the notion of alignment rather than the derivation. This is because the BPAOntoSOA framework lacks the integration of goals, as this shortcoming was not noted in the future work in (Yousef, 2010).

There are two ways for generating a Riva-based BPA model. The first direction produces the BPA model by establishing its fundamental blocks without depending on other models. The second direction produces a Riva BPA model through reusing the pre-existing Riva-BPA model or any of its components such as the as-is EBEs list, UoW diagrams, 1st cut and/or 2nd cut architectures. In simple words, the alignment concept embodies the notion of reuse unlike the former direction (i.e., the derivation one). For example, a Riva-BPA that is purely derived from non-goal-based entities or models (e.g., BPs) is anticipated to address a difficulty in the reusing for the alignment purposes to meet business needs' changes. This is because Riva-based BPA elements are not originally designed to meet particular business goals. However, this claim requires further research.

However, the generic benefits of reuse ought to be considered. Since the alignment strategy operationalises the notion of reuse, then it is anticipated to generate the goal-based BPA model with the minimal architecting effort. In this research, a reuse or an alignment is required when an analyst aims at improving organisational models (e.g., BPAs and BPMs) in order to meet a particular need that may emerge in the environment. Consequently, this permits meeting the need or changing through reusing pre-existing models for the desired improvement and thus increasing the competition among other similar business organisations.

With regard to the original BPAOntoSOA framework, the rationales beyond asserting the alignment strategy instead of the derivation are as follows: (1-) the pre-existing Riva BPA semantic models already exist and were evaluated, (2-) the BPAOntoSOA framework did not address the BSV that embodies the hard goals and soft goals while generating a Riva-BPA for an organisation. Therefore, it maybe asserted that the pre-existing BPA was generated with the absence of the goal-oriented models. Hence, this requires bridging the gap between the semantic Riva-based BPA and the semantic goal-oriented models, as presented in Chapter 4.

5.4 Rationales Behind the Quality Integration into the BPAOnt Ontology Instantiation Layer

Since quality requirements were not integrated into the original BPAOntoSOA framework (e.g., security), then their absence is one of the main shortcomings found in the work in (Yousef, 2010). This shortcoming was explicitly highlighted in the future work plans in (Yousef, 2010), and mainly the QoS issues. This is because the identified candidate software services generated using the BPAOntoSOA framework lack the identification of associated non-functional requirements.

This section aims at briefly emphasising the necessity of the quality requirements' integration into the original BPAOnt ontology instantiation layer with reference to using soft goal-oriented models, as these models involve an early identification of quality requirements. Therefore, this section is not concerned with the alignment or deriving the QoS, as this will be discussed in Section 6.5.

By and large, the quality requirements refer to the desired attributes or characteristics on the product, business activity or service, for example reliability is a desired quality requirement in the flight booking business process. According to what have been learned in Chapter 2, if a Riva-BPA of an organisation that considers the flight booking as one of its key activities with an interesting lifetime, then a Riva-BPA documents it as a UoW in the UoW diagram and thus in the process architecture. However, the current Riva method does not document the associated quality requirements that constrain the business process execution in the process architecture. In addition, no attempts were found regarding the quality integration into Riva method due to its recentness as a BP architecting method that appear to require further development. Although it is recent method, it is considered as an easy to understand and has the potential to extend with goals and quality requirements. Also, quality requirements have the potential to propagate into the detailed representation of BPs.

The Riva method starts with identifying and understanding the business domain an organisation is involved in. Since the GQOnt ontology involves the identification of soft goals as part of understanding an organisation as discussed in Chapter 4, then this ontology can be utilised in order to pave the way for quality requirements integration into Riva BPAs. In addition, the GQOnt ontology is rich with related concepts (e.g., soft goal, operationalisation, etc.) This would establish as a quality-oriented bridge between the GQOnt ontology presented in Chapter 4 and BPAOnt ontology developed by Yousef in (Yousef, 2010).

5.5 The Goal-based and Quality-Linked BPA: Refining the Original Riva Method

The current Riva method is revisited (shown in Figure 2.18 from Chapter 2) and extended into a method that derives a Goal-based and Quality-linked Riva BPA (GQ-Riva BPA) as depicted in Figure 5.3. The following are the steps utilised to integrate goals and quality requirements into a riva-based BPA:

Step 1: Agree the organisation's boundary

Agreeing on the organisation's boundary results in the scoping of the Business Universe (BU). Recalling the work of Chapter 4, the GO models that embody the *i** framework or the entire BSV is one of the alternatives in this research in order to shape the boundary of an organisation, as will be shown in the alignment approach proposed in Section 5.8.

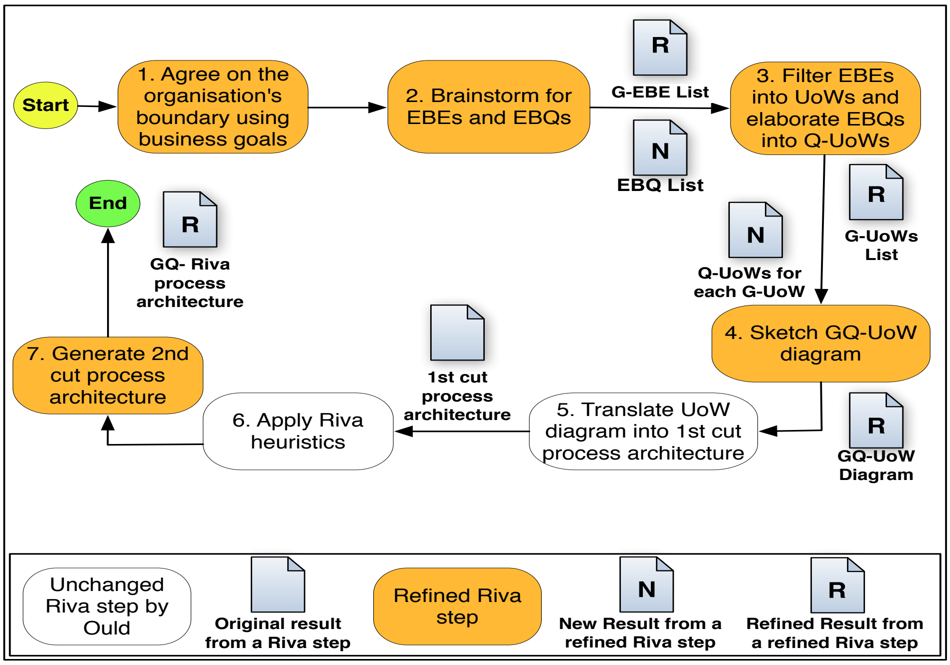


Figure 5.3: The Refined Riva Method

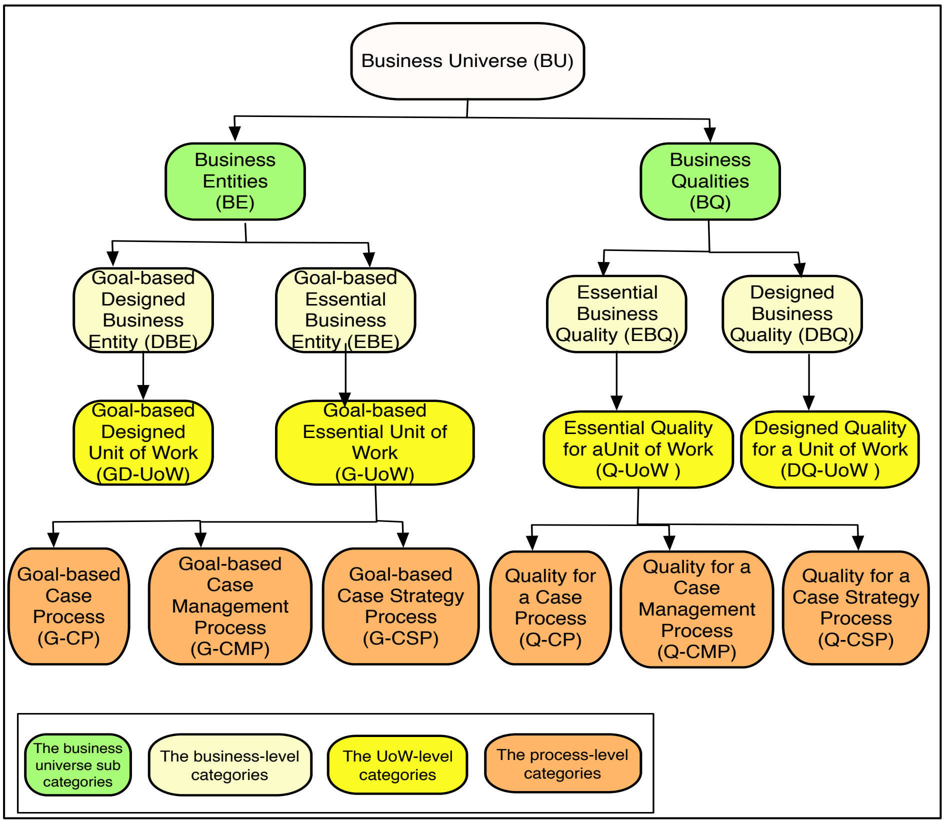


Figure 5.4: The Business Universe Categorisation for the Refined Riva Method.

The Business Universe (BU) represents the set of elements that are classified and appropriately related based on the Riva method guidelines/rules in order to manage the systematic derivation of the GQ-Riva BPA (i.e., the 2nd cut architecture). The BU acts as the home repository of the given information (i.e., for the Riva BPA design), which is processed in the form of generating its classifications where their instances are turned into knowledge for the Riva BPA design and associated BPs. The BU categorisation appears in Figure 5.4. In fact, the BEs and the BQs are the two main sub classes that constitute the BU. However, some researchers may wish to extend the classification into more than two main sub categories based on their point of view regarding the BU.

In Figure 5.4, Business Entities (BE) is the set of the sets (i.e., Riva original concepts) that were presented in the original Riva method (e.g., EBEs, DBEs, UoWs, D-UoWs, CPs and CMPs) (Ould, 2006). Where the Business Qualities (BQs) or the Quality of the Business is the second main sub set that mirrors its sibling, which is the BE, but from the quality point of view. Accordingly, the BQ is categorised into the sets Essential Business Quality (EBQ), Designed Business quality (DBQ), Quality for a UoW (Q-UoW), Designed Quality for a UoW (DQ- UoW), Quality on a CP (Q-CP) and the Quality on a CMP (Q-CMP).

The BQ set emerged due to the absence of quality within the current Riva BPA method (Ould, 2006).

Step 2: Brainstorm for EBEs and EBQs

With regard to the BE category, the identification of the EBEs mechanism is the same for the one in (Ould, 2006). However, the notion of goals (i.e., hard goals, aims and objectives) is added to the original EBE heuristics proposed by Ould. Recalling the background presented in Chapter 2 (Section 2.3.1), EBEs should be distinguished from the DBEs.

With regard to the BQ category, this step involves the identification of EBQs and factorising out DBQs. The Essential Business Quality (EBQs) requirements are identified, and they simply refer to the main quality requirements, NFRs or soft goals (e.g., security and information availability) within the agreed organisation boundary. In addition, EBQs are considered as the constraints that constrain the entire or subset of the identified EBEs. Hence, this highlights on the priority given for the identification of EBEs prior to EBQs. The identification of the EBQs is compulsory, as the business activities will not be accepted if they are not associated with these essential quality requirements. Designed Business Quality (DBQ) represents the set of quality attributes/characteristics or sub characteristics that manifest how a given organisation choose to achieve an EBQ. For example, if the organisation in the business of banking, then Security is its EBQ therefore, the business activities will not be accepted or taken if this quality requirement is not associated with the *Account* as an EBE and UoW afterwards. But, *Confidentiality* and *Authenticity* are considered as DBQ that forms a way the organisation may wish to choose in delivering security to the UoW *Account*. Another example, if the business of all organisation is in the administration of a faculty, where *Coursework Assessment* is EBE and becomes a UoW afterwards. Now, if this business entity did not deliver its activities taking into account *Timeliness/Efficiency* and *Accuracy* quality requirements as EBQs, then this entity must reconsider its activity again, otherwise it might fail the business the organisation is in. Hence, these EBQs are critical to survive the business and its competitiveness among others in the market.

Step 3: Filtering the EBEs into UoWs and Elaborating the EBQs into Q-UoWs

In this step the footsteps of the current Riva method followed in filtering the already identified EBEs that have an interesting life time into UoWs, as has been already learned in Section 2.3.1 Accordingly, the current filters are applied for the identification of the UoWs (Ould, 2006). However, a further filter is proposed to address a particular identified goal (i.e., an EBE) in the agreed boundary possessing a critical lifetime. This is because, by default, business activities within the agreed boundary are prepared, operated and managed in order to achieve a goal. Hence, the EBE that represents a goal in the business of the organisation is by default a UoW. In addition, the entire UoWs in the diagram stem from the goals of the business that the organisation is in, whether the UoWs are explicitly considered as goals or implicitly related to these.

The filtering notion is used in order to obtain the UoWs, and the elaboration notion is proposed in order to obtain the Quality on a UoW (Q-UoW). It is necessary to obtain the UoWs first and then to characterise them with their associated attributes. Simply, you cannot characterise an object with its desired attributes if it does not exist. If an EBQ is found as a desired quality attribute that characterises a UoW, considering that a UoW encapsulates a CP and a CMP, then the EBQ must be elaborated into a Q-UoW. Since the organisation is interested in the lifetime of the UoWs, then it must be interested in the lifetime of addressing their desired quality attributes or requirements that are represented in the form Q-UoWs. Recalling the example of banking, assume that the UoW *Account* is characterised with the desired EBQs *Responsiveness, Empathy and Security*. Then the three EBQs are elaborated into three Q-UoWs, respectively in relation to their UoW *Account*. The Q-UoW is related to its UoW using the *constrain* relationships regardless of the type of the Q-UoW. In particular, the type of a Q-UoW is one out of two. It is either an elaborative and designable quality requirement (i.e., designed using quality-oriented models such as the NFR framework) or not. The elaborative Q-UoW is represented using a grey hexagon that includes the Q-UoW label and the reference to the quality model between brackets. The not elaborative one is represented using a dotted hexagon and with no quality model reference. For example, the Q-UoW *Security* is classified as an elaborative and attached to the UoW *Account.* Hence, the quality-oriented model of Security must be designed with respect to the UoW *Account*, as it is considered as an elaborative Q-UoW. The *Responsiveness* and *Empathy* Q-UoWs for the *Account* are examples of the not elaborative Q-UoW. Therefore, no quality-oriented model is required to associate the individual UoW *Account*. Figure 5.5 depicts the Q-UoWs with respect to the *Account* (i.e., prior the red arrow). In this figure, the quality-oriented model in the Security Q-UoW is referenced with a not real reference number. However, Figure 4.13 in Chapter 4 is recommended as a good illustration for this example for the purpose of showing an example of the quality model for a Q-UoW. The diagram in Figure 5.5 prior to the red arrow is the illustration of the GQ-UoW diagram from the point of view of the individual bank account UoW. This diagram will be encapsulated into one UoW for the sketching of the UoW diagram. In short, the business architect is at charge to study and analyse each individual G-UoW regarding its associated Q-UoWs.

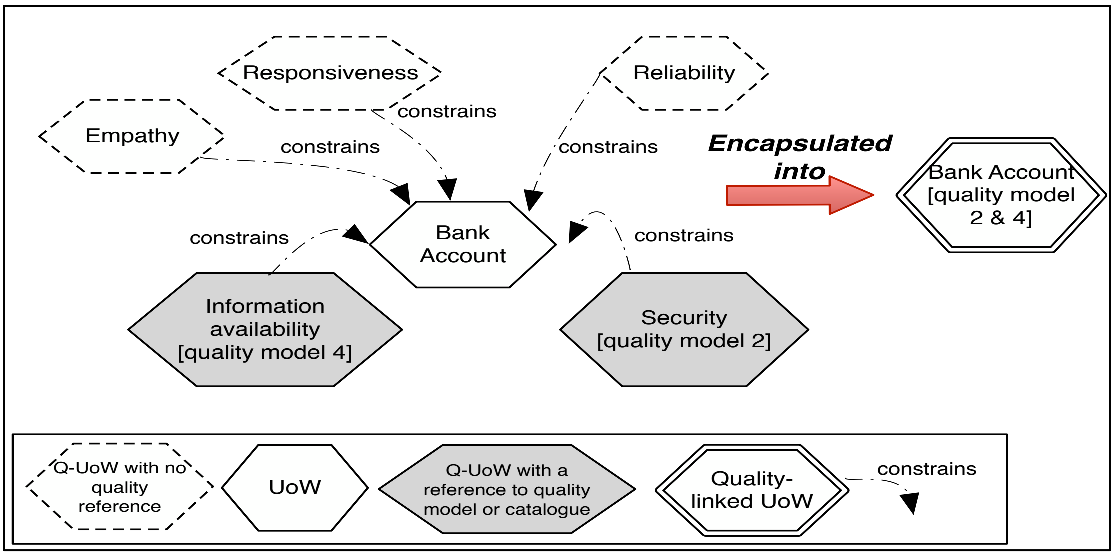


Figure 5.5: An Example for Encapsulating Q-UoWs into the G-UoWs.

Step 4: Sketching the GQ-UoW Diagram

In this step, the analyst works to encapsulate the diagram of each individual G-UoW into one GQ-UoW (i.e., represented using a double hexagon with brackets that unite the quality model references) as shown in Figure 5.5. The BP architect may face a set of Q-UoWs that are not elaborative and they all constrain a particular UoW. In this case, they are encapsulated in order to generate a GQ-UoW that is represented using the double hexagon but without the brackets. The encapsulation is conducted in order to sketch the original UoW diagram proposed by Ould, which is refined in this research into the GQ-UoW diagram. The resulted GQ-UoWs from the encapsulation are related using the ordinary generate dynamic relationships as in (Ould, 2006).

Figure 5.6 illustrates a simple UoW diagram for the example of the organisation that is in the banking business. The top UoW diagram is designed using the Riva method (Ould, 2006), where the bottom one is designed using the proposed guidelines above.

For some reason, it depends on the BP architect whether s/he wish to encapsulate the Q-UoWs and the UoW into GQ-UoW or not. In fact, the architect may wish not to conduct the encapsulation in case there are few UoWs and associated Q-UoWs that both can be clearly presented and well fit into one GQ-UoW diagram.

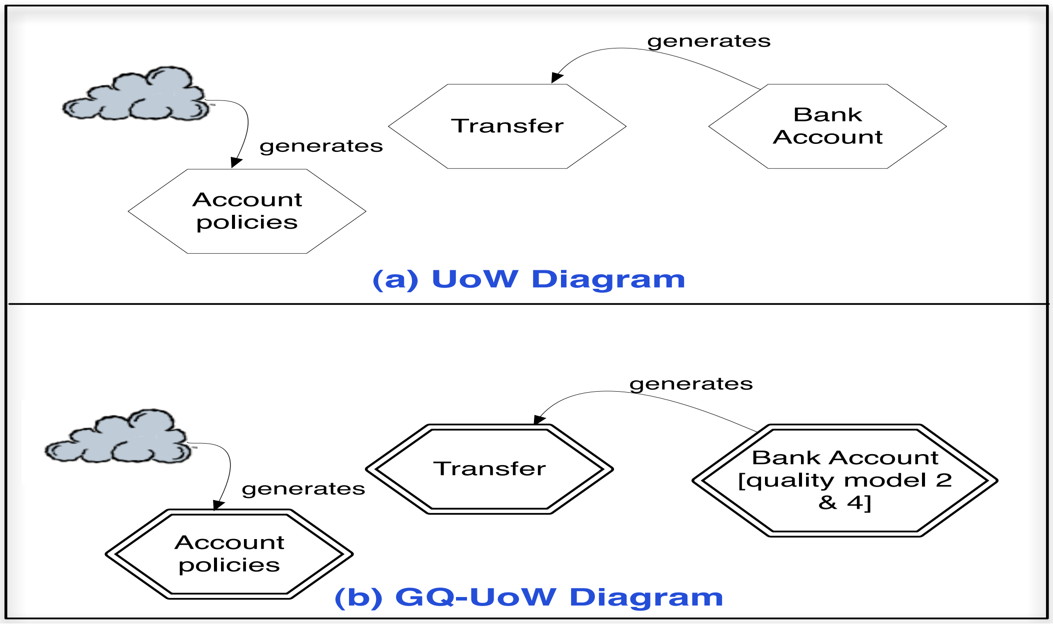


Figure 5.6: The Sketching Differences between the Original UoW Diagram and the GQ-UoW Diagram.

Step 5: Translating the GQ-UoW diagram into the 1st cut process architecture

Step 6: Applying the Riva Heuristics

Figure 5.3 shows that both steps are carried out as if it is the regular UoW diagram is translated into 1st cut process architecture using the original Riva guidelines (Ould, 2006).

Step 7: Generating the 2nd cut process architecture

The 2nd process architecture (i.e., Riva BPA) is generated using the original method (Ould, 2006) However, the architect should take into account the association of the Riva BPA with a table where each record/entry in the table is uniquely allocated for a particular process in the BPA as shown in Table 5.1. Each entry of a process presents the goal of the process, which is usually the title of the process, the goal that the process contributes to and the associated quality requirements along with their references, if applicable. The first and the second are called the Goals of the Process (GoP), where the third is called as the Quality of the Process (QoP).

The given information in the table must be associated with the 2nd cut Riva BPA in order to have them ahead before proceeding with the detailed design of the workflow of the business processes. That is, the BP designer is aware ahead about the process's goal(s), the parent goal that the process contributes to and the associated quality requirements (e.g., using the BSV) and thereby the design is anticipated to be consistent with the respect to the three aforementioned perspectives.

Table 5.3: The BP Description Table with regard to its Goals and Quality Requirements Within the Riva BPA.

Business process number	Business process name	Goals of the Process (GoP)	Quality of the Process (QoP)

5.6 The Goal-based and Quality-Linked Business Process Model: BPMN Process Modelling Language Example

Although the original Riva method is refined as was presented in the previous section, it still requires the association of the detailed representation of the workflow of the processes that set the final scene of the architecture taking into the account addressing the consistency of the information in the BPA. Thus, the associated BPMs must be aligned to their architecture.

In short words, an extension of the BPM is proposed, which is modeled using the example of the BPMN language, by explicitly establishing a BP appendix that is represented as a box below the BPM. The function of the designed appendix is to code and alert the missing, yet required information. Although some information of the BP is required, still it cannot be represented using the notion of the workflow (e.g., goals, quality requirements, risks, alignment-level, etc.). Therefore, the appendix attached below the BPM is required for particular situations that are difficult or ones that cannot permit the designer representing the required information using the notion of the workflow. One situation is when the required information is abstract (e.g., goal or the position of the process in any conducted alignment). Another situation is when the required information is complex or elaborated using another language or approach (e.g., the NFR framework as a required quality model representation). In this situation, the appendix works as a linker for further related and required knowledge for the process. If any element in the further information could be represented within the flow of the work then, it must be within the body of the BP workflow. Such an example is the research work in (Aburub, 2006) that improved the BPM using the NFR framework but without attaching the appendix.

With regard to goals and quality requirements, they are integrated and presented using the proposed appendix. Simply, an extension in the BPM is proposed through attaching its appendix that includes the explicit notations of goals and quality requirements. The former is explicitly represented using the oval notation, where the latter is represented using the cloud notation (i.e., the two notations are inspired from their common representation found in the literature and in Chapter 4). Figure 5.7 depicts a simple example of a BP modelled using the BPMN language along with its appendix for the integration of goals and quality requirements (i.e., soft goals). In its appendix, the bracket in the text of the soft goal part indicates the

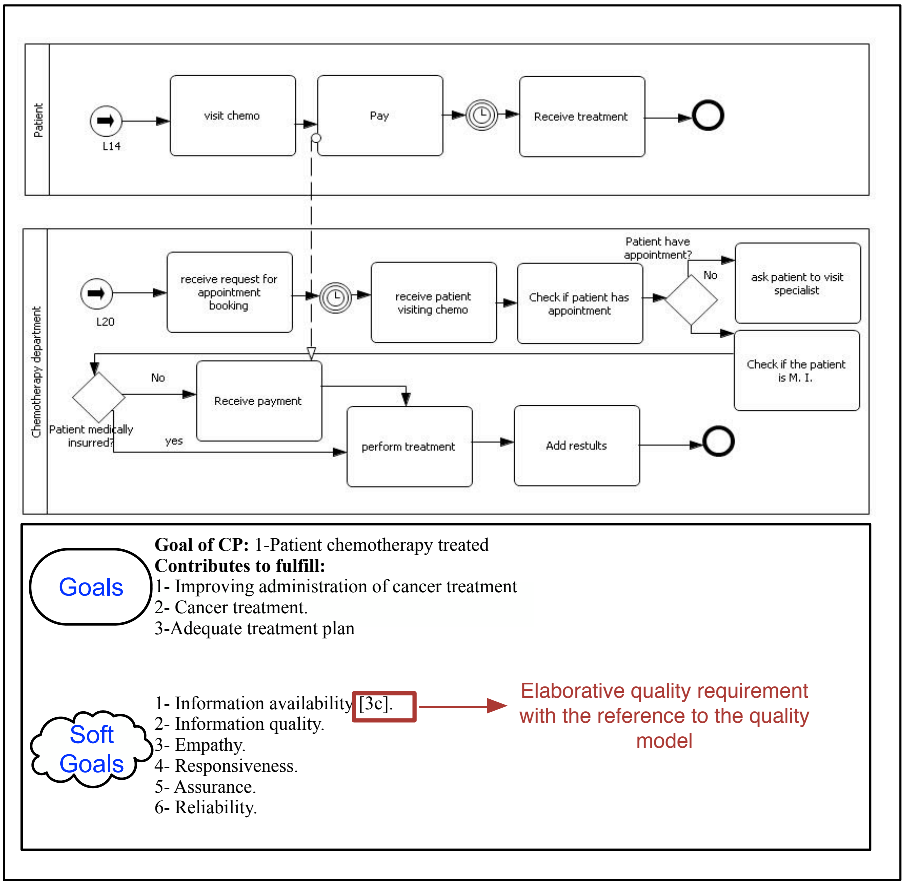


Figure 5.7: Handle a Patient Chemotherapy Treatment BP Example Represented Using the GQ-BPMN.

reference to further attached information that is the quality model of the information availability. In fact, the information given for the goals and soft goals must be derived from their architecture in order to follow the claim regarding addressing consistency between the BPAs and their architecture.

Finally, the notion of the BP appendix is not actually proposed for the BPMN BPs, yet it is anticipated to accommodate most BP modelling languages (e.g., RAD, UML activity diagram, FBMPL (Chen-Burger et al, 2002)).

5.7 The GQ-BPAOnt Ontology

The GQ-BPAOnt ontology is essentially based on the reuse and extension of the BPAOnt ontology (Yousef, 2010) using the refined Riva BPA method, namely the GQ-Riva BPA presented in Section 5.5. The GQ-BPAOnt ontology semantically blueprints the essential elements of the GQ-Riva-based BPA and the associated BPMs for a particular organisation using ontologies. The GQ-Riva BPA forms the backbone of the GQ-BPAOnt (i.e., the GQ case processes and the GQ case management processes along with the relationships between them) where each process in the BPA is elaborated into a corresponding BPM that is formally represented using the refined BPMN modelling (SUPER, 2008b) as was previously explained in Section 5.6. Since the srBPA ontology is refined, the sBPMN ontology should be consistent with the refinements embedded in the former ontology. Therefore, the original sBPMN ontology from the SUPER project is lightly refined for the purpose of addressing the consistency with the GQ-srBPA ontology.

During the development of the BPAOnt ontology in the BPAOntoSOA framework, Yousef merged the original srBPAOnt ontology and the borrowed sBPMN ontology (Yousef, 2010). Similarly, the GQ-BPAOnt reuses the original merging rules for the two refined ontologies in order to align the GQ-srBPA and GQ-sBPMN ontologies. The GQ-srBPA ontology is a refined ontology that encapsulates the srBPA ontology (i.e., considered as an original product within the BPAOntoSOA framework). Therefore, the original concepts or classes in the srBPA ontology are refined (e.g., refining the EBE to G_EBE) and extended (e.g., designing the new class EBQ) in order to ontologise the refined Riva method presented in Section 5.5.

This section aims to present the GQ-BPAOnt that stems from the original BPAOnt ontology and describe the components of the second layer of the new framework. It is necessary to highlight that the work of this section is presented without considering the alignment approach component (i.e., appearing in the left hand side in Figure 5.2), as this will be discussed in the next section. This alignment component is designed in order to address a consistent bridging between the BSV, Riva BPA and BPMs. Accordingly, section 5.7.1 presents the redevelopment of the srBPA ontology. Section 5.7.2 depicts the description of the GQ-srBPA ontology instantiator. These two sections are presented without considering the alignment approach too. In section 5.7.3, the sBPMN ontology of the SUPER project reused from the original Yousef's framework is refined and redeveloped in order to address the function of GQ-BPAOnt ontology instantiation layer. In addition, Section 5.7.4 presents the role of the sBPMN ontology instantiator component within this layer. Finally, this section is closed through presenting the merging rules of the two ontologies in Section 5.7.5.

5.7.1 The GQ-srBPA Ontology: A Goal-based and Quality-linked Redevelopment of the Original srBPA Ontology

In this section, the GQ-srBPA ontology conceptualises the refined Riva method (i.e., GQ-Riva) for the BPA modelling. The conceptualisation embodies the concepts that are represented using classes and the relationships between them. The classes are characterised using the relevant attributes. In addition, the redevelopment of the GQ-srBPA ontology is supported with the description of its classes using OWL restrictions.

5.7.1.1 The GQ-srBPA Ontology Language and Development Tool

Recalling the work of Chapter 4 Section 4.3.2, the GQ-srBPA ontology is similarly developed using the Protégé 3.4.1 tool and the OWL-DL language that supports the identification and the description of each class. The SWRL rules facility supports ontologising the Riva method steps and their execution using the Jess engine.

5.7.1.2 The GQ-srBPA Ontology Design Decisions

Recalling the Figure 5.3 and Figure 5.4, the **new classes** that extend the original design of the srBPA ontology are the EBQ, the Q-UoW, and the Constrain. Each process in the GQ 2nd cut architecture (i.e., GQ-Riva BPA model) must associate its goals and soft goals (i.e., the quality requirements). Accordingly, two classes are added in order to extend the number of classes that constitute the Riva objects and that are the GoP and QoP. Each of them must be related to a CP and a CMP. In addition, the class Quality_Model_Reference is designed in order to optionally represent the reference to the desired quality model.

The refined outcomes within the Riva process in Figure 5.3 influence the refinements of the current classes in the srBPA ontology. On the one hand, few pre-existing classes in the srBPA ontology have been refined from UOW, EBE, UoW_Diagram (Yousef, 2010) and the PA_2nd_cut_Diagram to G_EBE, G_UOW, GQ_UoW_Diagram and GQ_PA_2nd_cut_Diagram, respectively. On the other hand, few pre-existing classes in the srBPA ontology that represent the relations within the Riva method remain as they are without any required refinement (e.g., Generate, Outside_world, Start, Deliver and Request).

The classes in the GQ-srBPA ontology are related to each other using the pre-existing and the new object properties. In addition, the pre-existing Boolean properties are reused (e.g., the isUoW and the isActive Boolean properties). The isElaborative Boolean property joins the current ones and it allows the analyst to decide whether the desired quality requirement is elaborative or not. If it is elaborative, then it must have an associated Quality_Model_Reference instance.

The current SWRL rules within the srBPA ontology are reused where new ones are created in order to address the new steps in the Riva method. The SWRL rules allows designing each Riva step, where the body of the rule is filled with the relevant OWL-DL classes and properties.

5.7.1.3 The GQ-srBPA Ontology Classes and Properties

Since the original srBPA ontology forms the foundation of the GQ-srBPA ontology, then the original classes are reused and extended along with their associated properties. Appendix H presents the classes and properties that set up the GQ-srBPA ontology.

5.7.1.4 Ontologising the Refined Riva Method

In this section, the refined Riva method presented in Section 5.5 is ontologised and described using SWRL rules. Each rule is executable and refers to a particular step in the refined method. Since the ontology of the original method is developed, then a redevelopment is required in order to adjust the role of the component with the proposed refinements. Therefore, the original OWL-DL descriptions of classes along with the SWRL rules have been redeveloped with respect to the integrating goals and quality requirements. The work of this section is attached in Appendix I.

5.7.2. The GQ-srBPA Ontology Instantiator

The function of the GQ-srBPA ontology instantiator component extends its corresponding original function (i.e., regarding the srBPA ontology instantiator in the BPAOntoSOA framework (Yousef, 2010)) through formally representing the instantiation of the refined Riva BPA for an organisation, which is a goal-based and quality linked BPA. The formal representation is conducted using the OWL-DL ontology language and SWRL rules. In particular, the ontology of the original component (i.e., the srBPA ontology developed by (Yousef, 2010)) is redeveloped from the perspective of the Riva method and its conceptualisation.

SWRL rules aid the designer in implementing the Riva method steps. These SWRL rules are borrowed from (Yousef, 2010) and extended in order to address the function of the component in producing the instantiation of the GQ 2^{nd} cut architecture for a particular organisation. A particular SWRL rule is executed after activating the Jess engine. By running the Jess engine, a number of the mapped classes and their instances are generated.

5.7.3 The GQ-sBPMN Ontology: A Goal-based and Quality-linked Redevelopment of the Original sBPMN Ontology

Since the original Riva method required the association of the detailed representation of the BPs (i.e., CPs and CMPs) that constitute the BPA, the sBPMN ontology was borrowed by Yousef (Yousef, 2010) from (SUPER, 2008) in order to address this requirement. Accordingly, the refined Riva method still requires the detailed representation of the flow of work encapsulated in the BPMs, but taking into the account their consistency with the proposed refinements regarding the integration of goals and quality requirements.

In fact, the current sBPMN ontology does not reflect or reveal the embodiment of the process goals and/or desired quality attributes. The primary reason behind this lack is owing to the actual absence of implicit representation or explicit notations of the goals and quality requirements of a process modeled using the BPMN. Therefore, the refinements presented in Section 5.6 are implemented in the current ontology of the sBPMN ontology. However, the proposed graphical elements of (i.e., goal and soft goal) are not recognized in the current BPMN 2.0. Therefore, the two are anticipated not being translated into their corresponding XPDL 2.0 specification.

5.7.3.1 the Design Decisions of the GQ-sBPMN Ontology

Essentially, the sBPMN ontology is the foundation for the GQ-sBPMN ontology. In fact, the sBPMN ontology is entirely borrowed and extended regarding the integration of goals and quality requirements of the process using the notion of the appendix presented in Section 5.6. Hence, the design decisions carried out in a lightweight manner by identifying the class Appendix as a sub class of the Supporting Types. In addition, the classes Goal and Soft_Goal are created as sub of the Artifact class that was designed to express any additional information about the process. The proposed three classes are circled with red within the original design as shown in Figure 5.8 captured using the OWL Viz tab. By following the footsteps in (Yousef, 2010), no SWRL rules are required for implementation due to unneeded structured steps for generating a BPM. A design of a BPM is based on instantiating concepts and connecting them with relationships only.

5.7.4 The GQ-sBPMN Ontology Instantiator

The function of this component originally stems from the function of the sBPMN ontology instantiator by formally representing the organisation's BPMN process models using the OWL-DL ontology language (Yousef, 2010). However, a lightweight change is proposed regarding the input and the output of the component through involving the goals and the quality requirements. The input is the goal-based and quality linked BPMN process models for an organisation and the refined sBPMN ontology (i.e., GQ-sBPMN ontology) where the output is the instantiation of the formal representation of the GQ-BPMN process models.

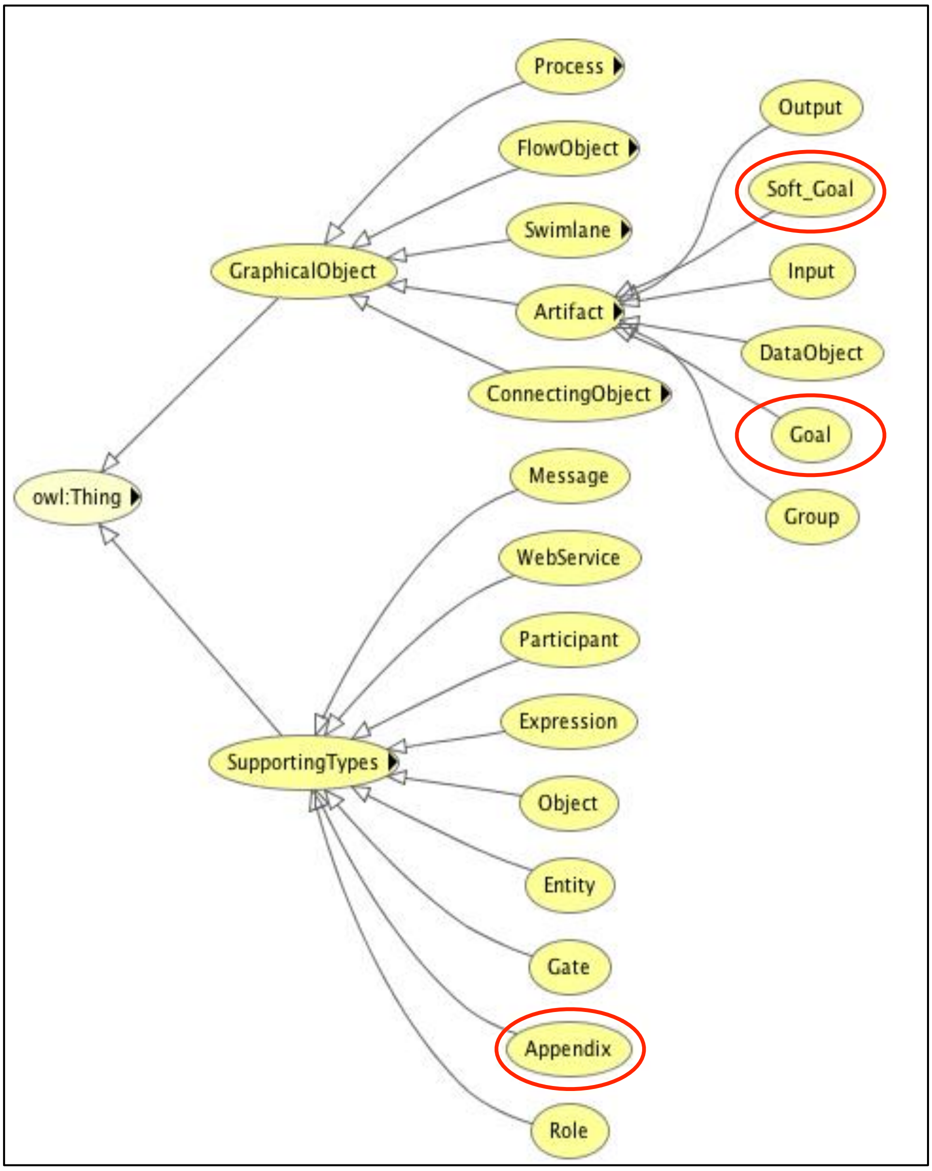


Figure 5.8: Part of the GQ-sBPMN Ontology Hierarchy

The footsteps of the work in (Yousef, 2010) are followed regarding the necessity of translating the BPMN models into XPDL first then running a rule in the SUPER project in order to automatically instantiate the GQ-sBPMN ontology.

5.7.5 The GQ-srBPA GQ-sBPMN Ontology Merger

The component carries out the original function of merging the two ontologies (i.e., the GQ-srBPA and the GQ-sBPMN) (Yousef, 2010) using the ontology importing facility provided by the Protégé tool. However, the integration of goals and quality requirements stimulated further merging rules. In particular, the refined Rive BPA (i.e., the GQ 2nd cut process architecture) entails the association of the GoP and the QoP elements for each CP and/or CMP that constitutes the BPA. Where the refined GQ-sBPMN encapsulates the two elements using the Goal and the Soft_Goal classes, respectively, considering the original fact that each CP and/or CMP is a process in the GQ-sBPMN. Accordingly, a business process can be extracted along with its goals and soft goals. The input into this component is the instantiated GQ-srBPA ontology for the given organisation with its associated instantiation of the GQ-sBPMN ontology. The output is the GQ-BPAOnt that refers to the formal instantiation of the two input integrated ontologies.

The merging rules for this component are derived from the refined Riva method presented in Section 5.5. The first rule (i.e., stemming from the original merger component regardless the proposed extension) states that for each process in the BPA, there is a corresponding process in the sBPMN. The second and the third merging rules are new and derived from the proposed extension to the Riva and the BPMN. The second rule states that for each process in the GQ-Riva BPA, must be uniquely defined using the GoP that is mapped to a Goal in the GQ-sBPMN. With regard to the quality perspective, the third merging rule states that for each process in the GQ-Riva BPA that is characterised with the desired quality requirements using the QoP element, there is a corresponding soft goal for the process in the GQ-sBPMN.

Accordingly, three functional object properties are defined as merging rules, where the first is borrowed from the original component, and the rest are newly established taking into the account that the domain are classes from GQ-srBPA ontology, where their range is classes from GQ-sBPMN ontology. The first merging rule (i.e., hasCorrespondingBPM) has the domain as the union of two classes CP U CMP where its range is the class Process. The second merging rule has the domain GOP class, where the range is Goal class. In this rule, the domain is related to the range using the object property hasCorrespondingBPMGoal. The third merging rule has the domain QOP class, where the range is Soft_Goal via the relation hasCorrespondingBPMSG. These three rules are:

1) CP: ∀ hasCorrespondingBPM only Process.

   CMP: ∀ hasCorrespondingBPM only Process.

2) GOP: ∀ hasCorrespondingBPMGoal only Goal.

3) QOP: ∀ hasCorrespondingBPMSG only Soft_Goal.

It is necessary to highlight that the merging is proposed regarding the mapping of concepts within the Riva method and the BPMN, but not the relationships between the concepts. For example, the merging rules in the GQ-srBPA GQ-sBPMN merger component are not proposed for mapping the relation generate in the GQ-UoW diagram to another concept or relation.

5.8 Aligning a Pre-existing Riva-based BPA to the BSV of an Enterprise

This section aims at presenting an alignment of a pre-existing Riva-based BPA to the BSV of an enterprise, which was previously addressed in the work of Chapter 4. However, the notion of proposed alignment is interpreted into two meanings, as shown in Figure 5.9. It is apparent that the two meanings have in common the input of the BSV of an organisation as a pre-alignment required input and the to-be GQ Riva BPA for the given organisation as the desired output. The necessity beyond conducting the alignment is to address the up-to-date identified goals and soft goals within the BSV in order to keep the harmony and the consistency link between the operating BPs in the organisation via their architecture first and then their flow of work afterwards.

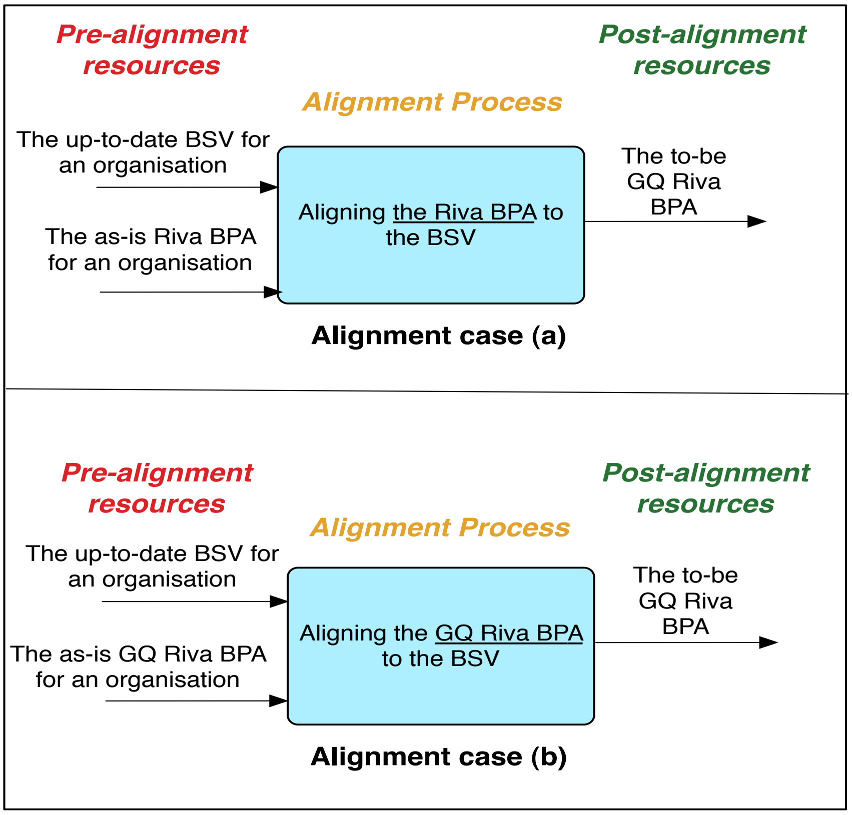


Figure 5.9: Two proposed Alignment Approaches.

The **pre-alignment resources** refer to the resources and knowledge treated as the input for the alignment, where the **post-alignment resources** refer to the resulted resources and knowledge after conducting the desired alignment. The pre-alignment and the post-alignment resources are denoted by the terms **as-is** and **to-be,** respectively in order to shorten the expression. The up-to-date BSV refers to the BSV for the given organisation that incorporates latest development and trends in the domain from the point of view of goals and soft goals. Once the required Riva BPA is properly aligned to the BSV, the BPA is anticipated to be well defined and its BPs are exploited using the GQ-sBPMN.

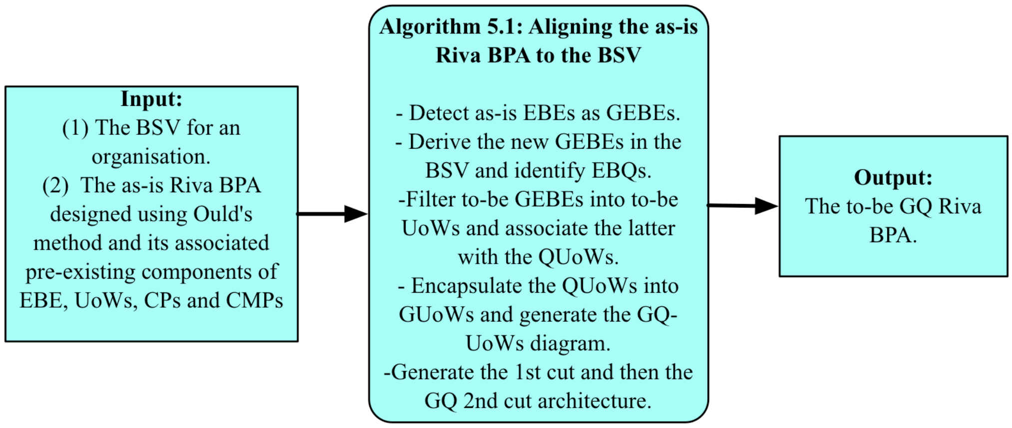
In fact, the two meanings are different in the required input of the Riva BPA, yet they still accommodate the notion of the alignment. In Figure 5.9 (a), the first meaning is depicted as the alignment case (a) that requires the EBE-driven Riva BPA as its input along with the BSV in order to result the GQ Riva-based BPA. In this situation, the pre-existing Riva BPA (i.e., as-is Riva BPA) is considered as EBE-driven, and thereby it is designed using the original Riva method proposed by Ould (Ould, 2006). Where the second meaning is depicted in Figure 5.9 (b) that considers its input the BSV and the pre-existing Riva BPA as Goal-driven (i.e., designed using the refined method presented in Section 5.5).

With regard to the first alignment case, which is case (a) in Figure 5.9, the alignment process is anticipated requiring higher effort from the architect, as the means of goals and quality are originally absent. Therefore, this requires effort for engaging the two means into the as-is Riva BPA in order to derive the to-be GQ Riva BPA. For example, the EBQs do not already exist as part of the pre-aligned resources and hence their construction requires higher effort than if they were already existed and reused. This alignment is implemented in the algorithm shown in Figures 5.10-5.12.

With regard to the second alignment case in Figure 5.9 (b), the alignment process is anticipated to require less effort than the first case. This is because the original concepts of goals and quality are already integrated in the as-is GQ Riva BPA, yet they are reused and refined rather than being newly engaged. For example, the as-is EBQs in the pre-aligned resources are refined and not constructed from scratch. This alignment is implemented in the algorithm shown in Figures 5.13-5.16.

Accordingly, the architect has the freedom to choose among the two alignment options based on the needs and the available pre-existing inputs.

Algorithm 5.1: Aligning the as-is Riva BPA, designed using the original approach, to the BSV of a Particular Business Organisation.



Description: This algorithm is designed in order to illustrate the first alignment case that requires the BSV and the pre-existing Riva BPA designed using the original approach (Ould, 2006) for the given business organisation. The output is the goal-based and quality-linked Riva BPA. This algorithm is based on detecting the pre-existing yet still required components, instead of deriving them, and deriving the new components that do not exist in the pre-existing Riva BPA input.

Input: (1) *The business strategy view components (e.g., HBG, IH_SG, Goal, Actor, HSD actor, etc), BSV_Components={$bsvc_0$, $bsvc_1$, ..., $bsvc_n$}*
(2) *The pre-existing Riva BPA that is designed using the original Riva method and its associated components as-is essential business entities, EBE={$aebe_0$, $aebe_1$, ..., $aebe_z$}, as-is unit of works, UOW={$auow_0$, $auow_1$, ..., $auow_p$} and the as-is processes that include the case processes and the case management processes, Process={ap_0, ap_1, ..., ap_m}*

Output: *The goal-based and quality-linked Riva BPA (i.e., designed using the refined Riva approach) that comprises of the to-be (i.e., goal-based) EBEs={$tbebe_0$, $tbebe_1$, ..., $tbebe_w$}, essential business qualities, EBQ={ebq_0, ebq_1, ..., ebq_t}, to-be GQ_UoWs={$tbgquow_0$, $tbgquow_1$, ..., $tbgquow_c$} relations between GQ_UoWs, GQ_UoW_REL={ur_0, ur_1, ..., ur_f} and to-be GQ Processes={gqp_0, gqp_1, ..., gqp_o}*

Algorithm:

Begin

Identify the set of components that exist in the business strategy view for an organisation, BSV_Component={$bsvc_0$, $bsvc_1$, ..., $bsvc_i$, ..., $bsvc_n$} $0 \leq i \leq n$;
Identify the set of the as-is essential business entities, EBE={$aebe_0$, $aebe_1$, ...,$aebe_u$,..., $aebe_z$},$0 \leq u \leq z$;
*Define the **Detected_EBEs, Detected_UoWs** and **Detected_Processes** lists that include the EBEs, UoWs and the processes detected in the BSV using the as-is EBEs, as-is UoWs and the as-is processes respectively.*
*Define the **New_EBEs, New_GUoWs** and the **New_Processes** lists that include the EBEs, the UoWs and the processes derived from the BSV, where they do not originally exist in the as-is EBEs.*

Figure 5.10: Algorithm the Alignment of the as0is Riva BPA to the BSV (Part 1 of 3).

Identify the set of the essential business qualities, EBQs={ebq0, ebq1, …., ebq_v, …, ebq_t}, $0 \le v \le t$;

//Detecting current EBEs, new EBEs and identifying EBQs

For each component $bsvc_i$ in the business strategy view in BSV_Component do the following

*If the $bsvc_i$ is EBE **and** exists in the as-is EBE list*

Insert the $bsvc_i$ in the Detected_EBEs list;

*Else if the $bsvc_i$ is EBE **and** do not exist in the as-is EBE list*

Insert the $bsvc_i$ in the New_EBEs list;

Else if the $bsvc_i$ is EBQ

Insert the $bsvc_i$ in the EBQ set;

End if else bsvc

End for each BSV component

Identify the set of the to-be essential business entities that represents the union of the detected EBEs and the new EBEs, to-be_EBEs= {$tbebe_0$, $tbebe_1$, …, $tbebe_q$, …, $tbebe_w$}, $0 \le q \le w$;

//Detecting current UoWs and identifying new UoWs

For each *$tbebe_q$ in the to-be EBEs do the following*

*If the $tbebe_q$ is UoW **and** exists in the as-is UoW list*

Insert the $tbebe_q$ in the Detected_UoWs list;

*Else if the $tbebe_q$ is UoW **and** do not exist in the as-is UoW list*

Insert the $tbebe_q$ in the New_UoWs list;

End if else

End for each to-be EBE

Identify the set of the to-be unit of works that represents the union of the detected UoWs and the new UoWs, to-be_GUoW= = {$tbuow_0$, $tbuow_1$, …, $tbuow_e$, …, $tbuow_r$}, *$0 \le e \le r$;*

//Constraining the G_UoWs with associated Q_UoWs

For each $tbuow_e$ in the to-be UoWs do the following

Identify the quality set within the unit of work $tbuow_e$, Q_UoW={$quow_0$, $quow_1$, …, $quow_a$, …, $quow_s$}, $0 \le a \le s$;

Identify the set of the constrains relationships between the $tbuow_e$ and its associated Q_UoW set, Constrain_Rel={cr_0, cr_1, …, cr_d, …, cr_j}, $0 \le d \le j$;

For each ebq_v in the EBQ set

If the ebq_v constrains the $tbuow_e$

Elaborate the ebq_v into $quow_a$ for the $tbuow_e$;

If the $quow_a$ has associated quality model reference

Attach the reference with the quow;

End if

Draw a constrain relationship cr_d from $quow_a$ to $tbuow_e$

End if

End for each constrain EBQ

Figure 5.11: Algorithm the Alignment of the as-is Riva BPA to the BSV (Part 2 of 3).

//Encapsulating the Q_UoWs into their G_UoWs in order to generate the GQ UoWs
Identify the set of the goal-based and quality-linked to-be unit of works that represents the encapsulated to-be_ GUoWs, to-be GQ_UoWs = {$tbgquow_0$, $tbgquow_1$, ..., $tbgquow_b$, ..., $tbgquow_c$}, $0 \le b \le c$;

For each $tbuow_e$ *in the to-be GUoWs do the following*
Encapsulate the associated Q_UoWs of $tbuow_e$ *into one to-be GQ_UoW* $tbgquow_b$;
End for GUoWs

//Drawing
Draw the GQ UoW diagram using the GQ_UoWs set;
Translate the GQ-UoW diagram into the to-be 1st cut architecture;
Translate the to-be 2nd cut architecture from the 1st cut architecture;

//Identifying the goals and the quality for each process in the generated 2nd cut BPA;

Identify the set of the to-be goal-based and quality-linked processes in the generated 2nd cut architecture from the alignment, GQ_Process={gqp_0, gqp_1, ...,gqp_x, ..., gqp_o}, $0 \le x \le o$;

Recall the goal-network in the BSV input;

For each identified process in generated 2nd cut architecture, gqp_x, *do the following*
Consider the name of the gqp_x *as the goal of the process;*
Use the goal-network in the BSV in determining the contribution of the gqp_x;
Recall the original GQ_UoW, $tbgquow_b$, *that generated the* gqp_x;
De-capsulate the original $tbgquow_b$ and allocate the desired quality requirements that constrain the gqp_x, along with their quality models if required;
End for each gqp

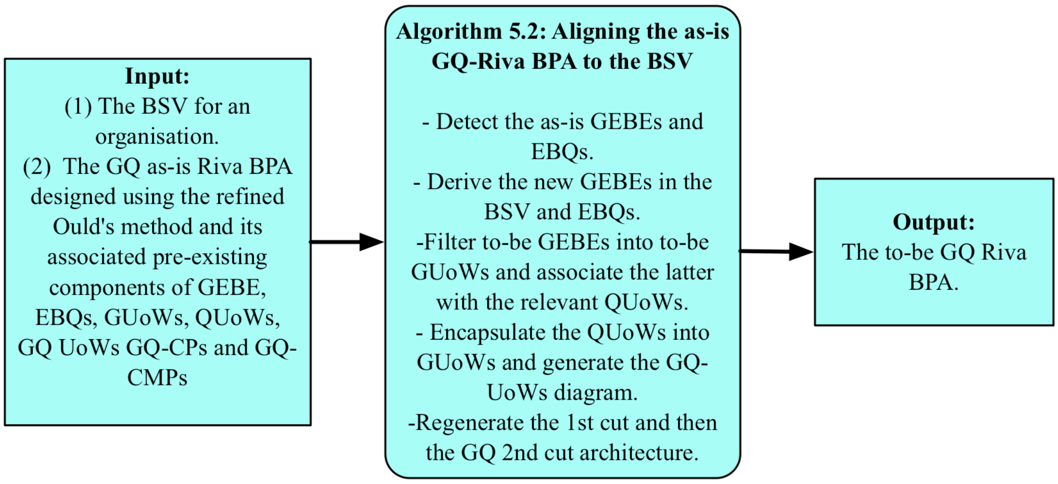
//Detecting processes and identifying new processes along with the alignment-level

Let Alignment_Level be the variable for each GQ_Process instance and has one out of three values: novel process, as-is redesigned process or as-is with no required redesign.
For each identified process in generated 2nd cut architecture, gqp_x, *do the following*
If the gqp_x *exists in the as-is processes list*
If the gqp_x *requires redesign*
Flag the Alignment_Level of gqp_x *as" as-is redesigned process";*
Else if the gqp_x *does not require redesign*
Flag the Alignment_Level of gqp_x *as" as-is process with no redesign";*
End if else
Insert gqp_x *in the Detected_Prcesses list;*
Else if the gqp_x *does not exist in the as-is processes list*
Flag the Alignment_Level of gqp_x *as" novel process";*
Insert gqp_x *in the New_Prcesses list*
End if else
End for each gqp

Turn the GQ_Process={gqp_0, gqp_1, ...,gqp_x, ..., gqp_o} $0 \le x \le o$, *into to-be GQ_Processes that represent the union of the detected processes and the new processes.*
End

Figure 5.12: Algorithm the Alignment of the as-is Riva BPA to the BSV (Part 3 of 3).

Algorithm 5.2: Aligning the as-is GQ Riva BPA, designed using the Goal-based and Quality-linked Riva Approach, to the BSV of a Particular Business Organisation.



Description: This algorithm is designed in order to illustrate the second alignment case that requires the BSV and the pre-existing Riva BPA designed using the refined approach (i.e., goal-based and quality-linked) for the given business organisation. The output is the to-be goal-based and quality-linked Riva BPA. This algorithm is based on detecting the pre-existing yet still required components, instead of deriving them. No effort is carried out in identifying new concepts

Input: (1) *The business strategy view components (e.g., HBG, IH_SG, Goal, Actor, HSD actor, etc), BSV_Components={$bsvc_0$, $bsvc_1$, …, $bsvc_n$}* (2) *The pre-existing Riva BPA that is designed using the goal-based and quality-linked Riva method and its associated components as-is essential business entities, asis_EBE={$aebe_0$, $aebe_1$, …, $aebe_z$}, as-is essential business qualities, asis_EBQ={$aebq_0$, $aebq_1$, …, $aeb_{f''}$}, as-is unit of works, asis_GQ_UOW={$agquow_0$, $agquow_1$, …, $agquow_{p'}$} and the as-is processes that include the case processes and the case management processes, asis_GQ_Process={$agqp_0$, $gqap_1$, …, $agqp_m$}*

Output: *The to-be goal-based and quality-linked Riva BPA (i.e., designed using the refined Riva approach) that comprises of the to-be (i.e., goal-based) EBEs={$tbebe_0$, $tbebe_1$, …, $tbebe_w$}, to-be essential business qualities, EBQ={$tbebq_0$, $tbebq_1$, …, $tbebq_u$}, to-be GQ_UoWs={$tbgquow_0$, $tbgquow_1$, …, $tbgquow_c$} relations between GQ_UoWs, GQ_UoW_REL={ur_0, ur_1, …, ur_f} and to-be GQ Processes={gqp_0, gqp_1, …, gqp_o}*

Algorithm:

Begin

Identify the set of components that exist in the business strategy view for an organisation, BSV_Component={$bsvc_0$, $bsvc_1$, …, $bsvc_i$, …, $bsvc_n$} $0 \le i \le n$;

Identify the set of the as-is goal-based essential business entities, asis_ EBE={$aebe_0$, $aebe_1$, …,$aebe_u$,…, $aebe_z$},$0 \le u \le z$;

*Create the **Detected_EBEs, Detected_EBQs, Detected_GUoWs, Detected_QUoWs** and **Detected_Processes** lists that include the EBEs, UoWs and the processes detected in the BSV using the as-is goal-based EBEs, as-is GQ UoWs and the as-is GQ processes respectively.*

*Create the **New_EBEs, New_EBQs, New_GUoWs, New_QUoWs** and the **New_Processes** lists that include the EBEs, the UoWs and the processes derived from the BSV, where they do not originally exist in the as-is EBEs.*

Figure 5.13: Algorithm the Alignment of the as-is GQ Riva BPA to the BSV (Part 1 of 4).

Identify the set of the as-is essential business qualities, EBQs={aebq0, aebq1, …., $aebq_g$, …, $aebq_f$}, $0 \leq g \leq f$;

//Detecting current EBEs, new EBEs and identifying EBQs

For each component $bsvc_i$ in the business strategy view in BSV_Component do the following

*If the $bsvc_i$ is EBE **and** exists in the as-is EBE list*

Insert the $bsvc_i$ in the Detected_EBEs list;

*Else if the $bsvc_i$ is EBE **and** do not exist in the as-is EBE list*

Insert the $bsvc_i$ in the New_EBEs list;

*Else if the $bsvc_i$ is EBQ **and** exists in the as-is EBQ set*

Insert the $bsvc_i$ in the Detected_EBQ list;

*Else if the $bsvc_i$ is EBQ **and** does not exist in the as-is EBQ set*

Insert the $bsvc_i$ in the New_EBQ list;

End if else bsvc

End for each BSV component

Identify the set of the to-be essential business entities that represents the union of the detected EBEs and the new EBEs, to-be_EBE= {$tbebe_0$, $tbebe_1$, …, $tbebe_q$, …, $tbebe_w$}, $0 \leq q \leq w$;

//To-be EBEs= Detected EBEs U New EBEs;

Identify the set of the to-be essential business qualities that represents the union of the detected EBEs and the new EBE, to-be_EBQs={tbebq0, tbebq1, …., $tbebq_v$, …,$tbebq_t$}, $0 \leq v \leq t$;

// To-be EBQs= Detected EBQs U New EBQs

//Detecting current UoWs and identifying new UoWs

For each *$tbebe_q$ in the to-be EBEs do the following*

*If the $tbebe_q$ is UoW **and** exists in the as-is GQ-UoW list*

Insert the $tbebe_q$ in the Detected_GUoWs list;

*Else if the $tbebe_q$ is UoW **and** do not exist in the as-is GQ-UoW list*

Insert the $tbebe_q$ in the New_GUoWs list;

End if else

End for each to-be EBE

Identify the set of the to-be unit of works that represents the union of the detected G_UoWs and the new G_UoWs, to-be_GUoW= = {$tbuow_0$, $tbuow_1$, …, $tbuow_e$, ..., $tbuow_r$}, *$0 \leq e \leq r$;*

Figure 5.14: Algorithm the Alignment of the as-is GQ Riva BPA to the BSV (Part 2 of 4).

//Constraining the G_UoWs with associated Q_UoWs

For each tbuow$_e$ in the to-be UoWs do the following

Identify the as-is quality set within the unit of work tbuow$_e$, asis_Q_UoW={quow$_0$, quow$_1$, …, quow$_h$, …, quow$_{h'}$}, 0≤h≤h';

Identify the quality set within the unit of work tbuow$_e$, Q_UoW={quow$_0$, quow$_1$, …, quow$_a$, …, quow$_s$}, 0≤a≤s;

Identify the set of the constrains relationships between the tbuow$_e$ and its associated Q_UoW set, Constrain_Rel={cr$_0$, cr$_1$, …, cr$_d$, …, cr$_j$}, 0≤d≤j;

For each ebq$_v$ in the EBQ set

If the ebq$_v$ constrains the tbuow$_e$ and exists as *quow$_h$* in the asis_Q_UoW set

Insert the *quow$_h$* in the Detected_QUoWs list along with the constrain relationships to tbuow$_e$;

Else if *the ebq$_v$ constrains the* tbuow$_e$ and does not exist in the asis_Q_UoW set

Elaborate the ebq$_v$ into quow$_a$ for the tbuow$_e$;

Insert the *quow$_a$* in the Detected_QUoWs list;

If the quow$_a$ has associated quality model reference

Attach the reference with the quow$_a$;

End if

Draw a constrain relationship cr$_d$ from quow$_a$ to tbuow$_e$

End if

End for each constrain EBQ

//Re-Encapsulating the Q_UoWs into their G_UoWs in order to generate the GQ UoWs

Identify the set of the goal-based and quality-linked to-be unit of works that represents the encapsulated to-be_ GQUoWs, to-be GQ_UoWs = {tbgquow$_0$, tbgquow$_1$, …, tbgquow$_b$, …, tbgquow$_c$}, *0≤b≤c;*

For each tbuow$_e$ in the to-be GUoWs do the following

Encapsulate the associated to-be Q_UoWs of tbuow$_e$ into one to-be GQ_UoW tbgquow$_b$;

End for GUoWs

//Drawing

Re-Draw the GQ UoW diagram using the GQ_UoWs set;

Re-Translate the GQ-UoW diagram into the to-be 1st cut architecture;

Re-Generate the to-be 2nd cut architecture from the 1st cut architecture;

Figure 5.15: Algorithm the Alignment of the as-is GQ Riva BPA to the BSV (Part 3 of 4).

//Re-Identifying the goals and the quality for each process in the generated 2nd cut BPA;

Identify the set of the to-be goal-based and quality-linked processes in the generated 2nd cut architecture from the alignment, GQ_Process=*{*gqp_0, gqp_1, ...,gqp_x, ..., gqp_o*}*, $0 \leq x \leq o$;

Recall the goal-network in the BSV input;

For each identified process in generated 2nd cut architecture, gqp_x*, do the following*

Consider the name of the gqp_x *as the goal of the process;*

Use the goal-network in the BSV in determining the contribution of the gqp_x;

Recall the original GQ_UoW, $tbgquow_b$, *that generated the* gqp_x;

De-capsulate the original $tbgquow_b$ and allocate the desired quality requirements that constrain the gqp_x, along with their quality models if required;

End for each gqp

//Detecting processes and identifying new processes along with the alignment-level

Let Alignment_Level be the variable for each GQ_Process instance and has one out of three values: novel process, as-is redesigned process or as-is with no required redesign.

For each identified process in generated 2nd cut architecture, gqp_x*, do the following*

If the gqp_x *exists in the as-is processes list*

If the gqp_x *requires redesign*

Flag the Alignment_Level of gqp_x *as" as-is redesigned process";*

Else if the gqp_x *does not require redesign*

Flag the Alignment_Level of gqp_x *as" as-is process with no redesign";*

End if else

Insert gqp_x *in the Detected_Prcesses list;*

Else if the gqp_x *does not exist in the as-is processes list*

Flag the Alignment_Level of gqp_x *as" novel process";*

Insert gqp_x *in the New_Prcesses list*

End if else

End for each gqp

Turn the GQ_Process=*{*gqp_0, gqp_1, ...,gqp_x, ..., gqp_o*}* $0 \leq x \leq o$*, into to-be GQ_Processes that represent the union of the detected processes and the new processes.*

Figure 5.16: Algorithm the Alignment of the as-is GQ Riva BPA to the BSV (Part 4 of 4).

5.9 Ontologising the Proposed Alignment Approaches

This section presents ontologising the alignment approaches proposed in Section 5.8. In short words, ontologising is carried out using the notion of the ontology imports and the creation of the alignment classes and relationships using OWL-DL that are both adjusted to a particular alignment approach shown in Figure 5.9. It in necessary to note a limitation in the ontologised alignment approaches, that is the extraction of as-is EBEs and to-be ones from a GQOnt ontology is carried out manually. This is because it is not easy to automate an activity that requires intelligent human-based decisions considering that EBEs vary from one person to another.

With regard to the first alignment approach (shown in Figures 5.10- 5.12) that requires the BSV and the as-is Riva BPA, they are both semantically represented using the GQOnt ontology and the original srBPA ontology, respectively in order to produce the GQ-srBPA ontology that semantically represents the GQ-Riva BPA resulted from the alignment. In this case, the user imports three different ontologies and carries out the alignment using the OWL-DL-based alignment classes and relationships. This case is depicted in Figure 5.17, where the corresponding implementation of this alignment is attached in Appendix H.

Similarly, the second alignment approach (shown in Figures 5.13- 5.16), that requires the BSV and the as-is GQ Riva BPA as pre-alignment resources in order to generate the to-be GQ Riva BPA, is ontologised using two different ontologies that are the GQOnt and the GQ-srBPA ontologies. This semantic alignment is shown in Figure 5.18, where its implementation is attached in Appendix H

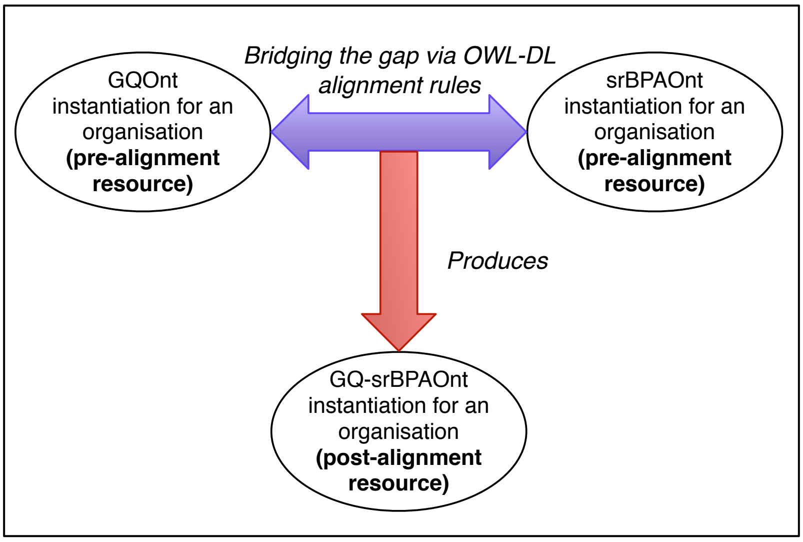


Figure 5.17: Aligning Three Ontologies For the First Alignment Approach

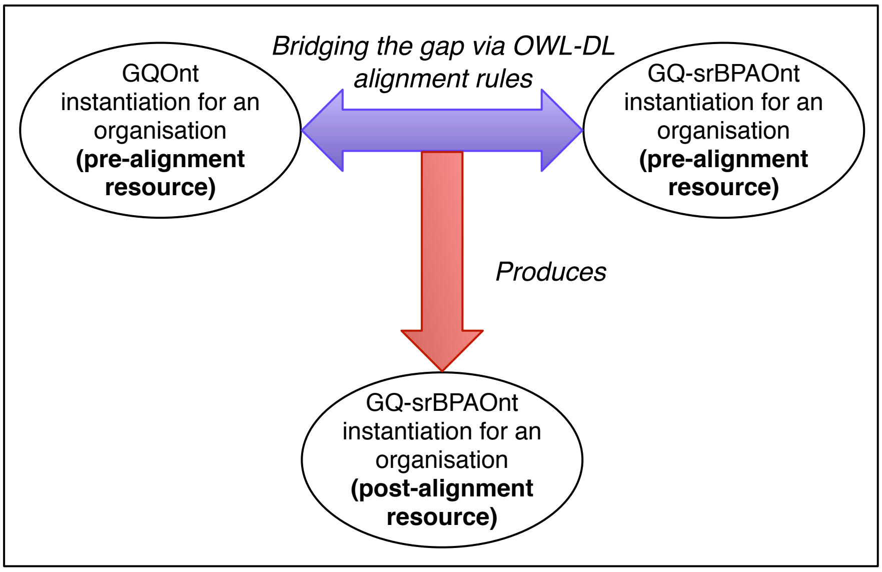


Figure 5.18: Aligning Two Ontologies For the Second Alignment Approach

5.10 Evaluating the Semantic Alignment of a Pre-existing Riva BPA to the BSV of an Organisation Using the CEMS Faculty of Administration Pilot Study

This section presents an initial evaluation of the semantic alignment proposed in the earlier sections within this chapter using the CEMS Faculty of Administration pilot study. The BSV and BPMs for this study are absent. Therefore, few interviews have been conducted in order to elicit the required missing BSV as part of the carried out investigation. The interview questions are attached in Appendix K, where their answers are shown as resulting BSV models of the CEMS Faculty of Administration in Appendix L. Due to the limited interview sessions conducted for the investigation, partial BSV of this pilot study is obtained (i.e., partial goal network and few SD models), as attached in Appendix L.

5.11 Discussion and Conclusion

This section aims at presenting a discussion of the entire work presented in this chapter, which has semantically aligned pre-existing Riva BPAs with the BSV of a particular business organisation. The two proposed alignment approaches were conducted with a special attention towards the goals and the quality requirements of the ultimate processes that set up the 2nd cut architecture. In addition, the two alignments cover the potential of having the pre-existing Riva BPA, which requires an alignment with the BSV, designed using either the original method (Ould, 2006) or the refined method that has integrated the notion of goals and quality requirements as was presented in Section 5.8. It is necessary to recall that the integration of goals must be firstly carried out, as they shape the identification of the BPs. Whenever the

BPs are well-identified, they can be characterised with associated consistent desired quality requirements.

With regard to the difference between the original Riva method for the BPA design (Ould, 2006) and the refined one, the former is EBE-driven and purely functional or behavioural through concentrating on the BPs' derivation rather than enriching them. The latter is based on the claim that the BPs must stem or be driven from the organisation's goals. Hence, using the notion of goals has enriched the current heuristics that identify the EBEs. In addition, a BP must be characterised with the desired constrains and business quality requirements. In this chapter, an attention has been paid for the importance of a BSV role that does not only afford the representation of the business strategies and tactics, but its capability in influencing the design of relevant models with an anticipated minimal architecting effort.

With regard to the BPMNs, they have been enriched using the notion of the appendix that is essentially designed in order to accommodate the required knowledge resulted from the alignment with traceability support to goals and soft goals. The notion of the appendix within a BP is anticipated to assist the designers in the future for further enriching through attaching the BP with missing, yet required information that cannot be engaged in the detailed representation of the BP workflow. Accordingly, in this chapter, an effort has been paid in addressing the consistency between the BPMNs and their relevant GQ Riva-based architecture.

Whenever an organisation tends to amend its goals or soft goals, it essentially should conduct the desired amendments through reusing the BSV in order to generate the up-to-date one. Accordingly, any simple or complicated amendments within the BSV are anticipated alerting the BP architect for the required alignment of the BPA and associated processes with the recently generated BSV. This is because the BPs in the BPA should be designed with the reference to the goals of the organisation along with their constraints and desired quality attributes that merit the BP among similar ones in the domain, and hence this would increase the competition in the same business market.

The semantic alignment aided with the ontologies assists the stakeholders in not only managing and reserving the high volume of information, but also proposing an elegant synthesis and linking of semantics between the two rich areas or views (i.e., the BSV and the BPA along with the associated processes).

The conducted pilot investigation and evaluation has apparently demonstrated minimal architecting effort for generating the to-be Riva BPA benefiting from the notion of detecting as-is Riva BPA elements (e.g., as-is EBEs, as-is G_EBES, etc.). In addition, it shows new knowledge coming from bridging the gap between the BSV and the GQ/Riva-based BPAs. In particular, the as-is Riva BPA designed using the Ould's method appears as a sub part of the partial generated to-be GQ Riva BPA with reference to the pilot study example. However, it cannot be guaranteed that this is the case for each conducted alignment. In the pilot study, it was apparent that the architect did not only benefit from the pre-existing Riva BPA models but from their ontological reused representation as well. Accordingly, the reusing benefit is anticipated to be remarkable in the presence of the BPs that are absent in the pilot study.

With regard to the two proposed alignment approaches, deriving a new knowledge is claimed. In the first alignment approach, a pure new knowledge is created in the output such as the GoP and the QoP that both did not exist before the alignment. In the second alignment approach, the current pre-alignment resources are refined in order to adjust to the recent BSV. Hence, the refinements represent new knowledge for the resulted BPA and associated BPMs. In addition, the notion of the early integration of GoP and the QoP is anticipated to assist in predicting an early cost execution calculation. However, this requires further research work and evaluation.

The alignment approaches act as gaps identifier or detector with the pre-existing Riva BPA where the gaps are filled after the alignment. This is particularly apparent with the first alignment approach where the goals and quality requirements are originally absent. By and large, it is necessary to have an aligned BPA with goals because the BPA itself is an important source to quickly know the key BPs in an organisation and how they are systematically organised and connected along with any of their identified gaps. For example, an organisation may run 200 BPs that are highly presented in the BPA. For some reason few of them updated their goals and this recalls the as-is BPA to investigate the change. Therefore, the analysts prefer to utilise the as-is BPA in order to produce the to-be BPA model. And if the BPA is not up-to-date (i.e., aligned) designed, then the BPA is highly anticipated to act as a useless resource that should represent an understanding of the organisation's main BPs.

Few limitations are highlighted at this stage of the work. First, the alignment cost determination is little as it lacks an advanced or rich representation of the BP costs after the conducted alignment. Second, the chapter proposed two alignment cases however, one of them has been only initially evaluated (i.e., the first alignment case) where the second is still theoretical and needs further evaluation. Third, since the BSV lacks addressing business risks and their mitigation mechanisms, then this limitation is propagated into the alignment approaches. Fourth, the time element is absent or not considered neither in the two proposed alignment approaches nor the two Riva methods (i.e., original and the refined). However, it is related and embodied in a way or another in the design cost perspective of BPMs based on the implication of an alignment. However, the cost and the time perspectives are still not well addressed. Fifth, the case strategy process (i.e., one of the process categories in the original Riva method as presented in Chapter 2) is still absent, as the relevant original information and heuristics regarding this kind of processes are not highlighted or very few in the best case (Ould, 2006).

Finally, the work of this chapter contributed to answering the research questions flagged in Figure 5.19 and participated within the activated research process phases as shown in Figure 5.20.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1					✓				⚑
RQ2					✓				⚑
RQ3					✓				⚑
RQ4									⌛

Figure 5.19: Answering Research Questions With the Reference to the Work of Chapter 5.

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									**Activated**
RP2					✓				**Active**
RP3					✓				**Active**
RP4(ai)					✓				**Active**
RP4(aii)									⌛
RP4(bi)									**Activated**
RP4(bii)					✓				**Active**
RP4(biii)									⌛
RP5									⌛
RP6									⌛

Figure 5.20: Related Research Process Phases With Respect to the Work of Chapter 5.

Chapter Six: The GQ-driven Software Service Identification Approach

6.1 Introduction

The purpose of this chapter is to present work required for the third layer in the proposed framework namely, the GQ- software service identification layer. Recalling that this layer is borrowed from the second layer of the original BPAOntoSOA framework (Yousef, 2010) and extended in order to integrate goals and quality requirements. The work of this chapter takes place in two phases within the research process presented in Chapter 3 Section 3.4 that are the second phase (i.e., the early theoretical phase) and the fourth phase (i.e., the original BPAOntoSOA framework enhancement phase). In addition, the work of this chapter is employed in order to accomplish the first essential requirement for the GQ-BPAOntoSOA framework development.

The aim here is to apply the SI method employed in the BPAOntoSOA framework (Yousef, 2010) but using the new GQ- Riva BPA method presented in Chapter 5. This chapter is structured as follows: Section 6.2 recalls the software service identification layer of the BPAOntoSOA framework and presents the amendments proposed in this layer integrated into the SI layer of the GQ-BPAOntoSOA framework. This revisit entails refining the SI approach used in the BPAOntoSOA framework and demonstrating the refined approach. Section 6.3 presents the refinements carried out within the original SI approach in order to generate goal-based and quality-linked RPA clusters. Also, this section presents an initial demonstration of the refined SI approach using partial GQ-Riva BPA model of UWE's CEMS Faculty of Administration. In Section 6.4, the goal-based and quality-integrated service identifier component is presented. In this section, the service identifier component algorithm presented in (Yousef, 2010) is recalled in order to reuse it in identifying services with respect to the GQ-BPAOnt ontology rather than the BPAOnt ontology. Section 6.5 presents the integration of QoS requirements into the identified services using the GQ-BPAOnt ontology. Similarly, the service capability identifier of the BPAOntoSOA framework is revisited to seek for the required amendments in Section 6.6. Finally, a discussion and conclusion of the work close the chapter in Section 6.7.

6.2 Comparing the Original Software Service Identification Layer and its Proposed Extension

This section briefly presents the structure of the original software service identification layer and its components. The two adaptations are shown in the Figures 6.1 and 6.2, respectively.

With regard to the original layer, the input was the BPAOnt instantiation for an organisation (Yousef, 2010). The input was processed in the layer within two components that are the service identifier component and the service capability identifier in order to produce the output of the candidate software services and their associated capabilities. In the original layer, an RPA-based SI approach was proposed in order to identify the candidate software services from the given Riva BPA taking into account meeting the SOA principles (Yousef, 2010), as was previously discussed in Chapter 2.

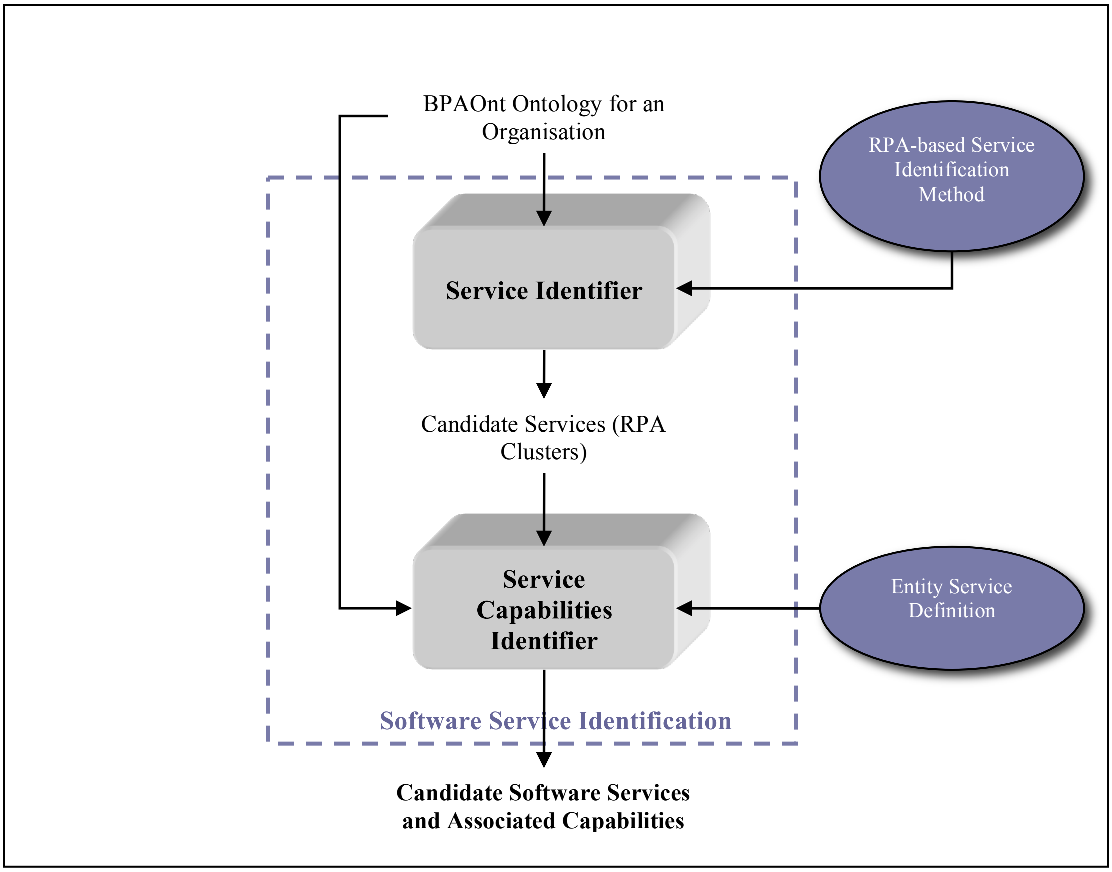


Figure 6.1: The Software Service Identification Layer in the Original BPAOntoSOA Framework [Source: (Yousef, 2010), Used with the author's permission].

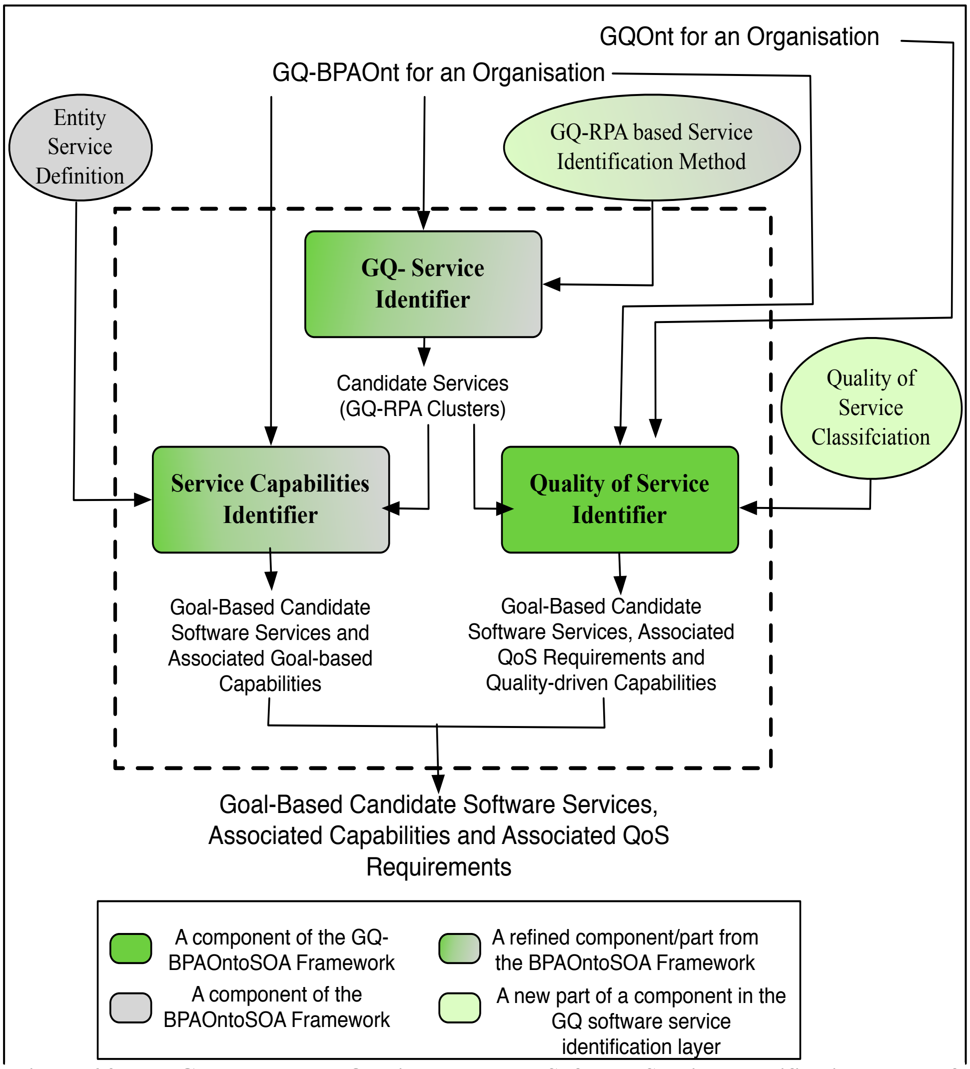


Figure 6.2: The Goal-Based and Quality-Integrated Software Service Identification Layer of the GQ-BPAOntoSOA Framework

6.3 The Goal-based and Quality-integrated Service Identification Approach: A Refined RPA-based Service Identification

The Goal-based and Quality-linked Service Identification (GQ-SI) approach is employed in the GQ service identifier component as shown in Figure 6.2. The approach follows the footsteps in the BPAOntoSOA framework with regard to deriving services as clusters. However, the clusters are GQ-RPA clusters instead RPA ones. It is necessary to recall that the ultimate identification of clusters is obtained after deleting the conditional request and deliver relationships using associated BPMs in a Riva BPA. The GQ-RPA clusters are explained next.

6.3.1 GQ-RPA Clusters

The concept of the RPA cluster was firstly introduced and developed by the work of the BPAOntoSOA framework. Recalling that the RPA cluster consisted of CPs and/or CMPs (Yousef, 2010). The RPA cluster either was a standalone CP or interrelated CPs and/or CMPs via the relations of the Riva-based 2nd cut architecture (Ould, 2006) (Yousef, 2010). In this section, RPA clusters are informed by the associated goals and quality requirements in order to form or set up the introduced GQ-RPA clusters.

The GQ 2nd cut architecture generated from the GQ Riva BPA method consists of the GQ CPs and/or GQ CMPs that are either standalone or connected with each other via the 2nd cut process architecture relations (e.g., request, start and deliver). It is apparent that the GQ-RPA clusters and the original RPA clusters have in common the connected CPs and/or CMPs. However, the GQ-RPA clusters are related to goals and quality requirements. The goal of a GQ-RPA cluster is what differentiates a cluster from another. In other words, there must not be clusters with the same goal. In addition, each goal-based cluster is constrained by the desired quality requirements such as confidentiality. Accordingly, if the GQ-RPA cluster consists of a stand alone GQ-CP or GQ-CMP, then the goal of the process is the goal of the cluster. If the GQ-RPA cluster consists of interrelated GQ CPs and/or GQ-CMPs, where each of them has associated goals, then the Common Goal (CG) that is collective goals (i.e., union of goals) among the GQ-CPs and/or GQ-CMPs in the entire cluster. Hence, now the cluster is a G-RPA cluster.

With regard to the quality requirements in the G-RPA cluster, standalone G-CP or G-CMP has associated quality requirements along with the quality reference(s) to the relevant quality model representation (e.g., NFR frameworks) that are already identified within the GQ-Riva

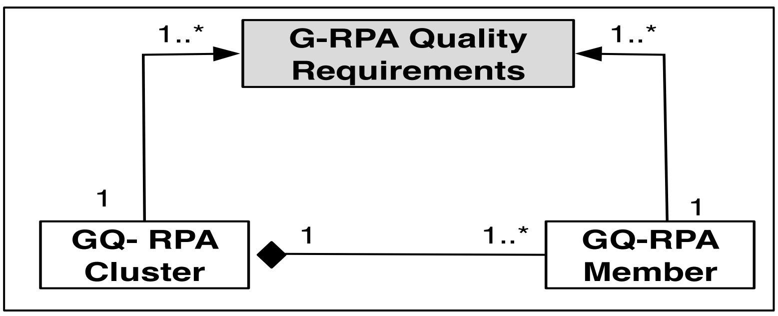


Figure 6.3: The Relation Between the GQ-RPA Cluster, GQ-RPA Member and the G-RPA Quality Requirements Using the UML Notations.

2nd cut architecture. In the G-RPA cluster, the quality requirements are classified into two categories: the member level category and the cluster level category. The first category is concerned with the quality requirements for a single process whether it is a standalone process or connected to another. The second category is concerned with the quality requirements that constrain the overall cluster it self, which may consists of a standalone process or of a set of connected processes. If the cluster is composed of a standalone G-CP or G-CMP, then the quality requirements of the cluster are the ones identified for the standalone process. However, if the cluster is composed of connected G-CPs and/or G-CMPs, then the quality requirements of the entire cluster are the union of the quality requirements identified for each member. The resulting quality requirements from the union describe the desired attributes that constrain the entire cluster. Therefore, quality requirements of the member levels must be already identified in order to determine the quality requirements of their cluster. Hence, quality requirements of the first category are part/subset of the second category. The quality requirements relation between the member level and cluster level is illustrated in Figure 6.3.

The partial GQ Riva 2nd cut architecture of CEMS Faculty of Administration is shown in Figure 6.4. The bounded circles in Figure 6.4 represent GQ-RPA clusters. Table 6.1 presents the detailed description of each GQ-RPA cluster in relation to their goals, quality requirements and GQ-RPA member(s). The coloured entries in Table 6.1 are with respect to the ones in Figure 6.4. In this pilot study, the NFR framework models are absent and thereby no quality references are utilised to identify quality requirements. The BPMs are absent too. Therefore, it is not possible to identify the conditional and normal relationships in order to make the GQ-RPA clusters. However, Figure 6.4 shows the two kinds of the GQ-RPA clusters, and they are assumed that they are generated with respect to the remained normal deliver and request relationships.

The goal-based and quality-linked clusters (GQ-C) are GQC1, GQC2 (i.e., new standalone cluster), GQC3, where each consists of an individual GQ-CP represented as “handle a student”, “handle a registration”, “handle an award definition”, and “handle an external, respectively. The GQC4 represents interrelated GQ-CPs and GQ-CMPs where their common goal is the union of eight members’ goals. The goal of a cluster is the goal of the member and not their contribution to other goals.

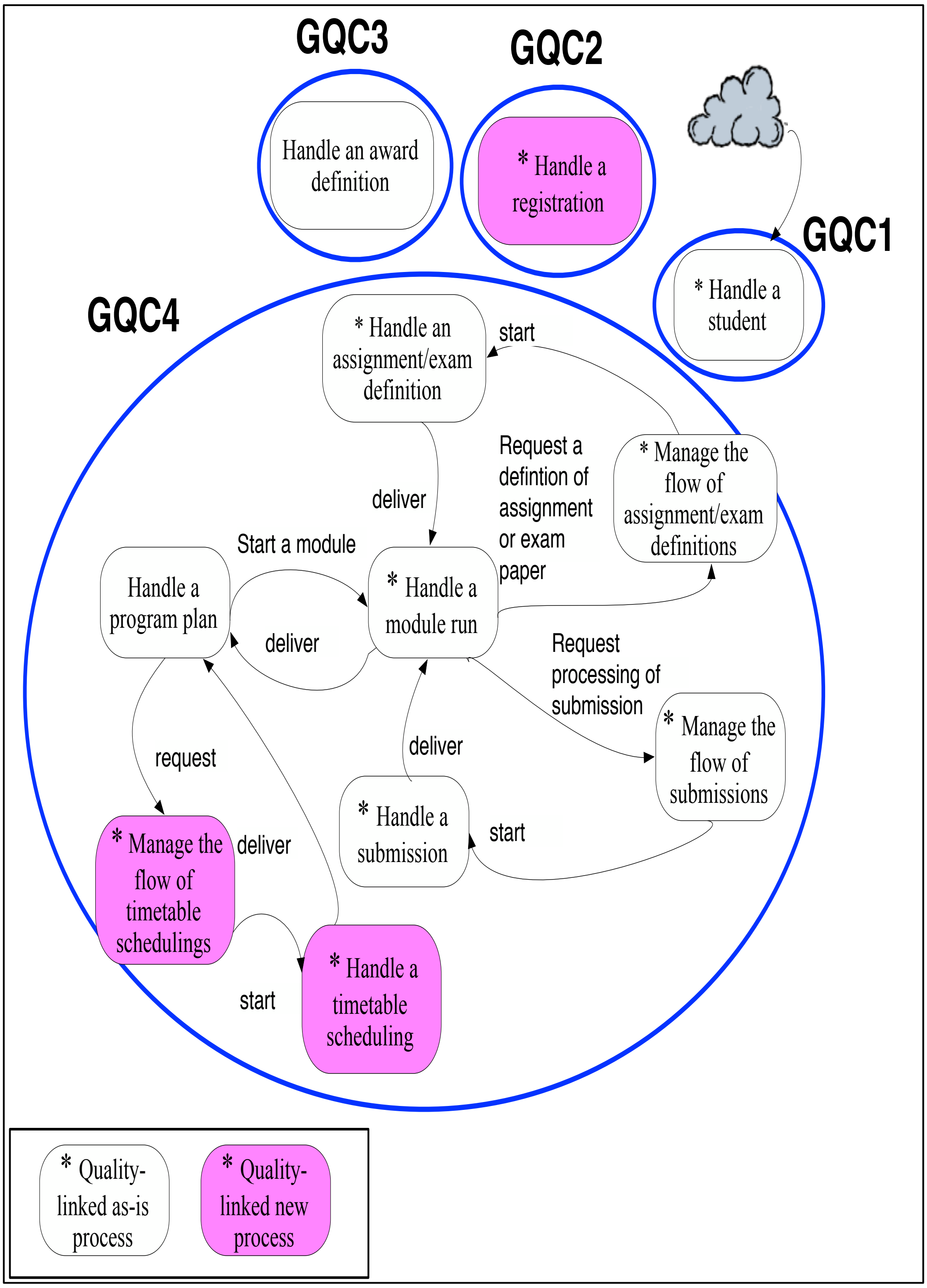


Figure 6.4: The partial GQ Riva 2nd cut architecture diagram for the CEMS Faculty Administration with GQ-RPA clusters represented in blue bounded circles [Source: (Green and Ould, 2004), Adapted with the author's permission].

Table 6.1: The partial GQ-RPA Clusters derived from the GQ Riva 2nd cut architecture for the CEMS Faculty Administration.

GQ Cluster number	Goal Of Cluster	Cluster Member(s)	Member Goals	G-RPA Cluster Quality Requirements
1	Student is handled	GQ-CP: Handle a student	**GCP**: Student is handled.	1- Quick [student enrolment].
2	Registration is handled	GQ-CP: Handle a student registration.	**GCP**: Student registration is handled.	1- Timely [registration].
3	Award definition is handled	GQ-CP: Handle an award definition.	**GCP**: Award definition is handled or received.	Not determined
4	Assignment/exam definition is handled and managed, module run is handled, submission is handled and managed, timetable scheduling is handled and managed and program plan is handled.	GQ-CP: Handle an assignment/ exam definition	**GCP**: An assignment/exam definition is handled.	1- Accuracy.
		GQ-CMP: Manage the flow of assignment/ exam definitions	**GCP**: The flow of the assignment/exam definitions is managed.	1- Accuracy.
		GQ-CP: Handle a module run	**GCP**: Module run is handled.	1- Timely [module run].
		GQ-CP: Handle a submission	**GCP**: A submission is handled.	1-Secuity [submission]. 2- Privacy [submission information].
		GQ-CMP: Manage the flow of submissions	**GCP**: The flow of submissions is managed.	1-Secuity [submission]. 2- Privacy [submission information].
		GQ-CP: Handle a timetable scheduling	**GCP**: A timetable scheduling is handled.	1-Accuray.
		GQ-CMP: Manage the flow of the timetables' scheduling	**GCP**: The workflow of timetable scheduling is managed.	1-Accuray.
		GQ-CP: Handle a program plan	**GCP**: The program plan is handled.	Not determined

6.3.2 GQ RPA Clusters as Goal-based and Quality-integrated Candidate Services

In this section, Yousef's hypothesis in relation to RPA clusters as candidate services' is revisited to address the aim of this research. The reconsidered hypothesis is:

"GQ-RPA clusters that can be identified from the 2nd cut architecture business process architecture diagram using the GQ Riva method can be considered as candidate services suitable for building goal-based and quality-supported SOA-based system."

Although goals and the quality requirements has been integrated into the Riva-based 2nd cut architecture, it is still required to maintain the desired simplicity of SI method proposed in the original BPAOntoSOA framework in (Yousef, 2010). The refinements which utilise goals and quality requirements in the RPA cluster in order to generate a GQ-RPA cluster will definitely involve refinements into the SI method as discussed below. The current SI method (Yousef, 2010) is revisited with respect to the integration of the goals and quality requirements. Moreover, the proposed mapping of the characteristics of GQ-RPA to services' definition and principles are revisited.

6.3.2.1 GQ-RPA Clusters to Satisfy Simplicity

Refinements to the Riva method remained systematic and simple to comprehend. These refinements are inspired from the original Riva method (Ould, 2005). For example, the EBQs were inspired to constrain EBEs, the QUoW concept to constrain UoW using the *constrain* dynamic relation in the GQ-UoW diagram. Also, the GQ Riva method has mitigated the difficulty regarding the identification of EBEs through systematically eliciting them from the BSV instead of brainstorming them. The GQ Riva method is concerned with modelling the behavioural and non-behavioural business entities instead of being solely for behavioural ones for the organisation that is under the process of designing its BPA and associated BPMs. Hence, the overall refinements within the Ould's Riva method have enriched the Riva method and thus in driving the simplicity of the GQ-RPA clustering approach. Accordingly, GQ-RPA clusters continue to enrich the original BPAOntoSOA framework SI approach using goals and the quality requirements.

The footsteps of the BPAOntoSOA framework are followed in relation to using the BPA instead of the BPMs in identifying the services for the same determined reason in (Yousef, 2010). This is because BPA has the appropriate level of abstraction that fits with SOA, where both do not require detailed and complex information (Yousef, 2010), yet their information is rich and still abstract after engaging the goals and the quality requirements.

6.3.2.2 Mapping the GQ-RPA Clusters to the Service Definitions and Principles

Although the desired simplicity of the GQ-RPA clusters is remained, it is still required to revisit the service definitions and principles with respect to the GQ-RPA clusters. In the BPAOntoSOA framework, the use of service definitions and principles were satisfied and mapped with respect to the notion of RPA clusters in (Yousef, 2010). In this section, the aim is to determine the extent to which the current service definition and principles satisfy the GQ-RPA clusters in relation to their association with candidate software services.

In the original BPAOntoSOA framework, it was considered that RPA clusters satisfy service definitions and principles (Yousef, 2010). The current service definition used in (Yousef, 2010) states the entity service as a configuration of a business service, where its functional boundary and context are derived from at least one business entity such as student, module, programme and administrator. The notion of the RPA within the GQ-RPA cluster does already satisfy the current definition of the entity service based on (Yousef, 2010). However, the engagement of goals and quality requirements in the RPA clusters highlighted the missing information, yet required in the current definition of the entity service. Hence, this has led to refine the current meaning and understanding of the entity service as follows:

"*An entity service is the configuration of a business-centric service that bases its functional boundary on one or more of the related goal-based business entities and bases its non-functional boundary on the entire quality requirements, which are guided and indicated by using the service-oriented NFRs classification, of the goal-based business entities. The context of the entity service is shaped using its functional and non-functional boundaries that both stem from the business strategic view of the organisation*".

The above refined definition describes the entity service as being comprehensive due to its comprising of multiple, relevant and required perspectives in the entity service (e.g., functional boundary, non-functional boundary, goal-based entity and the quality requirement). This definition highlights the important information that form the entity service. One part of this information is the need for the identification of the goal-based business entities in order to possess a well-defined functional boundary of the entity service based on goals yielding a firm and a precise determination of the purpose for the desired reusability. This makes entity services better in addressing the high reusability from the goal point of view, namely Goal of Service (GoS). The second part of information is the need of quality requirements of the goal-based business entities in order to identify the desired quality attributes and/or characteristics addressed for goal-based business service yielding a better selection of the entity amongst goal-based entity services that share a particular common purpose. That is, the quality requirements are what distinguish a goal-based entity from another in case they possess a common goal or purpose, and thus a better determination for the reusability based on the Quality of Service (QoS). The third part of information is set up based on the harmony between the functional and the non-functional boundaries that represent the first part and second part of required information respectively in the entity service. Overall, the entity service is now more cohesive due to the aforementioned information.

The GQ-RPA clusters comply with the new definition of the entity service. In particular, the GQ-RPA clusters consist of GQ-CPs and GQ-CMPs as members that are goal-based and quality integrated processes in the GQ Riva-based 2nd cut architecture. In the functional boundary, the GCP and GCMP have a corresponding GUoW that has generated them where the GCP handles an instance of the GUoW and the GCMP manages the flow of the instances of the GUoW. This GUoW is originally a GEBE that holds an interesting lifetime. Hence, the GQ-RPA cluster bases its functional boundary on goal-based entities, namely GEBEs. Similarly with the non-functional boundary, the QCP and QCMP are identified from each corresponding QUoW in the individual UoW diagram. Each QUoW is originally a QEBE that constrains GEBE(s). Hence, the GQRPA cluster bases its non-functional boundary on the entire identified quality requirements of a goal-based business entity, namely EBQs.

Loose coupling

The standalone GQ-CP or GQ-CMP in the GQ-RPA cluster does satisfy the loosely coupling principle. This is because it is not related to another cluster through any of the 2nd cut architecture relations. The GQRPA cluster that consists of related GQ-CPs and/or GQ-CMPs through the 2nd cut architecture relations (e.g., start, deliver, request), yet not related to another GQ-RPA cluster, satisfies this principle. This conformation the loose coupling SOA principle is similar to the one in (Yousef, 2010), as there is no change within the notion of the relations in the GQ-RPA clusters. Hence, the GQ-RPA clusters are loosely coupled in the GQ Riva 2nd cut architecture diagram.

For example, in the partial GQ BPA of the CEMS faculty administration in Figure 6.4, the standalone clusters GQC1, GQC2, and GQC3 are loosely coupled. The members of the GQC4 cluster are depending on each others to satisfy the goal of the cluster but they do not depend on another cluster. Accordingly, the GQC4 is loosely coupled too. As a conclusion, the BPA in Figure 6.4 consists of four GQ-RPA clusters that are loosely coupled.

Abstracting underlying logic

The RPA cluster acted as a bloack box because it considered its abstract processes as it underlying logic. Also, the GQ-RPA cluster acts as a black box but it considers the goal of the cluster its underlying logic. The goal embodies the abstracted processes. In addition, the goal of the cluster does not put into consideration any of the relations between the processes due that they may not refer to the abstract underlying logic anymore. Hence, this increases the sustainability of the cluster with its firmed goal(s) and decreases the associated information about the GQ-RPA cluster by not presenting the list of related processes and their relationships. This is because the goal of the GQ-RPA cluster embodies them yielding to a stateless cluster.

With regard to the two kinds of clusters in the GQ-RPA clusters, the goal of standalone cluster is the abstract underlying logic. In the GQ-RPA cluster that consists of related GQ-CPs and/or GQ-CMPs, the CG represents the underlying logic of the entire cluster.

Reusability

The service entity definition is refined, where refinements are inspired from the GQ-RPA cluster, by involving the GoS and the QoS that both collaborate in order to generate a well identified entity service rather than its functions. A well-identified entity service is assumed to provide a clear purpose of the service and thereby its purpose of reusability. Hence, this is anticipated to encourage higher reusability than the original RPA clusters.

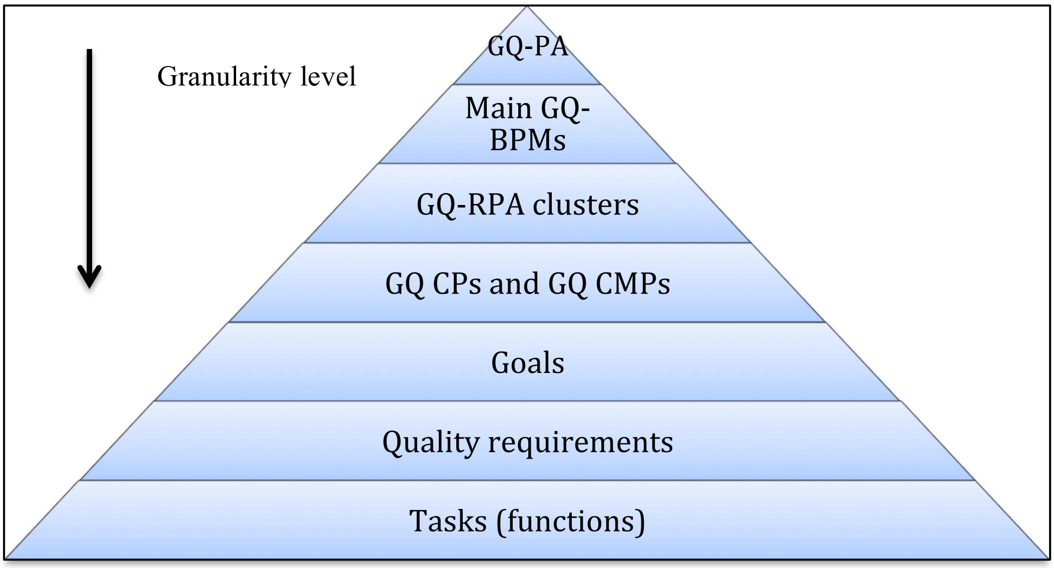


Figure 6.5: The Granularity level hierarchy of the GQ-PA and its contents that assist in identifying the candidate services.

Composability, stateless and discoverability

The granularity level of the GQ-RPA clusters remains similar to the granularity level of the RPA in the BPAOntoSOA framework (Yousef, 2010) that is in the middle between the GQ PA, which is too coarse-grained and the quality requirements, and tasks (functions) that are too fine-grained as shown in Figure 6.5. Moreover, GQ-RPA clusters conform to the stateless SOA principle, as the GQ-RPA cluster hides the information with respect to its functions and the interactions between its processes. Although the GQ-RPA clusters manage more information than the original RPA cluster, still it satisfies the stateless principle by hiding this load of information. The GQ-RPA cluster manifests its goals and quality requirements (without their relations) where both are abstract. For the discoverability of a service, GQ-RPA clusters do early assist in determining the design decisions of a service by allowing the goal of a service (GoS) along with its quality requirements (QoS) to be the ID of the service. Hence, this minimises the potential of creating redundant software services.

The rest of the SOA principles are concerned with the design of the service and by this it is agreed with the BPAOntoSOA framework in conforming to most of the SOA principles (Yousef, 2010).

6.4 The GQ Service Identifier Component

The proposed GQ-RPA cluster is the cornerstone of the GQ service identifier component and uses the resulted GQ-BPAOnt, the output from the second layer in the GQ-BPAOntoSOA framework, in order to semantically perform the service identification approach. It is apparent that this component borrows the function of the original service identifier component in the BPAOntoSOA framework in basing the work of GQ-RPA cluster, yet it involves few required refinements in order to meet the aim of this research. In particular, the identification of a cluster requires identifying not only the members, yet it also involves the identification of its goals and quality requirements. Recalling that the identification of goals for a standalone cluster is the goal of the member, where in the connected members the goal is the CG between the members. In addition, the identification of the main quality requirements for a standalone cluster is the quality requirements of the member where they are assembled from the members in connected cluster in order to arrive at a comprehensive desired characterisation. Therefore, a revisit to the original SI approach or algorithm, that hypothesised the RPA clusters as candidate services in (Yousef, 2010), used in the BPAOntoSOA framework is required in order to engage the proposed refinements.

6.4.1 Semantic Identification of GQ-RPA Clusters

The sematic identification of the GQ-RPA clusters is driven from the GQ-BPAOnt ontology. Accordingly, revisiting GQ-BPAOnt classes and properties steers the way for the semantic identification of the GQ-RPA clusters. Recalling that the identity of the cluster is denoted by its goal and main quality requirements, and thus this requires an extra effort while revisiting the GQ-BPAOnt ontology in semantically retrieving the goal of the process(es). The original class RPA_Cluster is renamed into **GQ_RPA_Cluster** in order to adjust with need of identifying goals and quality requirements regardless of the kind of the cluster. Two new ontology relationships are created in order to determine the cluster's goal and quality requirements. The first ontology property is **hasGoalOfCluster** that is with regard to identifying the goals, where the domain is GQ_RPA_Cluster and the range is **GoP** (i.e., already implemented in the GQ-BPAOnt ontology). The second relationship is **hasQualityOfCluster** that is for identifying the quality requirements for the cluster, where the domain is GQ_RPA_Cluster and the range is **QoP** (i.e., already implemented in the GQ-BPAOnt ontology). It is necessary to recall the original **PA2Elements** class designed in order to encompass all CPs and CMPs that belong to the 2nd cut architecture in GQ-BPAOnt

(Yousef, 2010). The PA2Elements members were instantiated using the original SWRL rules (Yousef, 2010):

CP(?cp) ^ belongsop2ndCutArchitecture(?cp, PA_2nd_cut_Diagram) → PA2Elements(?cp)

CMP(?cmp) ^ belongsop2ndCutArchitecture(?cmp, PA_2nd_cut_Diagram) → PA2Elements(?cmp)

Recalling that the **PA2Elements** class contains members that are all relationships in the 2nd cut architecture (Yousef, 2010).

Similar to the RPA cluster, the GQ-RPA cluster is either a standalone GQ-CP or related GQ-CP and/or GQ-CMP, as shown in Figure 6.6. The standalone GQ-RPA cluster consists of a standalone GQ-CP that belongs to the 2nd cut process architecture taking into account that it does not participate in any of request, deliver and start relations neither as a source nor as a destination. This kind of clusters is semantically identified in the same manner used in the original component (Yousef, 2010) through looking for the GQ-CPs that are with null value of the original properties: hasStartRelation, hasRequestRelation and hasDeliverRelation taking into account that the GQ-CP must not be a destination for any of those relation. The standalone cluster must have a goal that is retrieved through the property **hasGoP** (i.e., in the GQ_BPAOnt ontology). As for the main quality requirements of this cluster, they are retrieved using the property **hasQoP** in the GQ-BPAOnt ontology. The SWRL rules that describe this mapping are for the goals and quality requirements in a standalone cluster:

1- GQ_RPA_Cluster(?c) ^ isStandalone(?c, True) ^ PA2Element(?e) ^ CP(?e) ^ hasMembers(?c, ?e) ^ hasGoP(?e,?gop) → hasGoalOfCluster(?c, ?gop).
2- GQ_RPA_Cluster(?c) ^ isStandalone(?c, True) ^ PA2Element(?e) ^ CP(?e) ^ hasMembers(?c, ?e) ^ hasQoP(?e,?qop) → hasQualityOfCluster(?c, ?qop).

As for the second kind of the GQ-RPA cluster (i.e., connected GQ CP and/or GQ CMPs), members are connected through a start, deliver, or request relationships in the GQ 2nd cut architecture. However, the related members (i.e., processes) are not connected to another cluster. This arrangement is borrowed from the original service identifier component in (Yousef, 2010). The goal of the cluster is the goals of the members using the property **hasGoP**. The cluster's quality requirements are exploited from each member using the property **hasQoP** in order to identify its entire quality requirements. The two new properties, that are **hasGoalOfCluster** and **hasQualityOfCluster,** are used in this kind of cluster and illustrated within two algorithms as will be shown in the next section.

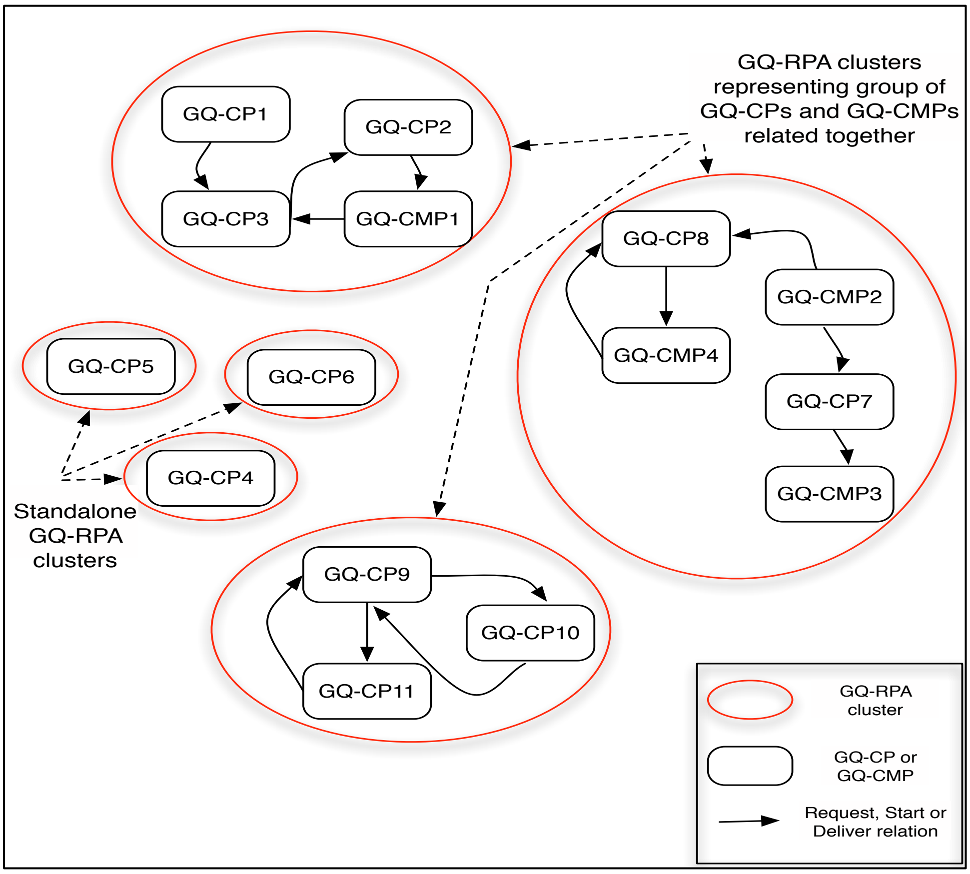


Figure 6.6: The GQ-RPA Cluster Examples

Finally, it is important to indicate that the integration of goals and quality requirements draws the behavioural and the non- behavioural characteristics of a cluster, whether it is of the first kind or of the second kind.

6.4.2 Refining the Algorithm for Identifying Software Services Using the GQ-BPAOnt Ontology

Recalling the original aim of the service identification, it was required to reuse the BPAOnt ontology in order "to identify the clusters" that are represented as "both isolated CPs, and the group of elements connected to each other, but not connected to other groups or standalone CPs" (Yousef, 2010). This section presents refinements of the original algorithms in (Yousef, 2010) that were used to identify members of the connected clusters. It is necessary to remind that the algorithms are carried out after ensuring setting the **isConditional** property (i.e., domain is Request and Deliver Classes where the range is Boolean), as recommended in (Yousef, 2010). Figure 6.7 shows an abstract illustration of algorithms used to derive GQ-RPA clusters and their associated GQ members. In these algorithms, the extension is depicted in red text, where the original body is reused (Yousef, 2010) and represented in black as shown in Figures 6.8 -6.10.

In Figure 6.8, the algorithm remains with its original aim in deriving clusters and their members; however, the clusters are GQ-RPA clusters instead of RPA only. In short words, the original algorithm started clusters' identification through selecting one relation and in the 2nd cut architecture for detecting the related processes (Yousef, 2010). The algorithm remains the required call to another algorithm that is responsible for identifying members of a given cluster. The second algorithm that appears in the Figures 6.9 and 6.10 remains the original purpose of identifying the members of the instantiated cluster. Similarly, the members are goal-based and quality-linked. During the search of the source and the destination in this algorithm, their goals must be extracted and added to the **Goal of the Cluster (GoC)** list considering that the algorithm is not interested in identifying the type of the member (i.e., CP or CMP). In addition, quality requirements of the member are extracted as well and added to the **Quality of Cluster** list. It is necessary to identify the quality requirements concerned only and not the whole quality model (e.g., NFR framework or SIG). The goal and quality lists are designed in order to assist in identifying the entire goal of the cluster and its quality requirements. The desired output from the two algorithms is the GQ-RPA clusters along with its GQ members.

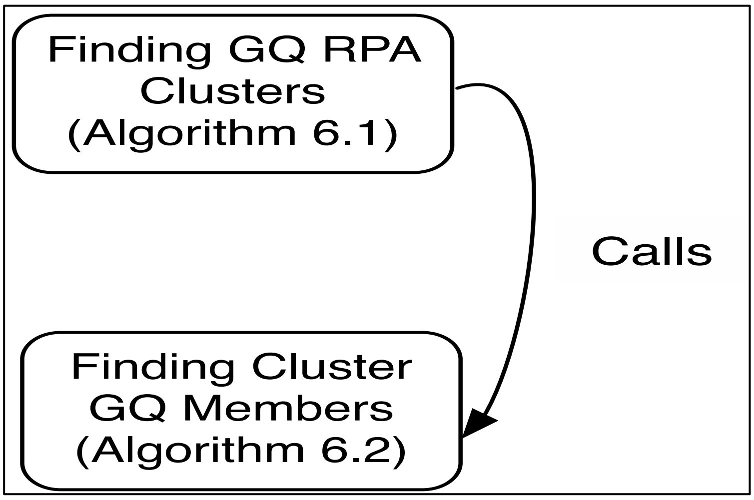
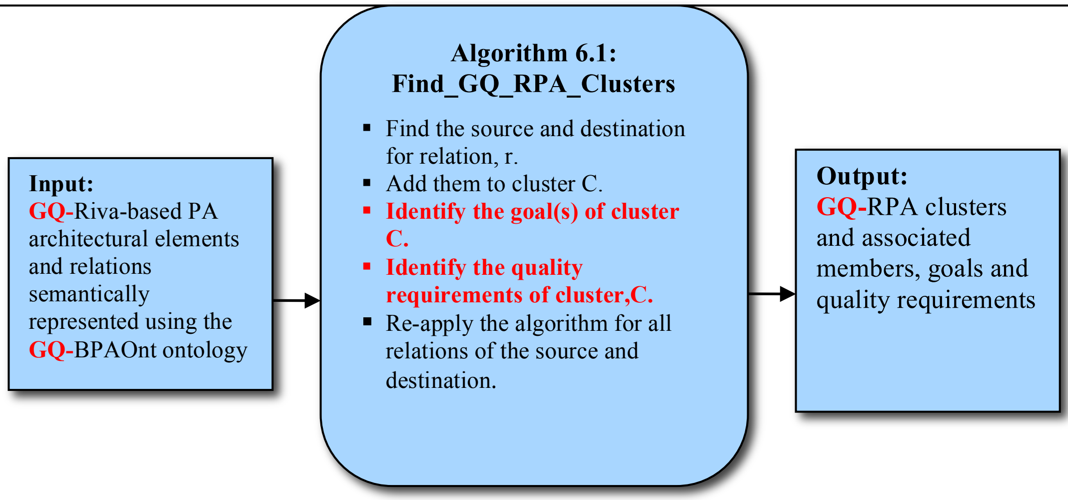


Figure 6.7: Two Algorithms for Deriving Connected GQ-RPA Clusters



Algorithm 6.1: Find_GQ-RPA_Clusters:

Input: *(1)* ***GQ-****BPAOnt ontology which imports an instantiated* ***GQ-****srBPA ontology, with all classes representing the Riva-based 2nd cut process architectural elements, and instances of each class. The classes are: PA2Elements and PA2Relations, representing the 2nd cut PA elements and the relations between them, respectively. (2) Global variables: checked_relations which is a list to keep those relations already checked in the whole PA diagram, and added_members, which is also a list used to keep those members already added from the whole PA diagram.* ***(3) Cluster variables****:* ***Goals_Of_Cluster keeps the goals of a cluster by deducing them from the goals of the identified members stored in the variable Members_Goals. Similarly, the quality requirements of a cluster, Quality_Of_Cluster, are derived from the values of the quality requirements of the members using the variable Members_Quality.***

Output*: the set of* ***GQ-****RPA clusters for the process architecture represented in the instantiated* ***GQ-****srBPA, identified as instances of class GQ_RPA_Cluster in the* ***GQ-****BPAOnt ontology, with the members of each cluster instance stored in the attribute hasMembers, goals of each cluster stored in the attribute hasClusterGoals and the desired quality requirements stored in the attribute hasClusterQualityRequirements.*

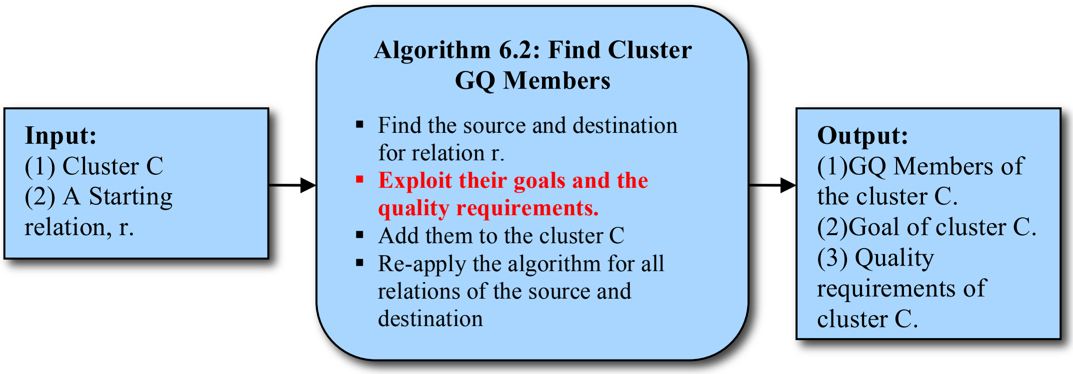
Begin
Find all relations, rel, in the 2nd cut PA Diagram: rel={r_0, r_1, …r_k, …, r_l}, $0 \le k \le l$,
For each relation r_k in rel,
Do

If r_k is not in checked_relaions and not an outside relation then

Goal_Of_Cluster=null. Quality_Of_Cluster=null.
Create an instance, c_x, of RPA_Clusterclass to be one of the RPA_Cluster class instances C, where C={c_0, c_1, …, c_x, …, c_m} $0 \le x \le m$
call Find_Cluster_GQ_Members, with cluster c_x and r_k passed with it.
c_x hasGoalOfCluster Goal_Of_Cluster
c_x hasQualityOfCluster Quality_Of_Cluster
End if

End do
End

Figure 6.8: Algorithm 6.1: Refining the Original Algorithm of Find_RPA_Clusters into Find_GQ-RPA_Clusters [Source: (Yousef, 2010), Adapted with the author's permission].



Algorithm 6.2: Find Cluster GQ Members:

Input: *(1) A cluster, C, which we want to identify its members, C is an instance of class GQ-RPA_Cluster with an attribute, hasMembers whose values are the members we identify. (2) A Starting relation, rel, from which we'll start the identification process. rel is an instance of class PA2Relation which has a source and a destination attributes attached to it. (3) Global variables: checked_relations which is a list to keep those relations already checked in the whole PA diagram, and added_members, which is also a list used to keep those members already added from the whole PA diagram. Members_Goals is a list of goals of the identified members and Members_Quality is a list of quality requirements of the identified members.*

Output: *the set of GQ-members belonging to cluster C, stored in the attribute hasMembers of the class instance C along with their goals and quality requirements.*

Begin
Define A and B as members of the PA2Element class;
A= source of rel; Goals_Of_A= list goals retrieved from A; Quality_Of_A= list of quality requirements of A.
B= destination of rel; Goals_Of_B= list goals retrieved from B; Quality_Of_B= list of quality requirements of B.
Add rel to checked_relations;
If B is not in added_members then,
    Add B to hasMembers for the class C
    Add B to added_members
    For each goal in Goals_Of_B
        If the goal is not added in the Members_Goals list
            Add the goal into Members_Goals list;
    End for each goal in B
    For each quality requirement in Quality_Of_B
        If the quality requirement is not added in the Members_Quality
            Add the quality requirement into Members_Quality list;
    End for each quality in B
    Goal_Of_Cluster= Goal_Of_Cluster U Members_Goals.
    Quality_Of Cluster= Quality_Of Cluster U Members_Quality.
    Find all relations, relB, connected to B: relB=$\{ra_0, ra_1, \ldots ra_i, \ldots, ra_n\}$*,* $0 \le i \le n$*,*
    For each relation ra_i *in relB*
    do
        If ra_i *is not in checked_relaions then*
            Recursively call Find_Cluster_members, with cluster C and ra_i *passed to it.*
        End if
    End do
End if
End

Figure 6.9: Algorithm 6.2: Refining the Original Algorithm of Find_RPA_Cluster Members into Find_GQ-RPA_Cluster GQ Members (Part 1 of 2) [Source: (Yousef, 2010), Adapted with the author's permission].

If A is not in added_members then
Add A to hasMembers for the class C
Add A to added_members
For each goal in Goals_Of_A
If the goal is not added in the Members_Goals list
Add the goal into Members_Goals list;
End for each goal in A
For each quality requirement in Quality_Of_A
If the quality requirement is not added in the Members_Qualit list
Add the quality requirement into Members_Quality list;
End for each quality in A
Goal_Of_Cluster= Goal_Of_Cluster U Members_Goals.
Quality_Of Cluster= Quality_Of Cluster U Members_Quality.
Find all relations, relA, connected to A: relA={ra_0, ra_1, ...ra_j, ..., ra_m}, $0 \leq j \leq m$,
For each relation ra_j in relB
do
If ra_j in not in checked_relaions then
Recursively call Find_members_of_cluster_through_relation, with cluster C and ra_i passed with it.
End if ra_i
End do
End if A
End

Figure 6.10: Algorithm 6.2: Refining the Original Algorithm of Find_RPA_Cluster Members into Find_GQ-RPA_Cluster GQ Members (Part 2 of 2) [Source: (Yousef, 2010), Adapted with the author's permission].

6.5 Identifying the Quality of Service

The identification of services involved determining the GQ-RPA clusters, their goals and associated quality requirements. However, this identification should not proceed in the detailed representation or description of the capabilities and the quality requirements, as each of them must be addressed using their corresponding component, as was early shown in Figure 6.2. However, it is necessary to recall that the non-functional boundary of the entity service, based on the new understanding, encapsulates the QoS requirements. Moreover, the non-functional requirements may derive further capabilities described here as the quality-driven capabilities. Such an example is the work of the NFR framework, which was used from Chapter 4 to Chapter 6, where a main NFR type is elaborated until possessing a set of dynamic operationalisations (i.e., capabilities) and/or static operationalisations (i.e., information) that propagate in the NFR framework in order to satisfy the main NFR.

Accordingly, the QoS component identifies the QoS requirements in the GQ-RPA clusters in addition to their detailed representation that may include the quality-driven capabilities using the GQOnt and GQ-BPAOnt ontologies. The presence of an appropriate NFR classification

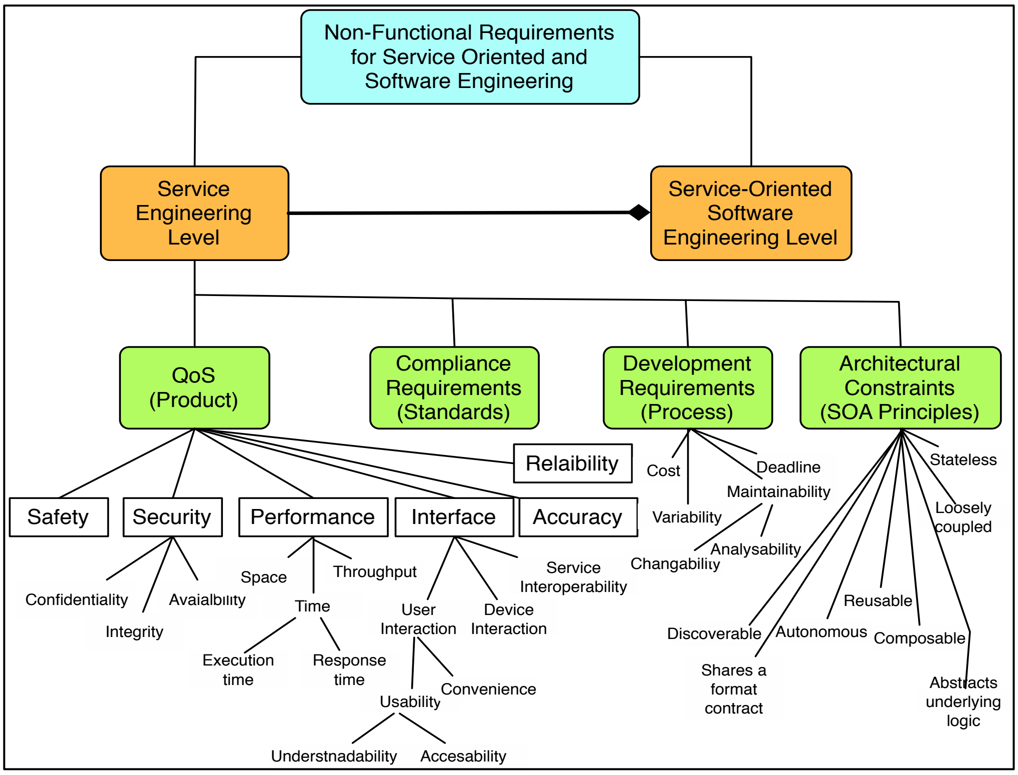


Figure 6.11: A New Non-Functional Requirements Multilevel Roadmap Taxonomy on Service-Oriented Software.

can support the function of this component through considering the QoS as one of the classified categories in order to appropriately classify the identified QoS requirements in this component. Figure 6.11 shows a NFR classification proposed in this research. The four green categories are designed as classes. However, the function of QoS identifier component is related only to the QoS category in Figure 6.11

The non-functional boundary of the entity service is driven from the GQ-BPAOnt ontology without involving the detailed semantic representation. This is because the GQ processes, that blueprint the BPA, associate with them the key quality requirements with their associated quality references if applicable. The identification of the QoS requirements is carried out in the BPA level rather than the BP one using the associated quality requirements in each GQ-CP and GQ-CMP. The quality-driven capabilities are derived from the associated quality models (e.g., NFR framework) in GQOnt ontology and particularly from the sQuality ontology. The SWRL rules that implement the derivation of key quality requirements as QoS and the derivation of the quality-driven capabilities are:

1) *GQ_RPA_Cluster(?c) ^ hasMembers(?c, ?cp) ^ hasQoP(?cp, ?q) ^ isElaborative(?q, false) → hasQoS (?c, ?q)*

2) *GQ_RPA_Cluster(?c) ^ hasMembers(?c, ?cp) ^ hasQoP(?cp, ?q) ^ isElaborative(?q, true) ^ hasQualityReference(?q, ?m) → hasQoS (?c, ?q) ^ hasQoSReference(?c, ?m)*

3) *GQ_RPA_Cluster(?c) ^ hasMembers(?c, ?cp) ^ hasQoP(?cp, ?q) ^ isElaborative(?q, true) ^ hasQualityReference(?q, ?m) ^ belongsToSIGDiagram(?d, ?m) ^ isDynamic(?d, true) → hasQualityDrivenCapability(?c, ?d)*

Using the UWE's CEMS Faculty of Administration, the QoS requirements are identified for each member (i.e., single service) using the GQ-Riva 2nd cut architecture, where their aggregation generates the QoS requirements of the cluster. The QoS requirements, of this pilot study, were previously shown in Table 6.1. It has been learned from Chapter 2 Section 2.2.2.2 that few quality-oriented approaches stimulate the derivation of functional requirements (i.e., operational capabilities) that fulfil the main quality requirement using the notion of elaboration. The quality-driven capabilities must be identified using an appropriate quality-oriented approach (e.g., the NFR framework) referenced with the QoS that represents the main NFR type. However, the NFR framework models are absent in this study and thereby the quality-driven capabilities are not identified. The proposed relation between the quality-driven capabilities and their candidate services is shown in Figure 6.12.

6.6 Identifying Service Capabilities

After the QoS requirements and their quality-driven capabilities (i.e., the non-functional boundary of a service) have been identified, it is now the turn to identify service capabilities that represent the functional boundary of the service. The work of this component is based on the remaining designed semantic relation between the GQ-srBPA and the GQ-sBPMN ontologies that stem from the BPAOntoSOA framework (Yousef, 2010). The original relation is slightly refined, due to the integration of goals and the quality requirements, and states that each CP and CMP in the GQ Riva-based 2nd cut architecture corresponds a semantic business process representation using the GQ-sBPMN ontology.

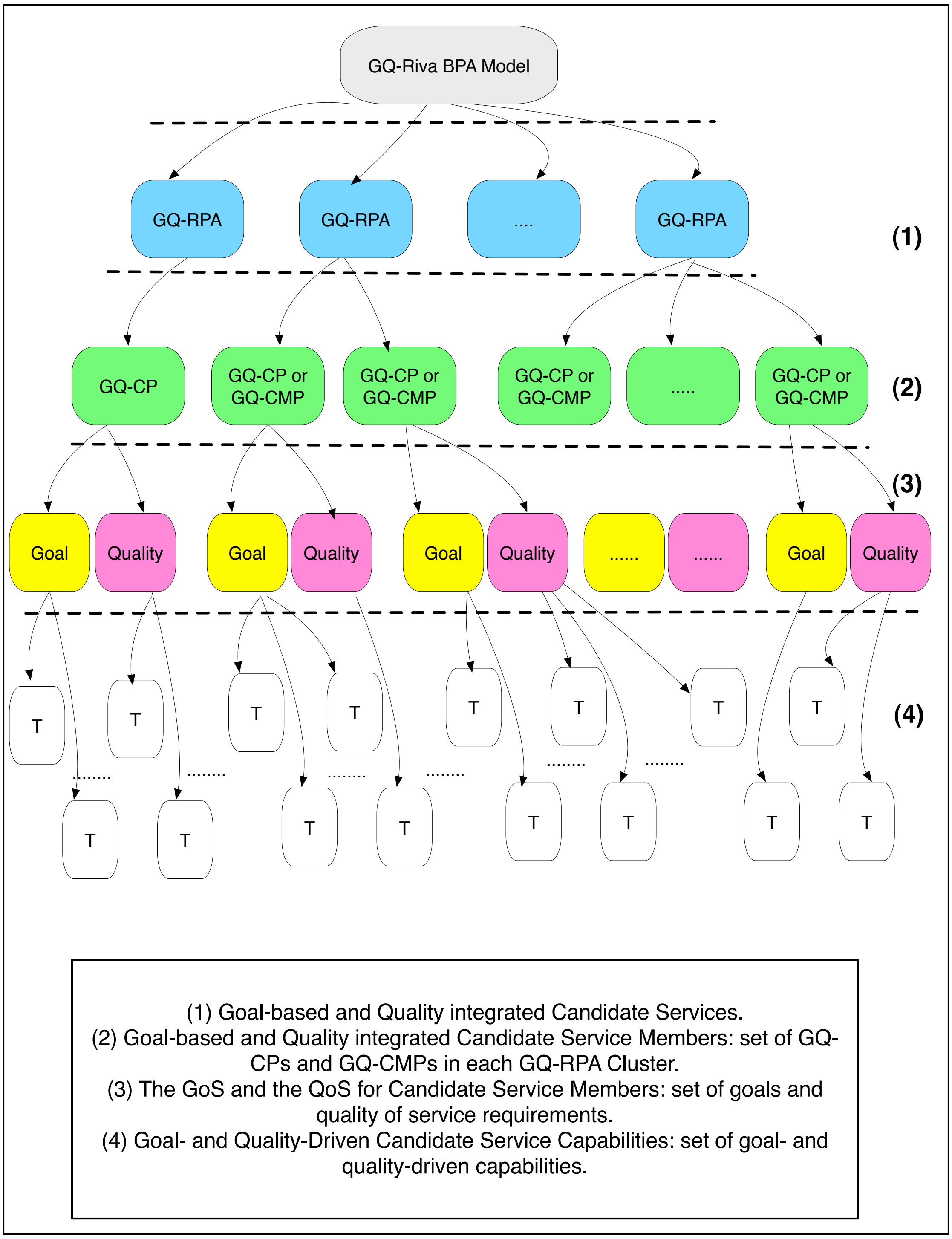


Figure 6.12: The relation between the goal-based and quality-integrated candidate services and their capabilities.

The original property hasCapability, created in the BPAOntoSOA framework, is reused, where its range is GQ_RPA_Cluster and its range the send task, and user task classes (Yousef, 2010). In particular, the original work is followed that states as below:

"We have defined the property hasCapabilities for each RPA cluster whose values are the set of *user* tasks and *send* tasks from sBPMN. From the sBPMN ontology design, we know that the class *Task* has the property, *hasGraphicalElementsProcess*, which is inherited form the class *Activity* which in turn is inherited from the class *FlowObject.* Class *Process* is related to both *CP* and *CMP* classes by the property *hasCorrespondingProcess.*" (Yousef, 2010) However, in this research, the property is for the GQ-RPA cluster taking into account that the resulted capabilities from the process must be implicitly goal-driven as shown in Figure 6.12. The original rules are adapted and shown in Figure 6.13, where the red text illustrates the slight required refinement.

Due to the absence of the business processes of the CEMS faculty of Administration study, the identification of capabilities are not carried out. Therefore, the image of this component will be clearer after evaluating the work using the CCR case study that consists of the Riva BPA and associated BPMs for the comprehensive evaluation.

GQRPA_Cluster(?C) ^ CP(?cp)^ hasMembers (?C, ?cp) ^ hasCorrespondingProcess (?cp, ?p) ^ sendTask (?st) ^ hasGraphicalElemetsProcess (?p, ?st) ^ -> hasCapability(?C,?st)

GQRPA_Cluster(?C) ^ CP(?cp)^ hasMembers (?C, ?cp) ^ hasCorrespondingProcess (?cp, ?p) ^ UserTask (?ut) ^ hasGraphicalElemetsProcess (?p, ?ut) ^ -> hasCapability(?C,?ut)

GQRPA_Cluster(?C) ^ CMP(?cmp)^ hasMembers (?C, ?cmp) ^ hasCorrespondingProcess (?cmp, ?p) ^ sendTask (?st) ^ hasGraphicalElemetsProcess (?p, ?st) ^ -> hasCapability(?C,?st)

GQRPA_Cluster(?C) ^ CMP(?cmp)^ hasMembers (?C, ?cmp) ^ hasCorrespondingProcess (?cmp, ?p) ^ UserTask (?ut) ^ hasGraphicalElemetsProcess (?p, ?ut) ^ -> hasCapability(?C,?ut)

Figure 6.13: The adapted SWRL Rules that set the capabilities using the hasCapability relation [Source: (Yousef,2010), Adapted with the author's permission].

6.7 Summary and Conclusion

This chapter addressed the work wrapped within the third and final layer in the proposed GQ-BPAOntoSOA framework. The chapter separately presented the function of each component in the GQ software SI layer with a special attention in highlighting the original work adopted from the BPAOntoSOA framework (Yousef, 2010) and the extension proposed regarding the integration of goals and quality requirements.

For the service identification component, the notion of composing the goals of the cluster members is carried out in order to identify the goal of the entire cluster. With regard to the quality requirements, the notion of composition is adopted too in order to identify the quality requirements of the cluster from its members. This was validated using the CEMS Faculty Administration pilot study.

In this chapter, the original SI approach has been refined through engaging goals and the quality requirements (GQ-SI approach) in order to meet the aim of this research. The refined SI approach is based on the GQ-RPA clusters rather than solely on the RPA cluster. The GQ-RPA cluster remained with the simplicity addressed by the original BPAOntoSOA framework through using the GQ-Riva 2nd cut architecture in identifying the candidate services. This is because the GQ-Riva remained simple even after the enriching it with goals and quality requirements. A candidate service is either a standalone (GQ-CP) or a related GQ-CP and GQ-CMP. Due to this enrichment to the original RPA cluster, the definition of the entity service is refined and involves the non-functional boundary next to the functional one. The proposed GQ-SI approach has satisfied most of the SOA principles using the GQ-BPAOnt. This has demonstrated the identification of the GQ-RPA clusters, using the UWE's CEMS Faculty of Administration. However, the pilot study is limited in providing its GQ-BPMs and the associated NFR framework models, as they are both absent. Therefore, the QoS and the service capabilities were not illustrated in the validation, but they are shown in the comprehensive evaluation using CCR case study in Chapter 7.

The GQ-BPAOnt, which comprises the GQ-Riva 2nd cut architecture, GQ-BPMs and associated NFR framework models, is a requirement in order to carry out the GQ-SI approach following by this the footsteps of the BPAOntoSOA framework (Yousef, 2010). The significance of the GQ-BPAOnt is attributed to the following:

1- The GQ-BPAOnt forms a rich semantic repository of the required information that assist in meeting the desired definition of the entity service and thereby identifying the candidate services using the GQ-RPA clustering approach. The GQ-RPA cluster from the GQ-BPAOnt steer the way to identify the QoS requirements and its quality-driven capabilities via the GQOnt ontology. The quality-driven capabilities join the already identified capabilities using the BPAOntoSOA framework.

2- The application of the GQ-RPA clustering approach in the GQ-BPAOnt in order to identify the candidate services has satisfied the relevant SOA principles for this work.
3- The GQ-BPAOnt has enhanced the automatic SI process proposed in (Yousef, 2010) through involving goals and the quality requirements.

The new QoS identifier component in the GQ-BPAOntoSOA framework has the responsibility of identifying the additional quality-driven capabilities, using the NFR framework, that in turn enrich the already identified capabilities using the GQ-BPMs. Moreover, the quality requirements on the candidate service are not only concerned with the service as a business-driven product, yet they are also concerned with the service development process, the service compliance to organisation's standards and regulations and finally the satisfaction of the SOA principles. However, the QoS category is the only interesting category within the aim of this research, where the remained categories are out the scope.

Finally, the work of this chapter contributes to answer research questions flagged in Figure 6.14. In addition, this chapter activates research process phases shown in Figure 6.15.

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1						✓			
RQ2									
RQ3						✓			
RQ4						✓			

Figure 6.14: Answering Research Questions Using the Work of Chapter 6.

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									**Activated**
RP2						✓			**Active**
RP3									**Activated**
RP4(ai)									**Activated**
RP4(aii)						✓			**Active**
RP4(bi)									**Activated**
RP4(bii)									**Activated**
RP4(biii)						✓			**Active**
RP5									
RP6									

Figure 6.15: Addressing Research Process Phases Using the Work of Chapter 6.

Chapter Seven: Evaluating the GQ-BPAOntoSOA Framework Using the CCR Processes Case Study

7.1 Introduction

The previous three chapters have separately dealt with each layer in the GQ-BPAOntoSOA framework. It is required at this stage of research to instantiate the entire framework using the Cancer Care and Registration (CCR) Process case study modelled for King Hussein Cancer Care Centre in Jordan in (Aburub, 2006) in order to evaluate this research work. It is important to denote that part of the GQ-BPAOntoSOA framework has been applied in (Yousef, 2010) using the same CCR case study. Thus, it is necessary to reuse the same case study applied in the original BPAOntoSOA framework in order to investigate the implication of integrating business goals and quality requirements. Therefore the relevant results are reused. In addition, this chapter illustrates the assessment of the entire GQ-BPAOntoSOA Framework in terms of meeting the requirements identified in Chapter 3 (Section 3.5.2) and assessing its functionality.

The chapter is structured as follows: Section 7.2 introduces the concern-based approach of (Kotonya and Sommerville, 1998) that is used in this chapter to assist in answering the research questions. Section 7.3 introduces the research evaluation framework that involves an application of the GQ-BPAOntoSOA framework using the CCR case study. The use of DSR methodology is assessed briefly before presenting the evaluation framework. Also, this section employs the concern-based approach as part of the evaluation of this research in order to answer the research questions that lead to proving or disproving the research hypothesis at the end.

7.2 The Concern-based Approach

The research roadmap for evaluation is represented using the concern-based approach, which was initially proposed by (Kotonya and Sommerville, 1998) as an attempt to derive the system requirements from its desired NFRs. The concern-based approach was found an appropriate approach for the research evaluation purposes in (Yousef, 2010) and (Khan, 2009). The approach can be described as a simple approach in reflecting functional and non-functional requirements of a system based on stakeholders main concerns and sub-concerns. In simple words, the concern-based approach directs the elicitation from the stakeholder's

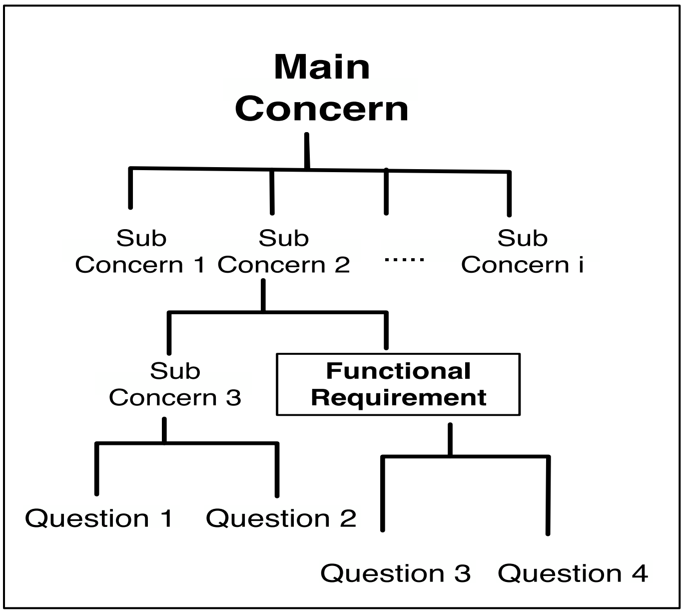


Figure 7.1: A Simplified Template of Kotonya's and Sommervillies's Concern-based Approach

main concern that represents a high-level objective, an NFR that is an essential characteristic of the system, critical business objective, or even a holistic requirement of the entire system (Kotonya and Sommerville, 1998). The concern is decomposed into sub-concerns, where each may derive few concern-related questions immediately. Also, Kotonya and Sommerville suggested another way for deriving these questions through refining concerns into system requirements that in turn are decomposed into questions. An illustration of the concern-based approach is depicted in Figure 7.1.

7.3 The Research Evaluation Framework

This section introduces the research evaluation framework in order to pave the way answering the research questions and thereby dis/approving the research hypothesises presented in Chapter 1 (Section 1.4). In particular, a comparison between the outputs from the instantiation of the GQ-BPAOntoSOA framework, and the ones derived using the original BPAOntoSOA framework paves the way to inform the effectiveness of the extended framework. The *effectiveness* of the GQ-BPAOntoSOA framework is informed by the capability of the new framework in deriving GQ SOA-able candidate software services, its associated capabilities and QoS requirements. Before presenting the research evaluation framework, it is necessary to assist the use of the DSR methodology in this research. This assessment is carried out using two criteria:

(1) The research process phases. This refers to whether or not the designed research process phases in Chapter 3 (Section 3.4) fit or match with the DSR method steps. It is possible addressing this criterion through investigating the relation of each research process phase with respect to the DSR method steps (Yes/No criterion).

Recalling the research process phases in Chapter 3, it is apparent that the designed research process for this research fits within the DSR.

a) The preliminary phase: steps 1 and 2 in DSR method.
b) The early theoretical framework design phase: step 3 in DSR method.
c) The investigation phase: steps 3 and 4 in DSR method.
d) The original BPAOntoSOA framework enhancement phase: step 3 in DSR method.
e) The conceptual framework development phase: step 3 in DSR method.
f) The application and evaluation phase: steps 4, 5 and 6 in DSR method.

(2) Outputs of this research with the reference to the DSR. (This is a qualitative criterion that employs the Vaishnavi & Kuechler Jr. (2007) definition of DSR's output).

a) Constructs: suggested enriching the Riva-based SI with respect to the integration of goals and quality requirements into the original BPAOntoSOA framework.
b) Models: an extended BPAOntoSOA framework, that is goal-based and quality-linked, namely the GQ-BPAOntoSOA framework.
c) Methods: (i) proposed a new representation of a BSV, (ii) proposed a novel alignment approach that bridges the gap between a BSV, Riva BPA and its associated BPMs and (iii) proposed a refined SI approach, that is goal-based and quality-linked.
d) Instantiations: SOA-able candidate software services generated from an alignment of a Riva BPA to a BSV after operationalising the GQ-BPAOntoSOA framework.

It is apparent that the nature of this research fits well with the DSR methodology with regard to its process and outputs.

With regard to the evaluation framework, it is multi-objective levelled based on specific features and outputs of the layers in the GQ-BPAOntoSOA framework considering that a case study (i.e., the CCR case study (Aburub, 2006)) is chosen as a research strategy to evaluate the work. Figure 7.2 shows the research evaluation framework, where its levels' objectives are listed respectively as below.

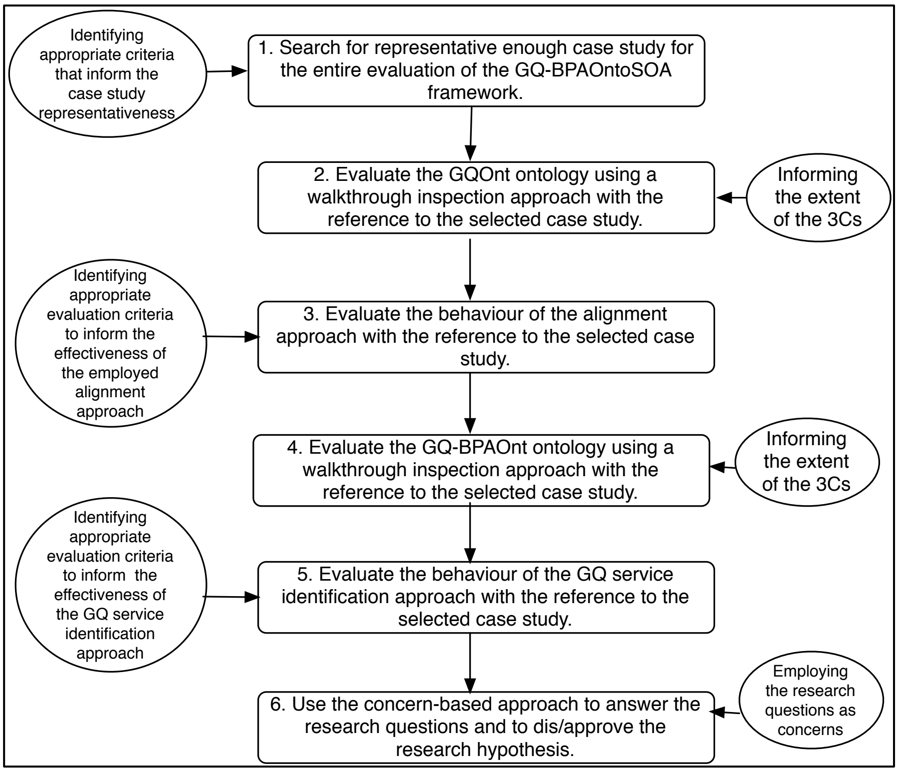


Figure 7.2: The Research Evaluation Framework

1) To assess the selection of a representative case study for the instantiation and evaluation of the GQ-BPAOntoSOA framework. This assessment is based on particular criteria, as presented in Section 7.3.1. Also, this section involves applying the GQ-BPAOntoSOA framework using the selected processes in order to start evaluating the framework as below.

2) To inform the extent of the correctness, completeness and consistency of the GQOnt ontology using the selected case study. This objective is addressed in Section 7.3.2. This objective is needed in order assess the semantic representation of a BSV for an organisation within the extended semantic framework. This is important in order to ensure the sematic feature of the GQ-BPAOntoSOA framework.

3) To inform the effectiveness of the alignment approach using the selected case study. The alignment approach is assessed in Section 7.3.3. This is needed in order to assess the representation of the GQ-Riva BPA and its associated BPMs, as both of them form the base for the identification of the software services.

4) To inform the correctness, completeness and consistency of the GQ-BPAOnt ontology using the selected case study. Section 7.3.4 shows addressing this objective. This is needed in order to ensure the semantic representation of the GQ-BPAOnt

before starting the identification of the GQ software services. This is because it is not right to start the semantic identification of GQ clusters (i.e., candidate software services) before assessing the base of these clusters.

5) To inform the effectiveness of the SI approach employed in the GQ-BPAOntoSOA framework using the selected case study. This objective is addressed in Section 7.3.5. This objective is very important as it shows the results from the original framework and the extended one. Accordingly, the SI effectiveness is informed and the GQ-BPAOntoSOA framework as well.

6) To answer the research questions using the concern-based approach (Kotonya and Sommerville, 1998) with the reference to the selected case study in order to dis/approve the research hypothesis. Section 7.3.6 illustrates the concern-based model as a road map to guide answering the questions in order to dis/approve the research hypothesis. Research questions are answered using the work in this chapter with the support of the work of previous chapters.

7.3.1 Assessing on the Selection of Representative Case Study

The CCR case is the first candidate case study for the evaluation purposes of the GQ-BPAOntoSOA framework, as it was employed in the evaluation of the GQ-BPAOntoSOA framework. However, few criteria are identified in order to select representative enough processes from the CCR case study in order to employ them for the evaluation. The CCR process is composed of six business processes that are collaborating in order to address the main objectives of the CCR. Table 7.1 shows the six processes of the CCR process with the identified criteria. A representative case study for the evaluation of the GQ-BPAOntoSOA framework is determined using the criteria:

1) A case study should has been used in the evaluation of the original BPAOntoSOA framework. This criterion is important in order to compare the resultant original outcomes (i.e., using the original framework) with the new ones using the GQ-BPAOntoSOA framework. The criterion considers, by default, the presence of the Riva BPA, its associated BPMs, candidate software services and their associated capabilities for the candidate case study.

2) A case study should take into the account addressing business goals. This is because business goals are integrated into the GQ-BPAOntoSOA framework.

3) A case study should take into account the integration of quality requirements into its models. Also, it should involve a representation of its quality-oriented models. This criterion is required, as the quality requirements conceptualisation for an organisation are integrated into the GQ-BPAOntoSOA Framework.

4) A case study should consider the participation of actors or roles (i.e., human or non-human).

5) A case study should avoid redundancy of data (e.g., redundant quality-oriented models).

6) A case study should consider the interaction of its business processes with respect to addressing a main objective. Also, this criterion should take into the account that a business process and its siblings were employed in the evaluation of the BPAOntoSOA framework.

Table 7.1 shows how the six processes address the second, the third and the fourth criteria. However, it is apparent that the cancer registration process does not address the first criterion. Therefore, the cancer registration process is excluded from the evaluation of the GQ-BPAOntoSOA framework. Also, Table 7.1 shows that the patients follow up process involve redundant data, as it is using the same quality-oriented models of the patient treatment process in (Aburub, 2006). Since the caner registration process is a sibling to the hospital registration process in order to address the main objective “improving cancer data collection and classification”, where the former process is already excluded, then the latter is excluded too. Table 7.1 shows that three processes in the CCR case study meet the six criteria above and are considered as representative enough for the aimed evaluation: patient reception process, cancer detection process and cancer treatment process. In this chapter, the CCR process term will be used to generally represent the three processes in order to shorten their description.

The selection of the appropriate processes is necessary in order to start applying the GQ-BPAOntoSOA framework that begins with deriving the BSV for the CCR case study (i.e., the selected three processes in Table 7.1) using the work of Chapter 4. The BSV for the CCR is attached in Appendix M. The GQ-BPAOnt ontology instantiation layer (i.e., second layer) in the GQ-BPAOntoSOA framework is applied using the inputs: the derived BSV for the CCR and its corresponding Riva-BPA designed using the original method (Ould, 2006). The application of this layer is attached in appendices N, O, P, Q and R. The software service identification layer is attached in Appendix S. The reader can notice that the application of the GQ-BPAOntoSOA framework using the CCR case study is attached in appendices due to the large space needed for representation.

Table 7.1: The Criterion Used to Select A Representative Case Study from the CCR Processes.

CCR Business Process	Used in the Evaluation of the Original BPAOntoSOA Framework?	Goal-Based	Quality-Linked and has Associated Quality models	Actor or Role-Oriented	Unique quality oriented models	Having at Least One Sibling Process and was Evaluated Using the Original BPAOntoSOA Framework
P1: Patient Reception Process	✓	✓	✓	✓	✓	✓
P2: Cancer Detection Process	✓	✓	✓	✓	✓	✓
P3: Cancer Treatment Process	✓	✓	✓	✓	✓	✓
P4: Patients Follow up Process	✓	✓	✓	✓	✖	✓
P5: Hospital Registration process	✓	✓	✓	✓	✓	✖
P6: Cancer Registration Process	✖	✓	✓	✓	✓	✖

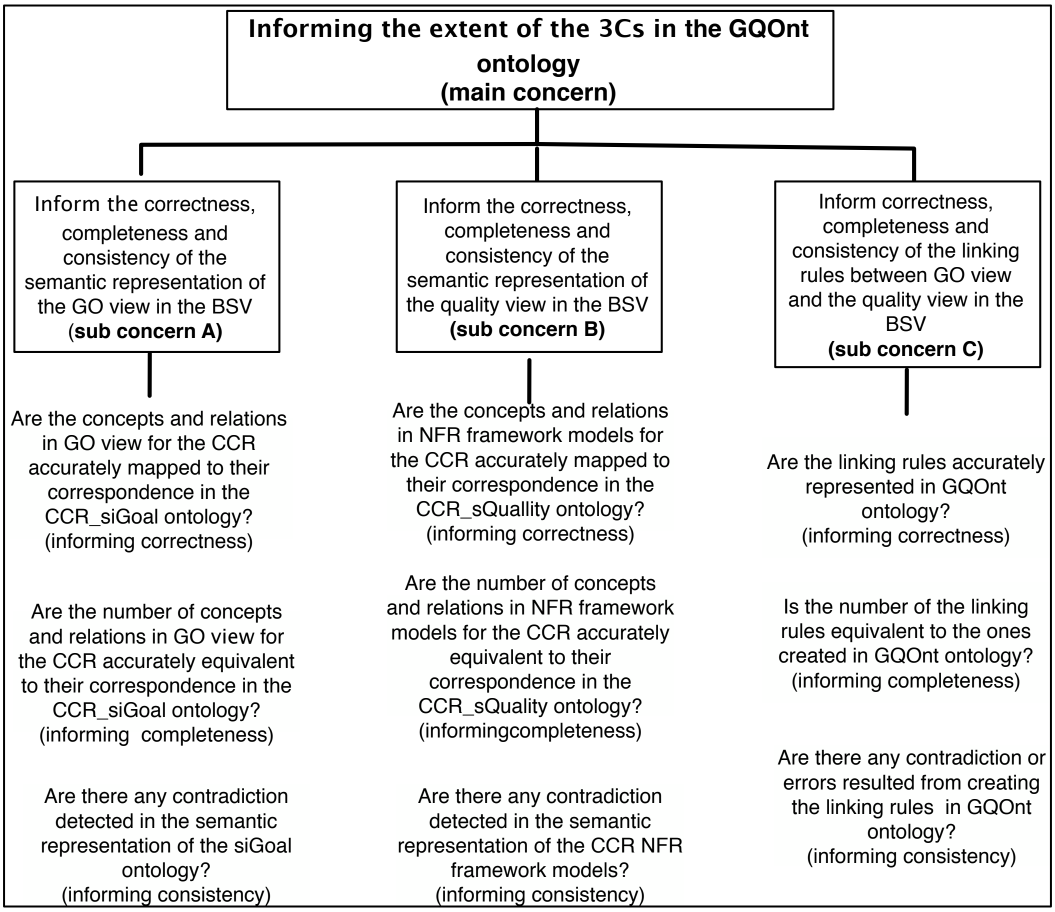


Figure 7.3: The Concern-based Diagram for Addressing the 3Cs of the GQOnt Ontology.

7.3.2 Informing the Extent of the Correctness, Completeness and Consistency of the GQOnt Ontology Using the Representative Case Study

This section represents the second-level in the evaluation framework that aims at informing the 3Cs (Correctness, Completeness and Consistency) of the instantiation of the GQOnt ontology using the representative case study. However, this first requires deriving the BSV for the selected case study in order to instantiate it using the GQOnt ontology. Since the GQOnt ontology is comprised of the three ontologies, then they are used to address the aim of this level. This section informs extent of the 3Cs of siGoal ontology and sQuality ontology using the GO view and its semantics, NFR framework and its semantics, as presented in (Sections 4.3.1 and 4.3.2). Informing the 3Cs of the GQOnt ontology is necessary, as the alignment approach is based on the GQOnt and GQ-BPAOnt ontologies. Figure 7.3 shows a roadmap that is designed using the concern-based approach in order to direct addressing the aim of this section.

The 3Cs values for each concern in Figure 7.3 are shown in Appendix T. Using the three tables attached in Appendix T, a researcher can derive observations for each for the three ontologies as below.

1- With regard to the siGoal ontology (i.e., sub-concern A in Figure 7.3), all elements and relationships designing the CCR GO view are accurately captured and semantically instantiated using the siGoal_CCR ontology. Hence this informs addressing correctness to some great extent. Also, the completeness is addressed, as the number of elements and relationships designed in the CCR GO view are equivalent to the ones in the corresponding semantic representation, namely siGoal_CCR ontology. Finally, no errors were detected regarding the consistency of the semantic representation. This is informed after running the built-in reasoner in Protégé (i.e., Pellet 1.5.2). Collectively, this informs addressing the 3Cs to some great extent for the siGoal_CCR ontology. Table T.1 in Appendix T presents the 3Cs values for the siGaol ontology evaluation.

2- With regard to the sQuality ontology (i.e., sub-concern B in Figure 7.3), all elements and the relationships that create the CCR's quality view are accurately mapped to their corresponding semantic representation in the sQuality_CCR ontology. In addition, the number of elements and relationships presented in the CCR NFR framework models are equivalent to the ones semantically presented in sQuality_CCR. Errors are free, where there is no consistency problems occurred while running the reasoner. Accordingly, this informs addressing the 3Cs to some great extent as a required part of the evaluation of the GQOnt ontology.

3- With regard to the linker (i.e., sub-concern B in Figure 7.3), the second and the third linking cases were captured within the CCR BSV that are implemented and instantiated correctly in the GQOnt_CCR ontology. This is because the first and the fourth linking cases were not needed in this case study. The completeness is addressed to a great extent, as the number of the carried out linking are equivalent to the ones in the GQOnt_CCR ontology, where there was no error captured while checking the consistency of the GQOnt_CCR ontology.

7.3.3 Informing the Effectiveness of the Alignment Approach Using the Representative Case Study

This section aims at showing the effectiveness of the alignment approach that aligns the BSV (i.e., employs the *i** and the NFR frameworks) of an organisation with its current Riva BPA with the reference to the CCR selected processes. The BSV for the case study is shown in Appendix M, where the as-is Riva BPA is shown in Appendix N (Section N.1). The GQ-Riva BPA generated model must comprise of goal-based and quality-integrated processes that are both traceable to and from the BSV. The *effectiveness* of the alignment approach refers to the capability of the alignment approach in producing a Riva BPA that addresses the up-to-date business goals and quality requirements. This may be manifested in the entire chunking of processes in the Riva BPA.

The effectiveness of the alignment approach is informed after conducting a comparison between the CCR Riva BPA model before and after the alignment. In addition, the effectiveness of the *i** framework and the NFR framework employed in the BSV is also informed.

Since the alignment approach is concerned with the BSV, Riva BPA and associated BPMs, then informing the effectiveness is addressed through walking through a number of criteria that informs the changes happened because of the carried out alignment. It is necessary to highlight that the criteria are identified with regard to the Riva BPA and the generated BPMs. Table 7.2 defines a number of criteria for the Riva BPA where its first and second row refer to the before (shaded in grey colour) and after (in white colour) defined criteria. The final row in Table 7.2 fills the associated criteria with the required quantitative or qualitative value based on the applied CCR representative processes. Table 7.3 defines criteria before and after the alignment from the BP's point of view. Similarly, the criteria are filled with the associated appropriate quantitative or qualitative value based on the selected CCR processes.

The following criteria are defined to investigate the alignment implication on ***the Riva BPA***:

1- Number of before and after EBEs. (Quantitative criterion value)
2- Number of before and after UoWs. (Quantitative criterion value)
3- Number of before and after processes in the resulted 2nd cut architecture. (Quantitative criterion value)
4- Alignment implication on the Riva method itself. (Qualitative criterion value)

With regard to investigating the alignment on the ***associated BPMs***, the following criteria are defined:

1- Determine the before and after processes status with regard to its integration of goals and quality requirements. (Yes/No criterion value).
2- The alignment implications on the process with regard to the required BP design effort. This criterion has one out of three value (i.e., novel process with full BP design, as-is process with required redesign and as-is process with no required redesign). (Qualitative criterion value).

Also, the effectiveness of the alignment approach is informed through informing the effectiveness of the employed *i** and the NFR frameworks using two criteria that are:
1- The effectiveness of the newly derived EBEs (Qualitative criterion).
2- Ability to derive new concepts needed in the Riva method (Qualitative criterion).

Table 7.2: Before and After the Alignment Riva BPA Criteria for the CCR Representative Processes

Before the Alignment Criteria (Riva BPA Point of View)			After the Alignment Criteria (Riva BPA Point of View)							
Number of as-is EBEs	Number of UoWs	Number of Processes in the 2nd Cut Architecture	Number of Overall to-be EBEs	Number of Detected EBEs	Number of New EBEs	Number of Overall to-be UoWs	Number of Detected UoWs	Number of New UoWs	Number of Processes in the 2nd cut Architecture (new/as-is)	Alignment Implication on the Riva Method or BPA
52	11	11	77	52	25	18	11	8	18 (8/11)	1- It allowed classifying the Riva elements into new and detected (reused). 2- It allowed reusing the as-is Riva BPA elements and models in order to address the BSV.

Table 7.3: Before and After the Alignment BPM Criteria for the CCR Representative Processes

Before the alignment (process point of view)			After the alignment (process point of view)					
As-is process name	Was it goal-based?	Was it quality-linked?	To-be process	Is it goal-based now?	Is it quality-linked now?	Addressing/linked to the BSV via goals	Is it unique between the generated processes?	Alignment implication on the process
1-Handle a patient general reception process	No	No	1-Handle a patient reception process	Yes	Yes	GoP: Patient reception Fulfills: 1- Improving administration of cancer treatment.	Yes	Redesigning
null	null	null	2-Handle an appointment booking	Yes	Yes	GoP: Appointment booking Contribute to fulfil: 1-Improving administration of cancer treatment. 2-Patient general reception	Yes	Novel
null	null	null	3-Handle a patient registration	Yes	Yes	GoP: General patient reception diagnose is checked. Contribute to 235ulfil: 1-Improving administration of cancer treatment 2- Patient general reception	Yes	Novel
null	null	null	4-Handle a general reception patient diagnose check	Yes	Yes	GoP: Non-diagnosed patient is reception. Contribute to fulfill: 1-Improving administration of cancer treatment 2- Patient general reception	Yes	Novel

Table 7.3 (Cont'd): Before and After the Alignment BPM Criteria for the CCR Representative Processes

Before the alignment (process point of view)			After the alignment (process point of view)					
As-is process name	Was it goal-based?	Was it quality-linked?	To-be process	Is it goal-based now?	Is it quality-linked now?	Addressing/linked to the BSV via goals	Is it unique between the generated processes?	Alignment implication on the process
2- Handle a cancer detection	No	No	5-Handle a cancer detection	Yes	Yes	GoP : Non-diagnosed patient is reception. Contribute to fulfill: 1-Improving administration of cancer treatment	Yes	Redesigning
null	null	null	6-Handle a patient diagnose	Yes	Yes	GoP : Patient is diagnosed for cancer detection. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer detection.	Yes	Novel
null	null	null	7-Handle the identification of cancer type and site	Yes	Yes	GoP : Cancer type and site are determined. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer detection.	Yes	Novel
3-Handle a patient treatment	No	No	8-Handle a patient treatment	Yes	Yes	GoP: Patient reception for cancer treatment. Contribute to fulfill: 1-Improving administration of cancer treatment	Yes	Redesigning

Table 7.3 (Cont'd): Before and After the Alignment BPM Criteria for the CCR Representative Processes

Before the alignment (process point of view)			After the alignment (process point of view)					
As-is process name	Was it goal-based?	Was it quality-linked?	To-be process	Is it goal-based now?	Is it quality-linked now?	Addressing/linked to the BSV via goals	Is it unique between the generated processes?	Alignment implication on the process
null	null	null	9- Handle a patient diagnose check	Yes	Yes	GoP: Patient diagnose is checked. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Patient treatment	Yes	Novel
null	null	null	10- Handle an adequate treatment plan	Yes	Yes	GoP: Adequate treatment is received. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Patient treatment.	Yes	Novel
4- Handle a lab test	No	No	11-Handle a lab test	Yes	Yes	GoP:: a lab test is performed Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient diagnose.	Yes	No required redesign
5- Handle an imaging test	No	No	12-Handle an imaging test	Yes	Yes	GoP:: an imaging test is performed Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient diagnose.	Yes	No required redesign

Table 7.3 (Cont'd): Before and After the Alignment BPM Criteria for the CCR Representative Processes

Before the alignment (process point of view)			After the alignment (process point of view)					
As-is process name	Was it goal-based?	Was it quality-linked?	To-be process	Is it goal-based now?	Is it quality-linked now?	Addressing/linked to the BSV via goals	Is it unique between the generated processes?	Alignment implication on the process
6- Handle a patient medical records	No	No	13-Handle a patient medical records	Yes	Yes	GoP:: Medical record of patient is handled. Contribute to fulfil: 1-Improving administration about cancer treatment	Yes	No required redesign
7- Handle an outpatient-clinic reception	No	No	14- Handle an outpatient-clinic reception	Yes	Yes	GoP: outpatient clinic reception. Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Cancer detection	Yes	No required redesign
8- Handle a chemother-apy treatment	No	No	15- Handle a chemother-apy treatment	Yes	Yes	GoP: Patient chemotherapy treated. Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient treatment 3-Adequate treatment plan is received	Yes	No required redesign

Table 7.3 (Cont'd): Before and After the Alignment BPM Criteria for the CCR Representative Processes

Before the alignment (process point of view)			After the alignment (process point of view)					
As-is process name	Was it goal-based?	Was it quality-linked?	To-be process	Is it goal-based now?	Is it quality-linked now?	Addressing/linked to the BSV via goals	Is it unique between the generated processes?	Alignment implication on the process
9- Handle a radiotherapy treatment	No	No	16- Handle a radiotherapy treatment	Yes	Yes	GoP: Patient radiotherapy treated. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer treatment. 3- Adequate treatment plan is received	Yes	No required redesign
10- Handle an inpatient care	No	No	17- Handle an inpatient care	Yes	Yes	GoP: Inpatient care. Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient treatment	Yes	No required redesign
11- Handle a patient admission	No	No	18- Handle a patient admission	Yes	Yes	GoP: patient is admitted. Contribute to fulfill: 1-Improving collection about cancer cases	Yes	No required redesign

The observations from the evaluation outcomes are as follows. First, for each criterion; a remarkable change has been noticed before and after the alignment for the entries in the Tables 7.2 and 7.3. And this change appeared because of the carried out alignment and particularly because of the concept mapping performed through exploiting needed Riva objects from the BSV. Second, the criteria support addressing and assuring consistency through classifying the EBEs in Table 7.2 and through determining uniqueness of the resulted processes in Table 7.3.

From Table 7.2, it is apparent how the number of EBEs has increased (i.e., from 52 to 77) after the alignment with the BSV. Since an EBE definition does not refer to any quality perspective, the reader should notice the effectiveness of the *i** framework in not only generating new EBEs (i.e., 25 new EBEs) but also involving the original ones (i.e., as-is EBEs). Also, it is noticed that the 52 as-is EBEs are subset of the 77 to-be EBEs as shown in Appendix N. In addition, some of the resulted new EBEs from the BSV and particularly from *i** framework (e.g., appointment booking) turned into new UoWs (i.e., 8 new UoWs filtered from the 25 new EBEs) and each resulted a new process (e.g., handle a patient appointment booking CP in Table 7.3) in the Riva 2nd cut architecture. Hence, this brings a new knowledge to the business analyst and the organisation and assists in improving design decisions. New EBE that is UoW means new process(es) in the Riva-BPA that leads to new RPA cluster(s) or new member(s) of a preexisting RPA cluster considering that a RPA cluster is a candidate SOA software service. The significance of the new EBEs that are new UoWs to the RPA clusters is discussed in in Section 7.3.5 with examples shown in Tables 7.4 and 7.5. In short words, the number of original RPA clusters is increased from 10 to 17 because of the new EBEs and UoWs. One can observe how the entire BPs (i.e., neither goal-based nor quality-linked) before the alignment are all GQ-BPs after the alignment. The effectiveness of the *i** framework and the alignment is shown in Figure 7.4. A very important change appears in the GQ-Riva BPA that comprises a related GQ-BPs that classified into three categories:

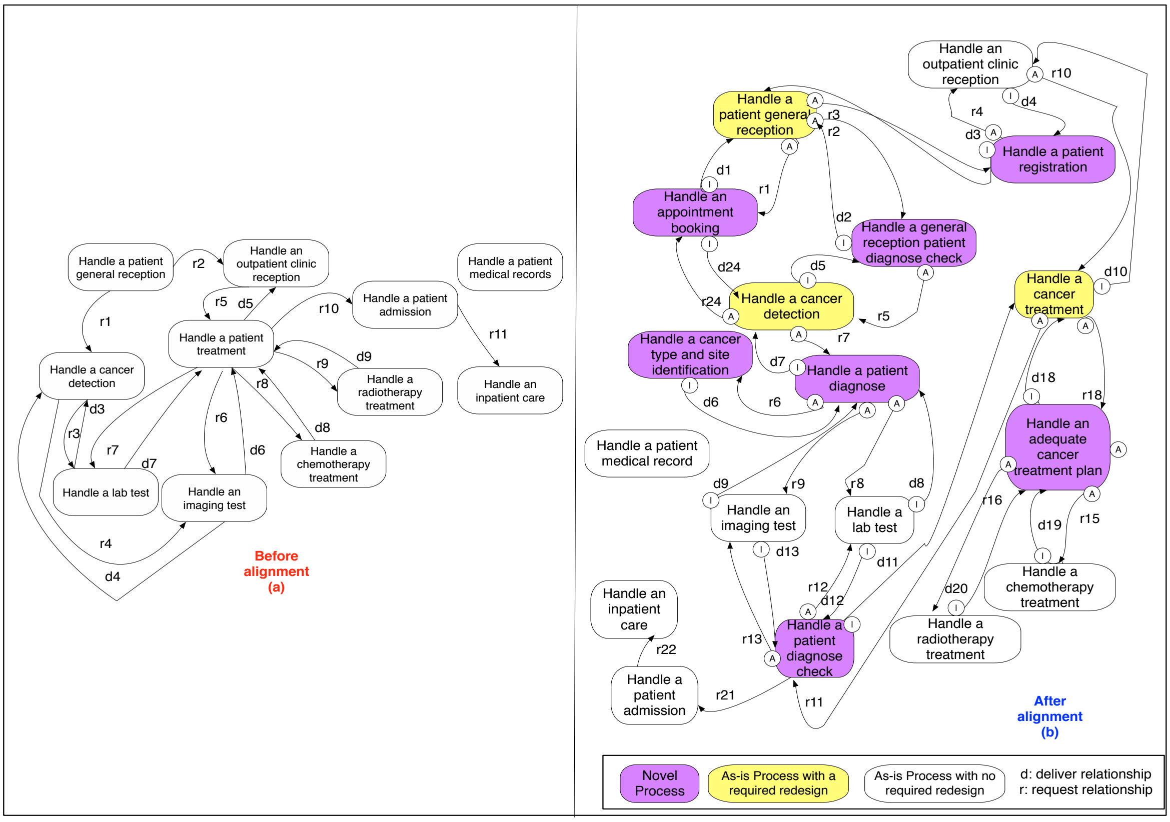


Figure 7.4: Before and After the Alignment of Riva BPA with a BSV Using the CCR Representative Processes.

novel processes, as-is process with required redesign and as-is processes with no required redesign.

Also, Table 7.2 highlights the effectiveness of the NFR framework linked into the *i** framework through identifying new concepts (i.e., quality concepts) for the Riva BPA in order to reduce the gap with the BSV from the quality point of view. For example, the EBQ concept was introduced and has been elaborated into a new concept (Q-UoWs) that turns at the end as quality of process (QoP) in the 2nd cut architecture. Finally, the integrated NFR frameworks into an *i** framework are identified with a reference number that is used in to represent elaborative Q-UoWs and QoP as a detailed quality model representation. This has further enriched the Riva-BPA and its associated BPMs. In short words, it was observed that the *i** framework assisted only in identifying the main quality requirements in BPA and associated BPMs, where their detailed representation is addressed through the linked NFR framework

7.3.4 Informing the Extent of the Correctness, Completeness and Consistency of the GQ-BPAOnt Ontology Using the Representative Case Study

This section aims at showing the evaluation of the GQ-srBPA ontology and the GQ-srBPA-sBPMN merger ontology. This evaluation stems from the evaluation of work in the original BPAOnt in (Yousef, 2010). However, it is carried out here with respect the integration of goals and quality requirements. The work of this section is similar to the one in Section 7.3.2, but with respect to the second layer of the GQ-BPAOntoSOA framework. Figure 7.5 shows

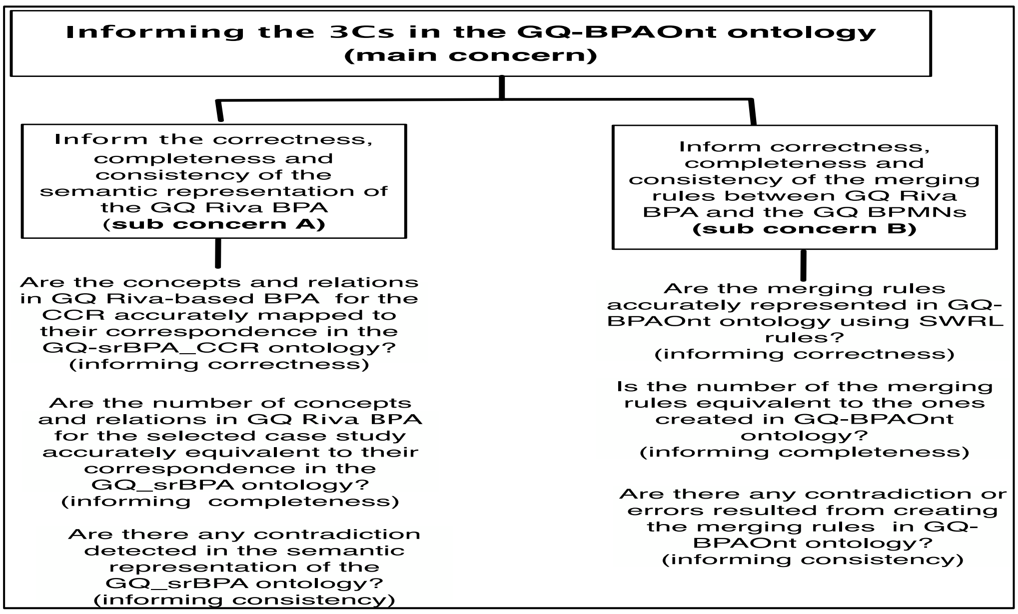


Figure 7.5: The Concern-based Diagram for addressing the 3Cs of the GQ-BPAOnt Ontology

the concern-based diagram for informing the 3Cs for the GQ-BPAOnt. The 3Cs values are either informed by Yes or No. Appendix U presents the 3Cs values for the GQ-srBPA_CCR ontology and the GQ_srBPA-sBPMN_CCR merger ontology.

Using the appendix U, a reader can notice the points below.

1- Walking through the GQ_srBPA_CCR manifests a good accurate mapping of concepts and relationships with the ones in the CCR GQ-Riva BPA shown in Appendix N. Also, the number of instances of concepts and relationships in the GQ_srBPA_CCR are equivalent to the ones in CCR GQ-Riva BPA to some good extent. Finally, no errors were generated after running the built-in reasoned (i.e., Pellet 1.5.2). Accordingly, this shows addressing the 3Cs to a good extent for the evaluation of the GQ-srBPA_CCR ontology.

2- Walking through the merging rules implemented in the GQ_srBPA-sBPMN merger ontology shows an accurate mapping, where the three merging rules are presented in object properties in this component. Also, the reasoner did not show any inconsistency error during its execution. Accordingly, it can be observed using Appendix U that the 3Cs are addressed to a good extent in this evaluation.

7.3.5 Informing the Effectiveness of the SI Approach Employed in the GQ-BPAOntoSOA Framework Using the Representative Case Study

This section aims at informing the effectiveness of the GQ-SI approach as part of the evaluation framework. The *effectiveness* of the GQ-SI approach means the capability of this approach in producing GQ candidate software services, their associated capabilities and QoS requirements.

The effectiveness of GQ-SI is informed after identifying criteria that address interesting aspects for this research and particularly regarding goals and quality requirements. Also, the effectiveness is informed after a comparison is carried out between the generated clusters in the BPAOntoSOA framework and the ones generated using the new framework as shown in the Tables 7.4 and 7.5, respectively.

The SI approach have addressed most of the SOA principles unless that the services must share a formal contract and the services must be discoverable principles, as they are concerned with service design, as shown in Section 6.3.2.2. The GQ-RPA-based SI approach stems from the SI-approach implemented in the BPAOntoSOA framework, which facilitates assessing the GQ-SI approach conformance to the SOA principles (Erl, 2007). The current

SOA principles lack addressing the GoS and the QoS, where these two complement the current principles in (Erl, 2007). Moreover, the entity service definition does not address the GoS and the QoS. Therefore, the entity service definition has been refined and introduced in Section 6.3.2.2. Informing the effectiveness of the GQ-SI approach highlights further effectiveness of the GQ-BPA model and associated GQ-BPMNs in carrying out the GQ-SI approach.

In this part of evaluation, the current SOA principles are revisited and inspected for their adaptability to integrate goals and quality requirements for each principle without violating their fundamental context. Moreover, the driven capabilities that stem from goals and quality requirements are assessed.

The following criteria are defined to investigate the effectiveness of ***the SI approach used in the original framework and the GQ-SI approach used in the extended framework:***

1- Number of clusters generated using the original and the extended frameworks. (Quantitative criterion value).
2- Determine whether the pre-existing clusters (i.e., generated using the original framework) and the ones derived using the extended framework are goal-based. (Yes/No criterion value).
3- Determine whether the pre-existing clusters (i.e., generated using the original framework) and the ones derived using the extended framework are quality-linked. (Yes/No criterion value).
4- Determine cluster type (Standalone/Related criterion value).
5- Number of capabilities (Quantitative criterion value).
6- Investigate whether the pre-existing clusters and the ones derived using the extended framework are addressing SOA principle (Yes/No criterion value).
7- Capabilities classification (Yes/No criterion value).

All the criteria above are determined as shown in Tables 7.4 and 7.5 for the clusters identified using the original framework and the ones identified using the extended framework, respectively. However, it is necessary to investigate the implication of integrating goals and quality requirements into the SI approach (qualitative criterion value), as shown in Table 7.5. In addition, Figure 7.6 shows the identified clusters using the original BPAOntoSOA framework and the GQ-BPAOntoSOA framework for the CCR representative processes. The entries in Table 7.5 are colored with the reference to the ones in Figure 7.6 (i.e., pink refers to new clusters, yellow refers to pre-existing clusters with some required refinements because of the integration of goals and quality requirements and the white color means that the original clusters remain as they are with some light refinements).

Table 7.4: Criteria for Clusters Generated Using the Original BPAOntoSOA Framework for the CCR Selected Representative Processes

Before: In the original BPAOntoSOA framework (Candidate software services point of view)						
Cluster name	Was it goal-based?	Was it quality-linked?	Cluster type: Standalone or Related	Does it address SOA principles?	Number of capabilities?	Capabilities classification?
C1: General Reception	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	8	No
C2: Outpatient Clinic Reception	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	2	No
C3: Cancer Detection	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	8	No
C4: Lab Test	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	3	No
C5: Imaging Test	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	6	No
C6: Patient Treatment	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	9	No

Table 7.4 (Cont'd): Criteria for Clusters Generated Using the Original BPAOntoSOA Framework for the CCR Selected Representative Processes

Before: In the original BPAOntoSOA framework (Candidate software services point of view)						
Cluster name	Was it goal-based?	Was it quality-linked?	Cluster type: Standalone or Related	Does it address SOA principles?	Number of capabilities?	Capabilities classification?
C7: Chemotherapy Treatment	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	4	No
C8: Radiotherapy Treatment	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	4	No
C9: Patient Medical Records	No	No	Standalone	Yes, unless sharing formal service contract and being discoverable.	5	No
C10: Patient Admission and Care	No	No	Related	Yes, unless sharing formal service contract and being discoverable.	6	No

Table 7.5: Criteria for the Clusters Generated Using the GQ-BPAOntoSOA Framework for the CCR Selected Rpresentative Processes

After: In the new GQ-BPAOntoSOA framework (Candidate software services point of view)							
Cluster name	Is it goal-based?	Is it quality-linked?	Does it address SOA principles?	Cluster type?	Number of capabilities?	Capabilities classification?	Any new concepts emerged in SI approach?
C1: General Reception	Yes	Yes	Yes, unless sharing formal service contract and being	Standalone	5	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster

			discoverable.				classified.
C2: Appointment Booking	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	1	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C3: Patient Registration	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	7	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C4: General Reception Patient Diagnose Check	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	2	Yes	Yes. The QoS concept, the goal-driven capabilities. Also, the cluster classified. and quality-driven capabilities
C5: Outpatient Reception	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	2	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C6: Cancer detection	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	3	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C7: Identification of cancer type and site	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	3	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C8: Patient diagnose	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	2	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C9: Lab test	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	3	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.

C10: Imaging test	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	6	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C11: Cancer Treatment	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	1	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C12: Patient Diagnose Check	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	3	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C13: Adequate Treatment Plan	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	5	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified.
C14: Radiotherapy Treatment	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	4	No	Yes. The QoS concept and the goal-driven capabilities Also, the cluster classified.
C15: Chemotherapy Treatment	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	3	No	Yes. The QoS concept and the goal-driven capabilities Also, the cluster classified
C16: Patient Medical Record	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Standalone	5	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified
C17: Patient Admission and Care	Yes	Yes	Yes, unless sharing formal service contract and being discoverable.	Related	6	Yes	Yes. The QoS concept, the goal-driven capabilities and quality-driven capabilities. Also, the cluster classified

Using the tables 7.4 and 7.5, a reader can derive the following observations.

1- The number of clusters has been increased from **10 to 17** and this is because of the newly identified clusters (e.g., C2: Appointment Booking) using the GQ-SI approach. This is mainly justified in the integration of goals into the GQ-BPAOnt ontology.

2- The new clusters involve the identification of QoS requirements. A reader can notice the absence of the QoS requirements in the 10 old clusters (i.e., identified using the original BPAOntoSOA framework) as shown in Table 7.4. In this regard, a reader should notice that the GQ-SI approach remained the role of the original SI approach regarding the manner followed in identifying clusters, where the GQ-SI approach has constrained the clusters with the QoS requirements using the GQ-BPAOnt. This manifests one of the major refinements integrated into the original SI approach.

3- According to the evaluation work in (Yousef, 2010), the 10 clusters in Table 7.4 addressed the SOA principles in (Erl, 2007), where service should be: loosely coupled, abstracting underlying logic, reusable, composable and stateless. These principles are revisited with regard to the identified clusters in Table 7.5 and the ones in the right hand side in Figure 7.6. Therefore, it is required to assess addressing these principles with respect to the GQ-RPA clusters in Table 7.5 and Figure 7.6.

   i) The loosely coupled principle: this principle is addressed through reducing the dependency relationships between software services (i.e., GQ RPA clusters). It has been discussed in Section 6.3.2.2 that the standalone clusters address this principle, as they do not participate in any relationships in the GQ 2nd cut architecture. Figure S.1 in Appendix S shows this with 16 standalone GQ-CPs after deleting the conditional request and deliver relationships. Accordingly, this has derived 16 standalone clusters as shown in Table 7.5. Only one connected cluster is identified (i.e., C17) with two related GQ-CPs. This related cluster addresses this principle, as it is not related to another cluster through relationships in the GQ-Riva BPA.

   ii) Abstracting underlying logic principle: this principle presented with respect to the GQ-RPA cluster in Section 6.3.2.2. It considered the GQ-RPA cluster as a black box that abstracts the underlying logic through the goal of the cluster, where the capabilities are embodied in the processes that are embodied in the goal of the cluster. All GQ-RPA clusters in Table 7.5 act as black boxes described by the goal of the cluster.

iii) Reusability: first, this principle has an inverse relationship with the granularity level of the service (Erl, 2007). This principle was addressed in the evaluation work of Yousef (Yousef, 2010) through encapsulating large functionality in order to arrive to what is called coarse-grain service. Second, in this evaluation, it is recommended to address reusability through using the purpose of the service (i.e., goal of cluster) as discussed in Section 6.3.2.2. Figure 7.7 shows the hierarchy presented in Chapter 6 with respect to the selected CCR processes in this evaluation. In this Figure, the level of the GQ-RPA clusters remains middle between the most coarse-grained element (i.e., GQ 2nd cut architecture) and the finest elements (i.e., tasks). Therefore, it could be claimed that the GQ-RPA clusters have addressed an appropriate granularity level. In addition, the GQ-RPA clusters are all identified with a purpose (i.e., goal of cluster). Accordingly, this assists in addressing high reusability. The 17 clusters in Table 7.5 are reusable.

iv) Composability: is another face of reusability, where service aggregate in order to produce an application or solution that is reusable too. In this evaluation, this principle is addressed in the same manner in (Yousef, 2010). That is, composability is carried out when the need to an appropriate level of granularity in order to "maximise composition opportunities". Services represent business processes that are related through connection points that lead to the end to the BPA model. Therefore, services can be connected and composable too.

v) Stateless: this principle is based on minimising the amount of managed information taking into the account remaining the service stateful, yet only when it is required (e.g., request or response). This principle is addressed in the same manner in (Yousef, 2010), as it is essentially based on the heuristic of deleting conditional requires and deliver relationships in the 2nd cut architecture. "Accordingly, we do not add a CP or a CMP to a cluster if it can only be requested according to some condition" (Yousef, 2010). For example, C11 and C12 are limited in connection, as they do not participate in any request/deliver relationships although the roles in the processes are connected.

4- The GQ-SI approach has generated a new classification of capabilities, where capabilities are classified as goal-driven or quality-driven. This is clearly shown in the clusters identified for the CCR representative processes as attached in Appendix S (in Table S.1).

5- In Table 7.5, the number of capabilities has been decreased in the yellow clusters (i.e., as-is clusters with a required refinements). This is because of the integration of goals that has assisted in identifying their relevant capabilities. This assists in paving the way addressing higher correct, complete and consistent software services with respect to goals.

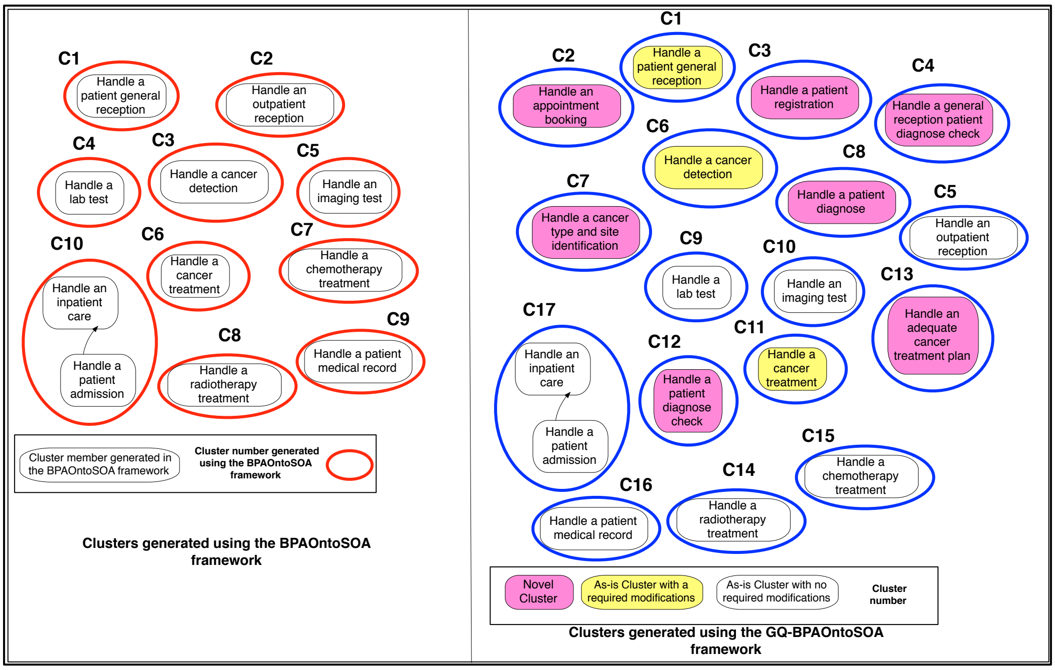


Figure 7.6: The Identified Clusters Using the Original and the Extended Frameworks For the Selected Representative CCR Processes

6- In Table 7.5, the colored entries let a reader notice the new classification of clusters that stems from the GQ-BPAOnt in the GQ-BPAOntoSOA framework.

The 17 clusters in Table 7.5 are classified into three: 7 new clusters in pink (C2, C3, C4, C7, C8, C12 and C13), 3 as-is clusters with required refinements in yellow (C1, C6, and C11) and 7 as-is cluster with no considerable refinements in white (C5, C9, C10, C14, C15, C16 and C17). This classification is generated as an implication of the integration of goals and quality requirements into the original BPAOntoSOA framework. A reader can notice the difference in the division of clusters in Figure 7.6. The 17 clusters in Table 7.5 and Figure 7.6 attained an effective or desirable division that is based on the integration of goals. This assists in distinguishing a cluster from another using goals as ID as presented in Section 6.3.1. The effective division of the 17 clusters is also determined by the uniqueness of clusters (i.e., derived from the uniqueness of the BPMs as shown in Table 7.3). All BPMs in Table 7.3 are unique and accordingly the identified clusters from theses BPMs are unique as well.

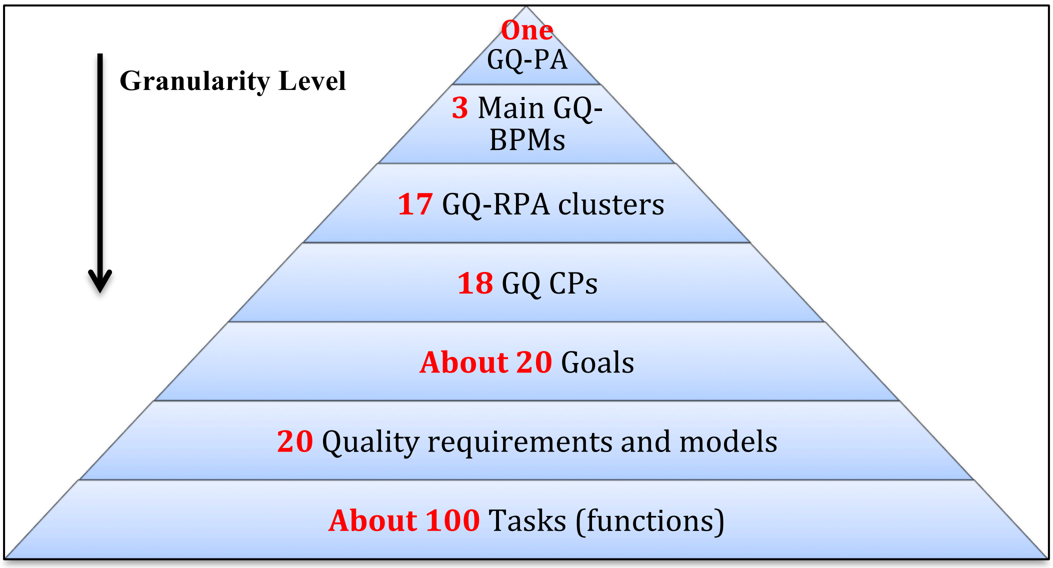


Figure 7.7: GQ-RPA Clusters Position in the GQ-PA Granularity Hierarchy

7- New concepts have been identified such as goal of cluster, quality of cluster, goal-driven capabilities and quality-driven capabilities. This has enriched the description or specification of clusters.

8- The two types of clusters were detected in the pre-existing clusters. These two types still appear in the identified clusters using the GQ-SI approach as shown in Figure 7.6. This means that the GQ-SI approach is still capable to derive two types of clusters (e.g., one related cluster and 16 standalone clusters) taking into the account the integration of goals and quality requirements as shown in Table 7.5 and Figure 7.6.

In the light of the above, the GQ-SI appeared better to a good extent and certainly it is more effective than the original SI approach, as the latter missed the identification of goals and quality requirements although it addressed the SOA principles (Yousef, 2010). By and large, the instantiation of the GQ SI layer led to a better specification or description of services through explicitly specifying goals and quality requirements. A consumer is now capable to see the service as goal-oriented rather than function-oriented. Also, a consumer is anticipated to be satisfied after specifying the QoS requirements that are not less in importance than the capabilities, as the consumer can reject or accept a service because of an absence or presence of a QoS requirement. This has not manifested the effectiveness of the GQ-SI approach only, yet it also paved the way for informing the effectiveness of the GQ-BPAOntoSOA framework in relation to the 3Cs using the same case study. This is because the resultant SOA-able candidate software services from the third layer represent the output from the GQ-BPAOntoSOA framework. The original framework did not consider addressing the 3Cs for the identified software services. With regard to the GQ-BPAOntoSOA framework, the consistency, completeness and correctness are informed using criteria below respectively.

1- Software services consistency. This criterion means that no two software services share same business goals and quality requirements. In other words, it is claimed that a software service is consistent when it is coherent and supports backward traceability to business goals.

2- Software services completeness. This criterion is determined after conducting a comparison between the number of the software service resultant from the original framework and the one resultant from the extended framework. This involves identifying the relationship between the two numbers. Software services address higher completeness when the original framework-driven software services are subset of the ones derived using the extended framework.

3- Software service correction. This criterion means that software services derived using the GQ-BPAOntoSOA framework can be accurately mapped to their elements in BPMs, BPA and BSV. This means a software service's goals, quality requirements and capabilities have a backward traceability to BPMs, BPA and BSV.

It is necessary to highlight that a consistent software service is not necessarily correct and vice versa. Therefore, each of the 3Cs should be considered carefully. The 3Cs are determined using the information in Appendix S and Tables 7.4 and 7.5, where a reader can derive the following observations.

1- All software services identified using the GQ-BPAOntoSOA framework are consistent. This is because no two software services were found sharing same business goals and QoS requirements.

2- The 10 software services identified using the original framework are subset of the 17 ones identified using the GQ-BPAOntoSOA framework. Therefore, this criterion is addressed with no missing software services. This is because the 10 software services in Table 7.4 remain the same in the 17 software services identified using the GQ-BPAOntoSOA framework in Table 7.5.

3- Each software service resultant from the GQ-BPAOntoSOA framework is mapped to its drivers BPMs, BPA elements and BSV elements. This also involves accurate mapping of QoS and capabilities.

It is apparent now that the extended framework is more effective than the original framework using the work of this section and Appendix S in terms of addressing higher 3Cs with regard to business goals and quality requirements.

7.3.6 Answering the research questions using the concern-based approach

This section presents the research roadmap that is derived using the concern-based approach as shown in Figure 7.8. Using the concern-based approach (Kotonya and Sommerville, 1998) in Section 7.2, the four research questions, which have been presented in Section 1.4, are turned into four corresponding main concerns. In Figure 7.8, each concern derives its associated requirements and questions. Addressing concerns is carried out in the bottom-up manner (i.e., answering the derived questions under each concern). In this section, the research questions are answered using the bottom-up roadmap in Figure 7.9 along with a discussion for each research question (i.e., main concern) as below.

The first research question ***(RQ1)*** was answered after carrying out a deep analysis of each layer of the original BPAOntoSOA framework. Having each layer analysed, the main shortcomings were identified regarding goals and quality requirements. In order to arrive at appropriate answering for ***RQ1***, the question has been divided into two sub-questions, as shown in Figure 7.8. In short answering words, the absence of goals and quality requirements in the BPAOnt ontology instantiation layer stems from their original source of absence in the Riva BPA method and the BPMN modelling language, as shown in Chapter 5. Accordingly, this shortcoming has propagated into their semantic representation and influenced the function of each component in the original BPAOnt instantiation layer. Answering the first sub-question of RQ1 is addressed through refining the approaches designed the inputs of the BPAOnt instantiation layer (i.e., the Riva BPA and associated BPMNs) and accordingly the function of each of the three components that design this layer. This has been clearly addressed and shown in the work of Chapter 5 (Sections 5.5, 5.6 and 5.7). The second sub-question for ***RQ1*** has been answered in the work of Chapter 6 (Sections 6.3, 6.4 and 6.6) that has shown the reusing of the entire original software SI layer in order to refine the function of its two components with the respect to the integration of goals and quality requirements. This could be answered in short words that the original RPA-based SI approach did not consider the identification of goals and quality requirements for the derived candidate services. This is essentially justified by the absence of goals and quality requirements in the input for this layer (i.e., the BPAOnt instantiation for an organisation) and accordingly, the function of the RPA-based SI approach and the service capabilities identification should be refined with respect to the integration of goals and quality requirements. Answering RQ1 is shown in Figure 7.9.

The second research question (***RQ2)*** has been answered in Section 5.8 through presenting the alignment approach. Aligning a Riva BPA with a BSV is essentially based on detecting as-is EBEs and identifying new ones. The alignment approach has involved the identification of quality requirements as part of understanding an organisation. Also this research question is answered in Section 7.3.3 through identifying criteria that assist informing the effectiveness of alignment between the Riva BPAs and its associated BPMs & BSV. The related work of

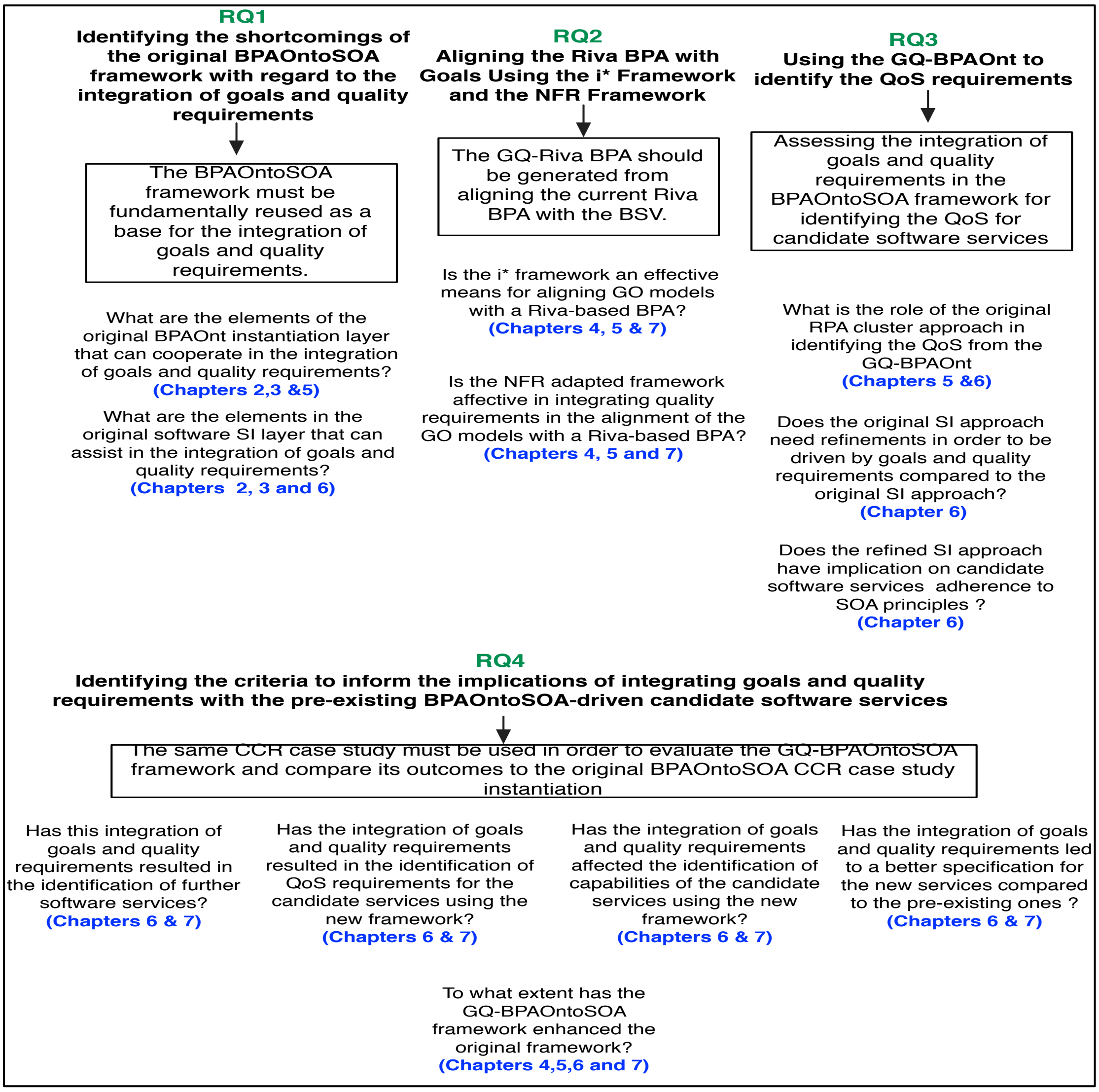


Figure 7.8: The Research Roadmap using the Original Concerned-based Approach

RQ2 was shown in Chapters 4 and 5. This research question is divided into sub-question as was shown in Figure 7.8. Recalling Table 7.2, the new 77 identified EBEs stem from the *i** framework models that 8 of them turn into new UoWs and then into new CPs in the ultimate Riva BPA model. This informs that the use of the *i** framework is effective as it has generated new knowledge in the Riva BPA. One limitation appears in the manual extraction (i.e., not automated) of EBEs from the corresponding *i** framework although it is better than deriving them using the original brainstorming approach. This manual extraction from the *i** framework requires less time and effort than the brainstorming. Also, the *i** framework assisted in identifying the main quality requirements only whereas their detailed representation is addressed using the relevant NFR framework. Table 7.3 gives a further answering to the first sub-question of RQ2 in Figure 7.8. In particular, the effectiveness of the *i** framework propagated to the BPMs and made them goal-based. The second sub-question of RQ2 concerns with the effectiveness of the NFR framework. This was also answered in Section 7.3.3. Table 7.2 has shown the effectiveness of the NFR framework with the help of the *i** framework in identifying new concepts in the Riva method for the quality requirements. Table 7.3 shows how the extent of this effectiveness of the NFR framework in the BSV propagated into the BPMs via the Riva BPA representing the quality of process and associated quality models (if needed). This has enhanced the representation of the Riva BPA and its associated BPMs. Answering ***RQ2*** is shown in Figure 7.9.

The ***RQ3*** is addressed after answering its their associated three sub-questions. The first sub-question was answered in the work of Chapter 6 as the original RPA clusters suffer from the absence of the QoS requirements, as presented in Section 6.3. Also, this sub-question has been answered within Section 7.3.5 after applying the GQ-SI within the GQ-BPAOnt of the selected CCR processes. In Section 7.3.5, it was clear that the GQ-SI approach employed the original SI approach for the identification of the GQ-RPA clusters that are derived in the same original manner. The identification of clusters paves the way constraining them by the desired QoS requirements and this was the role of the original SI approach. This appeared as the answer of the first sub-question. However, the GQ-SI approach has extracted the goals of the processes in the GQ Riva BPA model in order to consider them as goals of the clusters. This allows sharing this answer not only for the first sub-question, but also for the second sub-question that the RPA-based approach requires refinements as have been shown in the Section 6.3. In this chapter, the refined GQ-SI approach has been applied using the CCR representative processes, where Tables 7.4 and 7.5 show the implication of using the refined SI approach. Few implications have been noted in Section 7.3.5. Regarding the third sub-question, addressing the SOA principles using the services (i.e., GQ-RPA clusters) that were generated using the GQ-SI approach was carried out in Section 6.3.2.2. This is also was evaluated using the case study in Section 7.3.5, and it was apparent that the enhanced clusters have addressed the SOA principles: loosely coupled, reusability, abstracting underlying logic, stateless and composability, where the discoverability, sharing formal contract and autonomous of service principles were out of the scope of this research. Figure 7.9 shows how this question was answered using the related work of thesis chapters.

In Figure 7.9, the final research question ***RQ4*** is divided into five sub-questions and they are answered as below.

1- Has the integration of goals and quality requirements resulted in the identification of further software services?
Yes. This was shown in Section 7.3.5 using the selected representative processes for the CCR case study. The number of software services has increased after integrating goals and quality requirements into the original BPAOntoSOA framework from **10 to 17**. The increased number of services is justified by the integration of goals and quality requirements into the original Riva BPA and associated BMPs via an alignment with BSV.

2- Has the integration of goals and quality requirements resulted in the identification of QoS requirements for the candidate services using the new framework?
Yes. In Section 7.3.5, tables 7.4 and 7.5 show how that BPAOntoSOA framework driven software services were not quality-linked and how now they are integrated with the QoS requirements. Appendix S presents detailed specification of the quality-linked services. The identification of QoS requirements stem from the QoP in the GQ-CPs presented in the 2nd cut architecture, where the origin of the QoP stems from the BSV.

3- Has the integration of goals and quality requirements affected the identification of capabilities of the candidate services using the new framework?
Yes. And this was clearly manifested from two views, as shown in Section 7.3.5. First, the integration of goals has affected the identification of capabilities for as-is clusters with required refinements (i.e., yellow clusters in Figure 7.6). It was noticed that number of capbilites have changed (e.g., C6: cancer detection cluster capabilities have been reduced from 9 to 3). Second, the integration of goals and quality requirements has classified the service capabilities into goal-driven and quality-driven. The goal-driven capabilities represent the service capabilities in order to

inform the number of changed capabilities. Also, the quality-driven capabilities are concerned with addressing the QoS requirements only.

4- Has the integration of goals and quality requirements led to a better specification for the new services compared to the pre-existing ones?
Yes. This was shown at the end of Section 7.3.5 after informing the effectiveness of the GQ-SI approach and accordingly informing the effectiveness of the GQ-BPAOntoSOA framework regarding the 3Cs. The integration of goals and quality requirements has derived goal-based candidate software services; their associated capabilities and QoS requirements, where the original SI approach derived the candidate software services and their associated capabilities only. In addition, the integration of goals and quality requirements assisted in deriving candidate software services that are higher in consistency, completeness and correctness than ones derived using the original framework, as discussed at the end of Section 7.3.5. The reader can notice how the specification of services has been enriched using the tables 7.4 and 7.5 and Appendix S.

5- To what extent has the GQ-BPAOntoSOA framework enhanced the original framework?
The answer of this question involves qualitative information about the effectiveness of the GQ-BPAOntoSOA framework. Three means are required to tell about this effectiveness. First, it is required referring to the effectiveness of the employed alignment approach. Second, it is then required referring to the implication of employing the alignment approach on the SI. Finally, the effectiveness of the GQ-BPAOntoSOA framework involves informing about the 3Cs that were absent in the evaluation of the original framework.

With regard to the effectiveness of the employed alignment approach, it is possible to derive evaluation outcomes using the work in Section 7.3.3. The alignment approach in the GQ-BPAOntoSOA framework has enriched the Riva-based BPA representation by making it goal-based and quality-linked model. It also has derived a classification for the associated BPMs that is alignment-driven and cost-based classification as shown in Section 7.3.3. In addition, the alignment approach has bridged the gap between GO models and Riva-based BPAs. One of the noticed observations in this bridging is the newly added knowledge into the BPA that encompasses new EBEs, UoWs, CPs and/or CMPs. This added knowledge is needed, as it emerged because of the integration of goals and quality requirements into the original framework.

Using the carried out evaluation in Section 7.3.5, it is possible telling about effectiveness of the GQ-SI approach. The refined SI approach in the GQ-BPAOntoSOA framework has enriched the software services' specification by involving their goals and QoS requirements. This did not neglect the original approach of identifying the software services and their associated capabilities. The refined SI approach went further than this by classifying the capabilities into goal-driven and quality-driven as shown in Appendix S. In addition, the candidate software services are classified using the outcomes of the conducted alignment.

Finally, it is necessary informing the 3Cs as part of reflecting the effectiveness of the GQ-BPAOntoSOA framework. The 3Cs were presented in Section 1.2.2 as motivations in this research and considered them within the evaluation of the effectiveness of the GQ-SI approach in Section 7.3.5. In short words, the GQ-BPAOntoSOA framework-driven SOA-able candidate software services resulted a higher-level of consistency, completeness and correctness than the original framework due to the absence of business goals and quality requirements in the original framework.

The GQ-BPAOntoSOA framework has enhanced the original framework to the extent of deriving GQ candidate software services that address the mentioned SOA principles and 3Cs as shown in Section 7.3.5. Also, it has enhanced the original framework to the extent of deriving new services, new capabilities and QoS requirements. The GQ-BPAOntoSOA framework has derived classification for the BPMs resultant from an alignment with a BSV, as shown in Section 7.3.3. Similarly, this classification has propagated into the derived services as shown in Section 7.3.5. Accordingly, this has enhanced the specification of the SOA-able software services. Also, the extended framework is now capable to generate the sematic representation of the BSV in order to understand the business organisation from the point of view of the goals. All of this was absent in the original BPAOntoSOA framework.

Although it is possible to conclude that the GQ-BPAOntoSOA framework is more effective than the BPAOntoSOA framework, still it is considered simpler than other approaches. This is because of the below points.

1) The GQ-BPAOntoSOA framework is based on simple GO approaches, BPA method and SI approach.
2) The GQ-BPAOntoSOA framework is automated to some extent, where the automation increases while moving to the bottom of the framework.
3) It employs simple alignment approach that is essentially based on reusing as-is BPA model and its associated BPMs and aligning them with a BSV in order to obtain the desired to-be BPA and its associated BPMs. Alignment is another word of reusing that is more useful than deriving BPA models from a BSV. This is because alignment can be described as an effort saver with the comparison to other approaches.
4) The design of the GQ-BPAOntoSOA framework is based on the separation of concerns that facilitates understanding and addressing each concern.
5) Domain independent.
6) Since it is goal-based framework, it is easier to identify relevant software services and their capabilities with respect to goals (i.e., targets) rather than functionalities.

One may wonder about the need of informing the 3Cs for the GQOnt and the GQ-BPAOnt ontologies. This part of work was carried out in Sections 7.3.2 and 7.3.4 in order to address the semantic representation before initiating the work of the next layer in the GQ-BPAOntoSOA framework. For example, it is not appropriate to start the semantic identification of services using the GQ-BPAOnt without evaluating its semantic representation, as any error occurred in the GQ-BPAOnt ontology is highly anticipated to propagate into the services if it wasn't corrected.

After answering the four research questions, it is now required revisiting the research hypothesis in Section 1.4 and rewriting it with respect to the answered four research questions. It is concluded now that, "***Using the BPAOntoSOA framework, it is possible to semantically derive goal-based and quality-linked SOA services from the integration of business goals into the Riva-based business process architectures***".

Finally, the work of this chapter has addressed answering the research question in Figure 7.10 and activating the research process phases as shown in Figure 7.11.

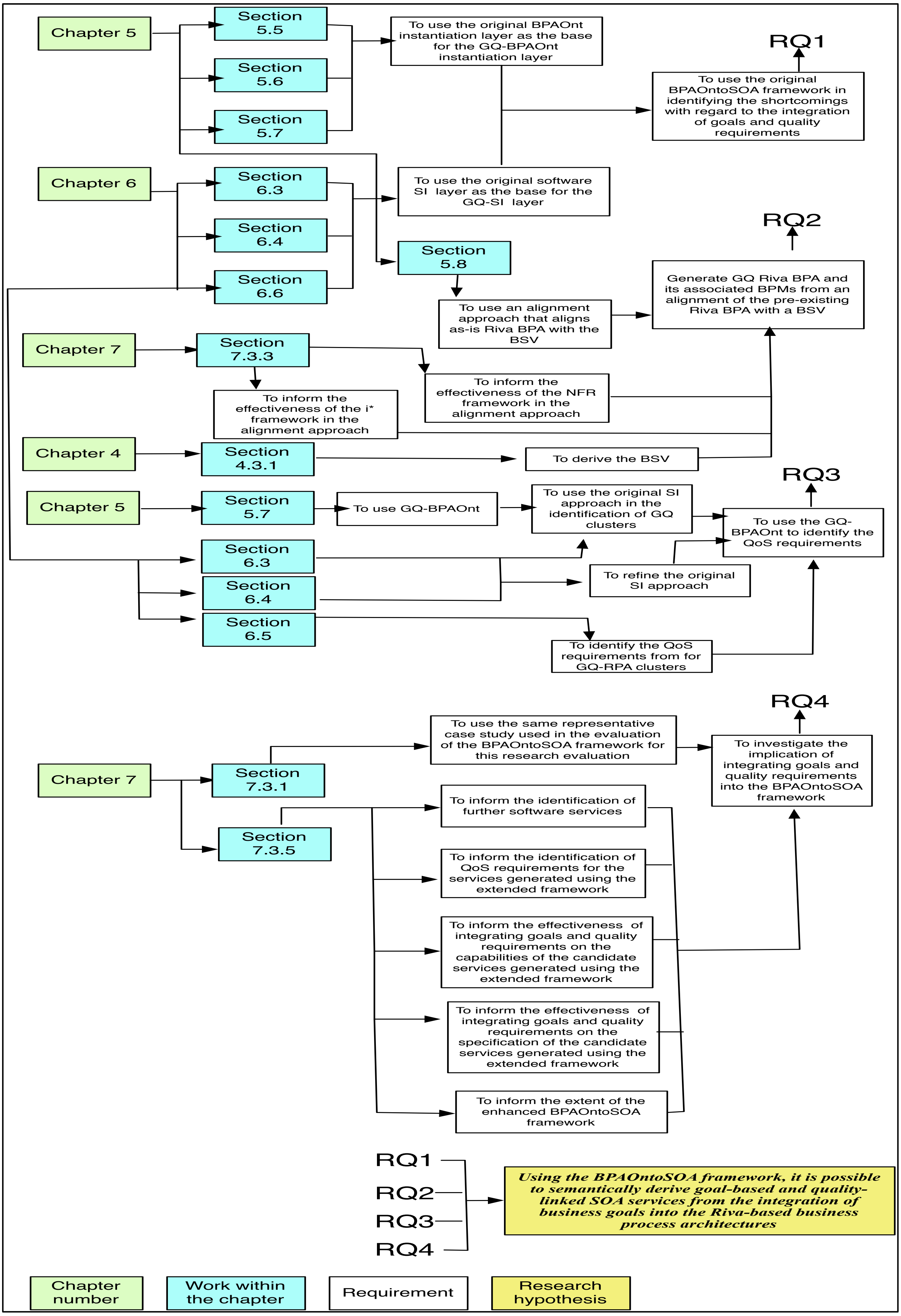


Figure 7.9: The Bottom-Up Answering of the Research Question in order to Prove or Disapprove the Research Hypothesis

Research Questions	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RQ1									
RQ2									
RQ3									
RQ4							✓		

Figure 7.10: Answering the Research Question Using the Work of Chapter 7

Research Phases	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Status
RP1									**Activated**
RP2									**Activated**
RP3									**Activated**
RP4(ai)									**Activated**
RP4(aii)									**Activated**
RP4(bi)									**Activated**
RP4(bii)									**Activated**
RP4(biii)									**Activated**
RP5							✓		**Active**
RP6							✓		**Active**

Figure 7.11: Activating the Research Process Phases Using the Work of Chapter 7

Chapter Eight: Conclusion and Future Work

This research is aimed at investigating the implication of integrating business goals and quality requirements into business process architectures. In particular, the BPAOntoSOA framework was the base for such investigation. The research led to extend the BPAOntoSOA framework through engaging goals and quality requirements in deriving the process of software service identification, namely through the extended GQ-BPAOntoSOA framework. The main purpose and function of the GQ-BPAOntoSOA framework is deriving candidate software services, their associated capabilities and QoS requirements. This functionality has highly dependent on reusing the function of the BPAOntoSOA framework.

Three fundamental stages were carried out in order to achieve the GQ-BPAOntoSOA framework. First, the GO models ontology and the soft goal models ontology were developed and linked in order to achieve the semantic representation of the BSV (i.e., GQOnt ontology) of an organisation, as presented in Chapter 4. The instantiation of the BSV is the driver of the GQ-BPAOntoSOA framework. Second, the pre-existing Riva BPA (i.e., neither goal-based nor quality-linked model) and its associated BPMs have been semantically aligned to the earlier BSV using a novel alignment approach. This semantic alignment entailed refinements within the original srBPA ontology and associated sBPMN ontology of the BPAOntoSOA framework. Accordingly, the alignment has been carried out using the GQOnt ontology as a base, presented in Chapter 5. In particular, this included the semantic representation of the GQ Riva BPA and associated GQ-BPMs in the form of GQ-BPAOnt ontology that forms the main input to the next final stage. For the software service identification, the original SI approach was reused and extended in order to adapt to the context of GQ-BPAOnt ontology. This stage of work was presented in Chapter 6. This has resulted in the identification and brief specification of goal-based, quality-linked and BPA-driven software services. In another words, this has contributed to bridging the gap between the business world and the systems world further ahead than the original BPAOntoSOA framework.

This research has employed the DSR methodology in order to solve the identified research problem through constructing the GQ-BPAOntoSOA framework using the aforementioned three fundamental stages. The DSR methodology steps fit with the notion and direction of the GQ-BPAOntoSOA framework development as artifact. The design was based on reusing the BPAOntoSOA framework and relevant theories such as the $i*$ framework, NFR framework, Riva BPA, BPMN and BPA-driven SI. This has led to derive new methods such as aligning a

Riva BPA to a BSV, proposing GQ-Riva BPA and GQ-BPMNs and proposing refined GQ-SI approach. Finally, the DSR methodology is elaborative as shown in Sections 3.2 and 3.4.

This chapter is structured as follows. Section 8.1 presents the main research outcomes and results. Section 8.2 shows a summary of a comparison between the original and the new framework. Section 8.3 shows how the work of thesis chapters relate with the research questions and research process phases. And finally, Section 8.4 proposes further research directions for this research.

8.1 Main Research Outcomes

- **The GQ-BPAOntoSOA Framework**

  This framework is one of the main outcomes of this research. It is apparent that this framework is originally based on the BPAOntoSOA framework (Yousef, 2010). The input to the this framework is the BSV for an organisation that is comprised of the set of GO models and their associated quality-oriented models designed using the NFR framework in this research. The output of this framework is the goal-based and quality-linked SOA's candidate software services with their associated capabilities and QoS requirements.

  The GQ-BPAOntoSOA framework is a three-layered framework. The first layer is concerned with the instantiation of the GQOnt ontology for a given organisation using its BPMs. The second layer is reused from the BPAOntoSOA framework (Yousef, 2010). However, this layer has been enriched in order to achieve the desired alignment with the GQOnt ontology instantiation. In particular, the srBPA ontology, which conceptualises the original Riva BPA, has been extended into GQ-srBPA ontology in order to bridge the link with the GQOnt ontology. Similarly, the sBPMN ontology has been extended to the GQ-sBPMN ontology to address the integration of goals and quality requirements in the GQ-srBPA and the GQOnt ontologies. Accordingly, the second layer in the GQ-BPAOntoSOA framework has resulted in the GQ-BPAOnt instantiation for a particular organisation. The third layer is also reused from the BPAOntoSOA framework, but it is enhanced in order to be goal-based and quality-linked. The main input to the third layer is the GQ-BPAOnt ontology, where the GQOnt ontology is a secondary input and used if needed, and the outcome is the goal-based candidate services' capabilities and associated QoS requirements. The derived services followed the footsteps of the BPAOntoSOA framework in adhering to the SOA principles that has resulted a richer specification of candidate software services than the BPAOntoSOA framework.

The GQ-BPAOntoSOA framework is described as being ontology-based, strategically driven, domain independent, and has used a novel alignment approach to derive a goal-based and quality-linked Riva BPA. The framework has used the original SI method used in the BPAOntoSOA framework in order to adhere to the SOA principles. Finally, the GQ-BPAOntoSOA framework contributes to bridging the gap between goal-oriented models and BPAs (one of the main outcomes of this research).

- **The GQOnt Ontology**

It has been earlier mentioned that the GQOnt ontology instantiation is the main outcome of the first layer of the new framework. The GQOnt ontology conceptualised the BSV for a given organisation. A BSV of an organisation is represented using the integration of the quality models into a set of interrelated GO models. In this ontology, the siGoal ontology has been developed on the behalf of the interrelated GO models and the sQuality ontology on the behalf on the quality models (e.g., NFR framework).

The GQOnt ontology is a key requirement in order to carry out the desired alignment with "as-is" BPAOnt ontology. Moreover, the GQOnt ontology has paved the way for enriching the Riva method embodied in the original srBPA ontology. In another words, the GQ ontology assistsed in generating a semantic representation of goal-based and quality-linked Riva BPA.

The GQOnt ontology does not only contribute to the GQ-BPAOntoSOA framework, yet it is an independent component that can be employed in order to represent the BSV of an organisation. This BSV has assisted in arriving at a common understanding about dependencies between the involved strategic entities (e.g., actors) and the rationales (e.g., goals) beyond these dependencies. The GQOnt ontology has the function of reserving all the required knowledge needed for understanding a BSV for an organisation. This repository reserves the agreed concepts and relationships between them. Accordingly, the BSV contributes towards achieving an early agreement regarding business goals and their associated tactics (i.e., tasks or operationalisations that address a given goal or soft goal) and it could be reused to review the organisation's business goals and associated soft goals. This facilitates the comprehension and communication between the stakeholders to determine the responsibility of each actor and what intentions they would achieve depending on other actors or not. Moreover, this repository is anticipated to formally represent how possible it is to satisfy and achieve highest-level business goals using the strategic rationale relationships.

On one hand, the GQOnt ontology provides many benefits for the GQ-BPAOntoSOA framework such as the alignment with the Riva BPA (one of the research outcomes). Moreover, it represents a key extension of the original BPAOntoSOA framework. On the other hand, the GQOnt ontology extends its benefits to the requirements engineering process. The anticipated benefits are as follows:

1) Since the GQOnt ontology stimulates the semantic derivation of a Riva BPA, this feature makes the GQOnt ontology independent from the Riva-based BPA existence status. Moreover, the GQOnt ontology is based on the separation of concerns principle where the goal-related elements (i.e., hard goals) and the quality-related elements (i.e., soft goals) are separated. However, they adapt to live in the same goal-oriented models (i.e., BSV) in order to explicitly and early represent the relation between the embodied functional and non-functional requirements of software services. Thus, the GQOnt ontology is anticipated to contribute to the elicitation activity within the RE process.

2) It is apparent that the GQOnt ontology involves the semantic representation of the hard and soft goals that are tactically required to fulfil an ultimate goal in an organisation. Therefore, the GQOnt ontology is considered as a repository of knowledge that comprises business goals or objectives and their fulfillers in an organisation. Hence, the GQOnt ontology could be reused to review the organisation's business goals and associated soft goals in the very early stage of the RE processes. Consequently, the GQOnt ontology is anticipated to assist the requirements engineer in validating the goal-based/driven requirements' completeness, correctness and consistency, as they must be specified to address the business goals and quality requirements for goal-based business-driven systems development. Moreover, the GQOnt ontology contributes to pave the way improving an organisation using hard and soft goals in the GQOnt ontology. This is because using goals appear as a simple and clear way for improving business organisations.

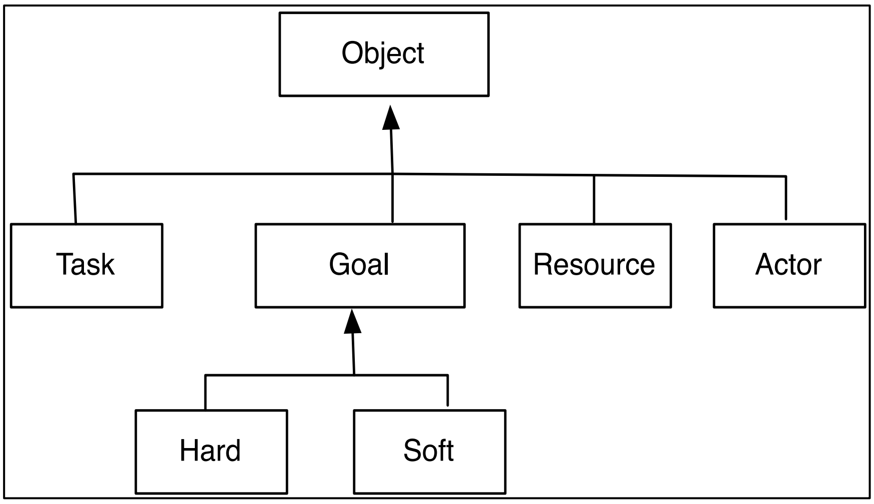


Figure 8.1: Simple Meta Model For Generalizing the Relation Between Object-based BPAs and GO Models.

❖ **Bridging the Gap Between Goal-Oriented Models and Business Process Architectures Via a Novel Alignment Approach**

This research has contributed to bridging the current gap between goal-oriented models and business process architectures and particularly the object-based ones (e.g., Riva BPA) (Dijkman et al, 2011). In particular, this research has paid a special attention in the bridging approach to the *i** and the NFR frameworks (i.e., approaches represented goal-oriented models) and to the Riva BPA modelling method in order to answer RQ2. However, from the bridging findings, the proposed bridging may be generalised to object-based BPAs and GO models trough using simple Meta modelling representation as shown in Figure 8.1.

The bridging has been carried out using a novel alignment approach of Riva-based BPAs with the organisation's BSV, which embodies goal-oriented and soft goal-oriented models. The proposed bridging approach is based on detecting EBEs found in the BSV, as they form the starting point to derive a Riva BPA. The detection involved deriving EBQs, as the bridging of the original BPAOnt with the GQOnt alerted for a required refinement within the original Riva method in order to adapt to the notion of the GQOnt and hence, deriving a GQ-Riva BPA. The notion of detection (i.e., deriving new or reusing pre-existing EBEs) is preferred to the brainstorming activity that maybe considered as a time consuming activity and not always reliable. This is because the brainstorming activity is based on a group of people that requires a preparation, where people vary in their skills and knowledge and therefore they can do bad or good brainstorming. Also, the brainstorming activity results maybe ambiguous or confidential. Few lessons have been learned from this alignment and they are:

Lesson 1. Since the notion of alignment is embedded in the GQ-BPAOntoSOA framework, then this has facilitated adapting to the strategic and functional changes in a dynamic environment. Accordingly, this point brings the **dynamism** characteristic for the GQ-BPAOntoSOA framework regarding business goals and quality requirements. In addition, this contributes to classify the processes into: novel processes, pre-existing with required redesign processes and finally pre-existing with no required redesign processes. This process classification appears significant, as it informs the business analyst or designer ahead about changes happened within an organisation that may require a particular response or attention. The three process categories are explained in Table 8.1. Moreover, this classification has enriched the original classification of processes in a Riva BPA (e.g., CPs, CMPs and CSPs). Hence, a business process in a Riva BPA is described now from two dimensions using the two classifications in Figure 8.2.

Lesson 2. The GQOnt ontology implicitly operates as a validation mechanism in the proposed alignment approach through restricting the derivation of EBEs from a BSV. In particular, all EBEs for the GQ-Riva BPA should originally stem or exist in the GQOnt ontology otherwise they are not counted. This contributes to obtaining higly validated and added EBEs for the building of a GQ-Riva BPA

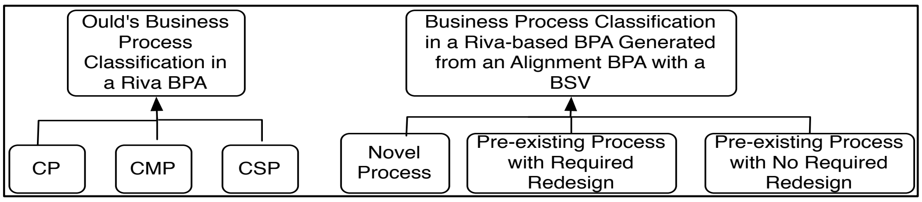


Figure 8.2: Two Business Process Classifications in a Riva-based BPAs.

Table 8.1: The Business Process Classification Generated from Aligning Riva-based BPA with a BSV.

Perspective	Novel Process	Pre-existing Process with a Required Redesign	Pre-existing Process with no Required Redesign
Attention required	Requires higher attention than the rest categories.	Requires higher attention than the pre-existing processes with no required redesign and less then the novel processes.	Requires the least attention between the rest categories.
Design	Requires entire design. Also, it requires connecting it to other	Requires partial redesign.	Requires very light or no redesign.
Linking to other processes	Requires a full study or research and understanding of the other related processes.	Requires a partial study or research and understanding of the other related processes.	May or may not requires a light study or research and understanding of the other related processes.
Business process validation	Requires identifying new validation mechanism.	Requires revisiting and amending pre-existing validation mechanism with respect to changes.	May or may not require a light amendment to the pre-existing validation mechanism.

Lesson 3. Bridging GQOnt and BPAOnt ontologies in the GQ-BPAOntoSOA framework has influenced the service identification process. The bridging has manifested a sequence that some of the goals in the GQOnt ontology are EBEs but not UoWs (e.g., IH-Gs), where some goals are not EBEs at all (e.g., HBG). However, most of the goals in the GQOnt ontology appeared as EBEs and UoWs too (e.g., goal and goal of SD). Hence, the resultant UoWs from the bridging derive corresponding CPs and CMPs in the 1st cut architecture. Each UoW ends with a corresponding CP and CMP or a corresponding CP only in the 2nd cut architecture (Ould, 2006), where both situations consider that the CPs and/or CMPs are generated from the bridging. Consequently, the resultant Riva BPA was employed for the identification of services

that are resultant from the bridging too. The bridging manifests another similar sequence with respect to soft goals, EBQs and QoS requirements. The two sequences are simplified in Figure 8.3. Hence, the bridging-driven process classification (in lesson 2) allows creating a corresponding classification of services in the GQ-BPAOntoSOA framework, as shown in Table 8.2. In this table, the categories are described from the point of view of three perspectives (i.e., capabilities, QoS requirements and requirements validation). However, the requirement validation perspective shows anticipation rather than being actually part of this research. This bridging-driven service classification is significant, as it alerts ahead the requirement engineers about changes applied within business-driven software services that may entail an attention. Hence, the requirement engineer can be informed about the changes without going back to business analysts for a negotiation.

Finally, the proposed alignment approach has demonstrated, using the CCR processes case study, that the "as-is" Riva BPA is a subset of the "to-be" GQ Riva BPA. However, this is not always the case. This is because the CCR processes did not change the business goals or quality requirements. Thus, the as-is CCR Riva BPA has been integrated with goals and quality requirements only. Consider that that the CCR processes' business goals or quality requirements have been amended, then it is not accurate to consider the "as-is" Riva-BPA subset of the "to-be" Riva BPA. Validating this assertion needs further research.

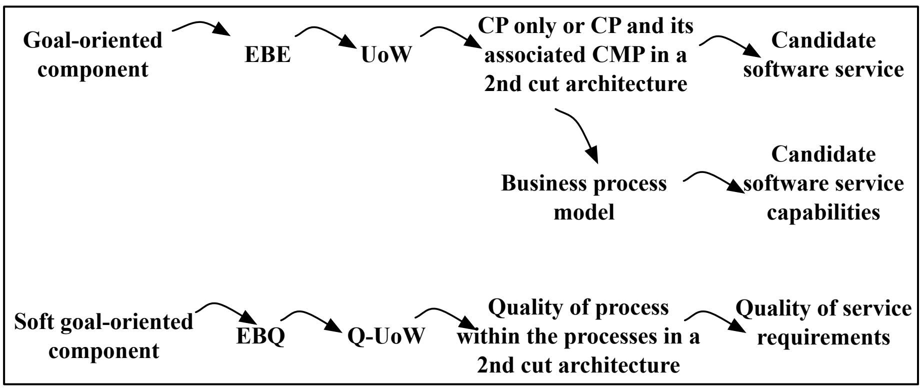


Figure 8.3: The Transition Stages of Goal and Soft Goal Components Aligned with a Riva BPA.

Table 8.2: The Candidate Software Services' Classification Generated from Aligning Riva-based BPA with a BSV.

Perspective	Novel Service	Pre-existing Service with a Required Reengineering	Pre-existing Service no Required Reengineering
Capabilities	New capabilities should be identified.	The pre-existing capabilities are analysed with respect to a change in a GQ Riva BPA generated from an alignment with a BSV.	The pre-existing capabilities are remained or lightly amended with respect to a change in a GQ Riva BPA generated from an alignment with a BSV.
QoS requirements	New QoS requirements should be identified.	The pre-existing QoS requirements are analysed with respect to a change in a GQ Riva BPA generated from an alignment with a BSV.	The pre-existing QoS requirements are remained or lightly amended with respect to a change in a GQ Riva BPA generated from an alignment with a BSV.
Requirements validation	New requirements validation mechanism is needed.	The pre-existing validation mechanism is amended with respect to the resultant capabilities and QoS requirements.	The pre-existing validation mechanism is remained or lightly amended with respect to the resultant capabilities and QoS requirements.

- **Enhancing the Service Identification Approach Used in the BPAOntoSOA Framework**

  This research has enhanced the service identification method used in the BPAOntoSOA. Although the SI method in (Yousef, 2010) considered addressing the SOA principles, still it did not consider addressing goals and quality requirements. The GQ-BPAOntoSOA framework extended the BPAOntoSOA framework by integrating the business goals and quality requirements. Hence, this requires enhancing the BPAOntoSOA's SI method used through considering the addressing of goals and quality requirements. The GQ-BPAOntoSOA employed an enriched simple goal-based and quality-linked amended SI method, which is originally extended from the BPAOntoSOA framework. This research outcome contributes to answer RQ3.

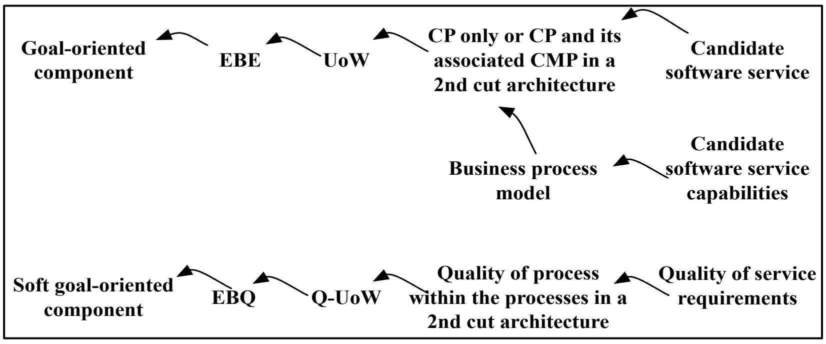


Figure 8.4: The Transition Stages from Candidate Software Services to their Corresponding GO Model.

The refinements involved extending the notion of the RPA cluster by letting the cluster members be distinguished from their goals and the clusters as well. That is, each member in the cluster is identified with goals and each cluster is identified by its goals as well. This similarly happens with refinements regarding quality requirements for the members and the clusters too.

The original SI approach was based on the BPAOnt and the refined SI approach is based on the GQ-BPAOnt, where both ontologies semantically represent the Riva BPA and associated BPMs for an organisation. Using the Riva method in both ontologies has credited the Riva method in being effective and reliable method that adapts with the integration of business goals and quality requirements in addition to being a consistent BPA model used again for the identification of candidate software services. The integration of a BSV into the original BPAOntoSOA framework has influenced the input and output of each component in the original framework. For example, a goal in a GQOnt instantiation is turned into an EBE with an interesting lifetime and ended as a GQ-process in the GQ-BPAOnt that is finally defined as a goal-based and quality-linked candidate software service. Hence, this supports proving the research hypothesis.

The observed implications on the resultant SOA model are illustrated as lessons:

Lesson1. The enhanced SI approach contributes to make a SOA model consistent with respect to the integrated goals and quality requirements that are generated from understanding a business organisation. Hence, this has contributed to enriching the current SOA principles in (Erl, 2007), where each service is suggested to be identified using goals and quality requirements. The goal-based and quality linked candidate software services can be traced backward to a corresponding GQ Riva-

BPA, its associated GQ-BPMs and BSV. This backwards traceability is shown as backward transition in Figure 8.4. This lesson contributes to proving the research hypothesis.

Lesson 2. A software service classification (shown in Table 8.2) has been generated as an implication from enhancing the original SI approach. Moreover, this classification is aided with a NFR classification (i.e., shown in Section 6.5) as guidance for software service-oriented systems. This classification contributes to cascade to the software services' design, implementation and testing. In other words, this classification informs ahead the designer and developer about effort needed for achieving their roles. However, this assertion requires further research.

❖ **Extending the Riva Method of Ould**

This research has extended the Riva method proposed in (Ould, 2005) to a goal-based and quality-linked BPA modelling method. The enriched Riva BPA is an outcome from the proposed alignment with the up-to-date BSV for an organisation.

The new representation of the Riva-based BPA reveals in the explicit representation of the quality requirements. For example, the new concepts are EBQ, Q-UoW and QoP. In addition, the Riva-based BPA elements are goal-driven from the BSV for an organisation rather than being generated from a brainstorming activity. For example, the Ould's Riva method is based on brainstorming the EBEs. However now, the brainstorming of EBEs has been replaced by detecting them from the BSV models. EBQs are detected as well as both EBEs and EBQs are the drivers of the refined Riva method. It is anticipated that the detection is very much better than brainstorming. This is because the detection is carried out from a BSV and thereby the generated EBEs are employed in the Riva BPA to address business goals and the desired quality requirements.

Accordingly, the goal-based and quality-linked EBEs assist in performing gap analysis in the detection of EBEs from the strategic view of the organisation to identify "as-is" EBEs that are not goal-based and to detect missing EBEs that are goal-based and thereby required in the Riva BPA. Having a GQ-Riva BPA has given lessons:

Lesson 1. Integrating goals and quality requirements into a Riva-based BPA contributes to enriching the business process identification in the business process management. That is, quality models are associated, if needed, in order to represent the integrated quality requirements. Hence, this paves the way for their explicit representation in the resultant GQ BPMs.

Lesson 2. The identification of quality requirements commenced from a BSV though detecting them in the GO models. However, it is possible to identify quality requirements through brainstorming activity without having a BSV present. In this situation, a business analyst should take into account the unreliability of the brainstorming activity as discussed in the third research outcome (i.e., bridging the gap between GO models and a Riva-based BPA). Moreover, deriving quality requirements from a BSV and/or BPA is highly consistent, as they stem from the understanding of a business organisation.

Lesson 3. A GQ Riva-based BPA is ideal candidate model with its associated BPMs that contribute to assist the business experts and software experts in establishing highly consistent, complete and correct requirement specifications. Moreover, the involved stakeholders are interested in utilising the generated requirement specification in order to carry out the design, implementation and testing activities within a software development process. For example, a GQ Riva BPA design implication is observed from the driven candidate software services, their associated capabilities and QoS requirements as shown in Figure 8.3. By and large, a GQ Riva BPA represents a new and rich knowledge regarding the behavioural and non-behavioural perspectives for BPs compared to the original Riva BPA.

However, the GQ Riva BPA modelling method is limited, as the method did not consider risks and their mitigation mechanisms in an ultimate Riva BPA model. In addition, the time element and designing cost of processes that constitute a GQ-Riva BPA are absent.

- **Goals-to-Services Traceability Network**

The chain of the GQOnt, the GQ-BPAOnt and the semantic generation of goal-based and quality-linked candidate services, respectively has created a semantic traceability network in both forward and backward directions. The traceability network starts from the goals and soft goals in the GQOnt ontology and ends in the generated goal-based and quality-linked services and vice versa.

This network contributes to trace any element in addition to providing how an element in linked to others (e.g., determine the deriving and derived elements). Hence, this contributes further to determine the significance of each element in the network through investigating the quantity of the derived elements from a particular element. For example, few EBEs appear not significant; as they are not considered as UoWs and hence they are not considered as candidate software services using Figure 8.3. Whereas some EBEs appear significant, as they derive corresponding candidate software services at the end as shown in Figure 8.3.

- **The Derivation of the GO Models (the BS Model, the HSD Model and the *i** Framework Models) from Role-Oriented, Goal-Based and Quality-Integrated BPMs**

In Chapter 4, an algorithm has been proposed that is concerned with deriving the GO models of a given BPMs for an organisation. The algorithm in Chapter 4 is composed of six algorithms from 4.1 to 4.6. The main input is role-oriented, goal-based and quality-linked BPMs in order to output the GO view within a BSV. Thus, this algorithm contributes to generate a new knowledge from the GQR-BPMs. Also, this algorithm contributes to structurally generate the GO view within a BSV instead of using interviews, unstructured or time consuming approaches. Deriving GO models is considered as a difficult task to achieve, where this algorithm has facilitated the derivation.

- **The Effectiveness of the *i** and the NFR Frameworks**

In this research, the *i** framework and the NFR framework appeared effective in the GQ-BPAOntoSOA framework. This effectiveness is inferred from few observations that are:

1) The NFR framework has been structurally integrated into the i* framework in order to represent a BSV of an organisation. The structured integration manifested a harmony between the *i** framework models and their associated NFR framework models. This harmony contributes to represent a rich BSV for an organisation.
2) The synthesis of the *i** framework and the NFR framework has generated most of the EBEs and EBQs, where the rest of them were generated from the HSD model in order to model a GQ-Riva BPA. This is apparent in the quantitative results generated in Chapter 7. Figure 8.2 illustrates the transition relation between GO components in the *i** framework models and the processes generated in the GQ

2nd cut architecture. Also, Figure 8.2 depicts the transition relation between the soft goal components in the NFR framework models and the integrated QoP into processes in the GQ 2nd cut architecture.

3) The effectiveness of the *i** and the NFR frameworks manifested in the newly derived candidate software services and their associated QoS requirements, as shown in Section 7.3.5. In particular, some of the *i** framework-driven EBEs turned into UoWs and consequently into BPMs in a GQ Riva BPA that is used to identify candidate software services. Similarly, the transition relations from the goal and soft goal components to the identified candidate software services, their capabilities and associated QoS requirements, respectively are shown in Figure 8.2.
4) The *i** framework and the NFR framework are characterised as domain independent GO approaches.

In the light of the above, the *i** framework and the NFR framework contribute to generate knowledge for business experts in a GQ Riva BPA, using the proposed alignment approach, and candidate software services for service-oriented software experts. In short words, the *i** framework and the NFR framework contribute to derive a higher-level of consistent, complete and correct software service specifications, as they stem from an understanding of a business organisation from the goal and quality perspectives. Since the GQOnt ontology embodies the semantic representation of the *i** and the NFR frameworks (i.e., domain independent), then a generic Meta model can be designed in order to generalize the representation of GO model and its associated quality-oriented models.

- **An Evaluation Framework for the GQ-BPAOntoSOA Framework**

This research work derived an evaluation framework that is considered as one of the research outcomes as presented in Section 7.3. The purpose of the evaluation framework was to assess the GQ-BPAOntoSOA framework in order to inform its effectiveness after evaluating every carried out work embedded in the GQ-BPAOntoSOA framework. The evaluation framework contributed to prove the research hypothesis at the end.

8.2 Comparing the Original and the New Framework

Table 8.3 provides a brief comparison between the original BPAOntoSOA framework and the newly extended GQ-BPAOntoSOA framework. The yellow shaded entries mean that the old and the new framework share the same feature.

Table 8.3: Comparing the Old and the New Frameworks

Perspective	Old Framework	New Framework
Architectural layers	Two	Three
Riva conceptualisation	Followed Ould's approach	Extended Ould's approach
Major input requirement/ Framework driven by	Riva-BPA and associated BPMs	A BSV, the pre-existing Riva BPA and its associated BPMs.
Output and adherence to SOA principles	Candidate services with associated capabilities. The output adheres to SOA principles.	Goal-based candidate services with associated capabilities and QoS requirements. The output adheres to SOA principles.
Ontology employed	OWL-DL	OWL-DL
Ontology integration	Linked two ontologies: srBPA and sBPMN	Linked 3 ontologies: Extended srBPA, GQOnt, remained the link between srBPA and sBPMN.
Alignment level	Pure Functional	Strategic and functional
Domain dependency	Domain independent	Domain independent
Models presented	1- Riva-based BPA 2- Business processes using BPMN	1- Goal-oriented models using i* and NFR framework. 2- Riva-based BPA. 3- Business processes using BPMN.
Traceability	Supports traceability in both directions	Supports traceability in both directions
Adaptability to changes/ dynamism	Adapts to: - 1- Behavioural changes.	Adapts to: - 1-Strategic changes. 2-Behavioural changes. 3- Non-behavioural changes.
Integration of quality requirements	Does not support the integration of quality requirements.	Supports the integration of quality requirements starting from the BSV until deriving the ultimate candidate software services.

8.3 Thesis Chapters, Research Questions and Research Process Phases

This section recalls Figure 1.3 used in Chapter 1 in order to show how each chapter was employed for addressing related research questions and research process phases. Figure 8.5 shows this relation.

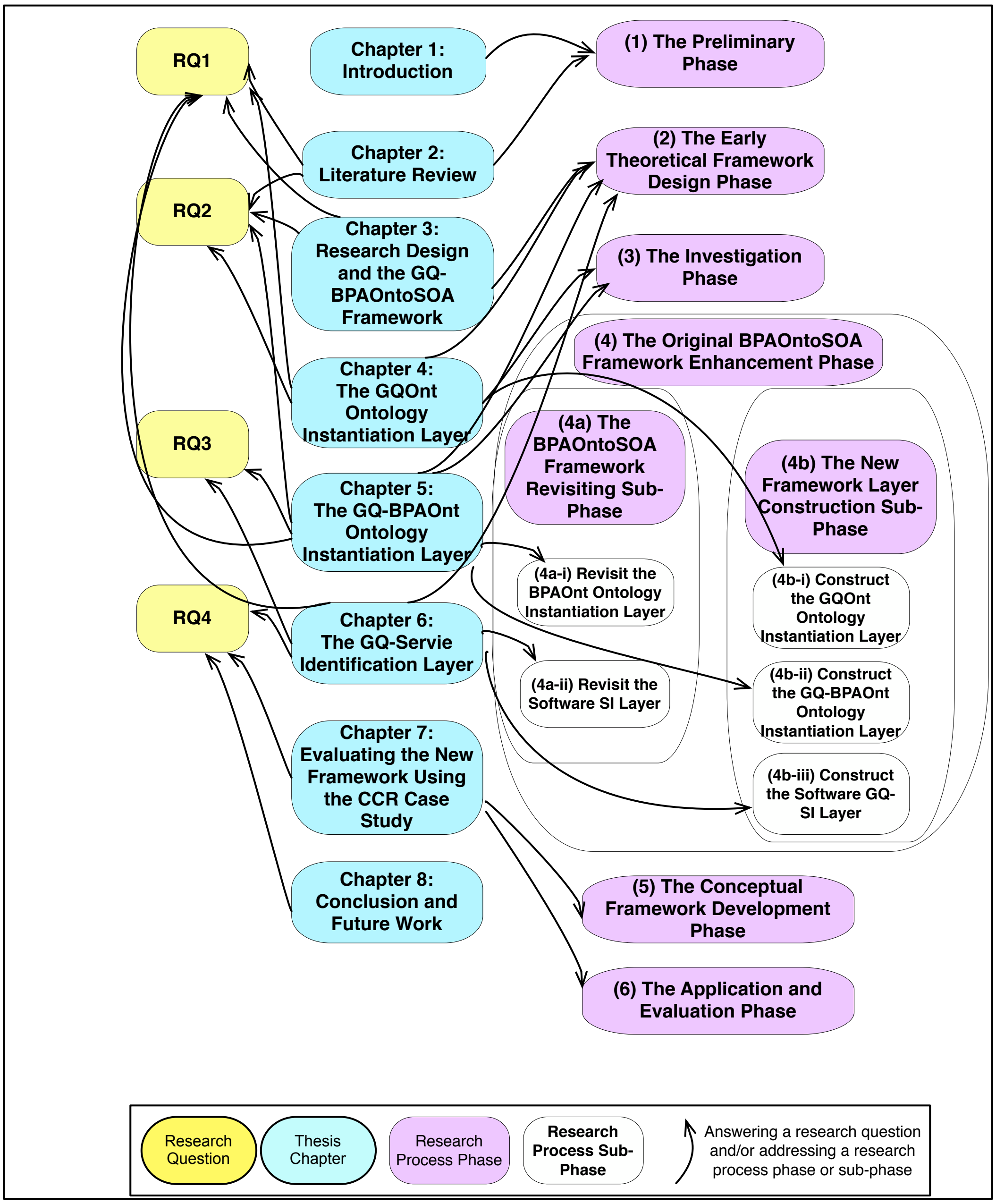


Figure 8.5: Relating the Presented Thesis Chapters with the Research Questions and Research Process Phases.

8.4 Future Work

This section suggests further future work that are anticipated to contribute to this research.

- **Enhancing the GQ-BPAOntoSOA Framework by Considering the Management of Associated Risks**

  The GQ-BPAOntoSOA has considered the derivation of goal-based candidate service capabilities along with the associated QoS requirements. However, the risks that may arise if any of the capabilities and/or QoS requirements are not achieved, as they are not considered yet. Risk management is proposed to be addressed in the GQOnt ontology instantiation layer in the GQ-BPAOntoSOA framework, where risk concepts are to be embedded. For example, the vulnerability level of dependencies between actors in the *i** framework were not addressed. In addition, the soft goal claim was not considered in the designed SIG graphs. Managing risks involves determining the priorities of hard and soft goals.

- **Cost Estimation for the GQ-BPAOntoSOA Driven Candidate Software Services using the GQOnt and BPAOnt Ontologies**

  The generated candidate services from the GQ-BPAOntoSOA framework did not consider their associated cost. The absence of the cost regarding addressing service capabilities and QoS requirements is anticipated to yield to unestimated design costs and thereby significant implication on the software services' implementation and their overall development. An early stage of cost estimation may be considered using the alignment implication classification.

- **Migrating the GQ-BPAOntoSOA Framework to the Cloud Paradigm**

  The cloud paradigm is a recent emerging trend of distributed computing. The GQ-BPAOntoSOA framework followed the BPAOntoSOA framework footsteps in operating the generated candidate services in service-oriented environment taking in to account their adherence to the SOA principles. However, many current software services have been migrated to operate in the cloud paradigm, where users access the services online. Accessing a cloud-based software service is carried out from anywhere since there is an access to the Internet. Thus, this increases the opportunity of delivering the software service to as much as possible number of users. Therefore, competing with other similar business services in the market. Hence, this requires persistent improvements on the released services. The GQ-BPAOntoSOA framework is anticipated to assist in the desired improvement due to the employment of business hard goals and soft goals (i.e., QoS requirements).

- **Developing a Diagramming Tool for Modelling the BSV and GQ-Riva BPA**

  The massive number of generated interrelated models necessitates for a diagramming tool than can establish the link from the BSV components and instances of the GQ-Riva BPA with their associated BPMs, and vice versa.

- **Development of Domain Dependent GO Models**

  In order to specialise goals and soft goals for a particular business domain, it is suggested to develop domain dependent GO models to include domain specialised hard and soft goals. For example, healthcare, manufacturing and banking GO models. A domain dependent GO model is anticipated to require a domain dependent specialised skills and practices from the designers.

- **Generalise the Development of a Framework to include Meta Models for Goal-based and Object-based BPA Modelling Methods**

Although BPA modelling methods have been classified into five categories in (Dijkman et all, 2011), Meta models that generalise each category representation is still absent. Since this research was interested in involving particular goal-based and object-based BPA modelling methods, it is needed to develop a framework that generalise their representation using Meta models. Hence, this is anticipated to broad their usage and application.

References

Aburub, F. (2006) Process Modeling for Business Process Improvement, with particular reference to health care processes, PhD thesis, University of the West of England.

Aburub, F. and Issa, A. (2012) A Business Process Modelling-based approach to investigate complex processes. Business Process Management Journal. 18(1), pp.122-137.

Aburub, F., Odeh, M. and Beeson, I. (2007) Modeling Non-Functional Requirements for Business Process. *Information and Software Technology Journal*. 49(11-12), pp. 1162-1171.

Ahmad, M. and Odeh, M. (2012). Semantic Derivation of Enterprise Information Architecture from Business Process Architecture. Proceedings of 22nd International Conference on Computer Theory and Applications (ICCTA2012), 13-15 October, Alexandria, Egypt.

Ameller, D. and Franch, X. (2008) Service Level Agreement Monitor (SALMon). *In: The Seventh IEEE International Conference on Composition-Based Software Systems (ICCBBSS 2008)*, Madrid 25-29 February 2008, pp. 224-227.

Anton, A., McCracken, W. and Potts, C. (1994) Goal Decomposition and Scenario Analysis in Business Process Reengineering. In: 6th Conference on Advanced Information Systems Engineering (CAISE '94), Utrecht June 1994, pp. 94-104.

Aversano, L., Grasso, C. and Tortorella, M. (2012) "A Literature Review of Business/IT Alignment Strategies," Proc. 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012), Procedia Technology, vol.5, 2012, pp.462-474, doi:10.1016/j.protcy.2012.09.051.

Baader, F., Horrocks, I. and Sattler, U. (2002) Description Logics for the Semantic Web. *Kunstliche intellihenz*. 16(4), pp. 57-59.

Barros, O. and Julio, C. (2011) Enterprise and Process Architecture Patterns. *Business Process Management Journal.* 17(4), pp.598-618.

Bittencourt, R., Araujo, R. and Cappelli, C. (2010) Experiences on the use of business models for identifying quality requirements for information systems. *In: Seventh International Conference on the Quality Information and Communications Technology (QUATIC)*, Porto 29 September – 2 October 2010, pp. 192-197.

Boehm, B., Brown, J. and Lipow, M. (1976) Quantitative Evaluation of Software Quality. *In: The Second IEEE International Conference on Software Engineering*, California, pp. 592-605.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F. and Mylopoulos, J. (2004) Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems Journal*, 8(3), pp.203-236.

Burgess, C., Krishna, A. and Jiang, L. (2009) Towards Optimising Non-Functional Requirements. *In: The Ninth International Conference on Quality Software IEEE*, South Korea 24-25 August 2009, pp. 269-277.

Cabral, L. and Domingue, J. (2009) Translating Semantic Web Service Based Business Process Models. In: Asia-Pacific Service Computing Conference (APSCC 2009), Singapore 7-11 December. pp.1-6.

Chan, Y. and Reich, B. (2007) State of the Art IT alignment: What Have We Learned?. *Journal of Information Technology.* 22 (not known) pp.297-315, doi:10.1057/palgrave.jit.2000109.

Chen-Burger, Y-H., Tate, A. and Robertson, d. (2002) Enterprise Modelling: a declarative approach for FBPML. *In: Proceedings of European Conferences Artificial Intelligence, Knowledge Management and Organisational Memories Workshop*, Lyon. France 22 July 2002.

Chung, L., Nixon, B. and Mylopoulos, J. (2000) *Non-Functional Requirements in Software Engineering*. London: Kluwer Academic Publishers, .

Clements, P. and Bass, L. (2010) Using Business Goals to Inform Software Architecture. *In: 18th IEEE International Requirements Engineering Conference,* Sydney 27 September – 1 October 2010, pp. 69-78.

Collis, J. and Hussey, R. (2009) *Business Research A Practical Guide for Undergraduate and Postgraduate Students*. 3rd Edition, Houndmills: Palgrave Macmillan.

Creswell, J.W. (2009) Research design: qualitative, quantitative, and mixed methods approaches. 3rd Edition, London: SAGE.

Cysneiros, L., Leite, J. and Neto, J., (2002) A framework for integrating non-functional requirements into conceptual models. *Requirements Engineering*. 6(4).pp. 97-115.

Davis, A. (1992) *Software Requirements: Objects, Functions and States.* Upper Saddle River: Prentice Hall.

Decreus, K., Snoeck, M. and G. Poels, (2009) "Practical Challenges for Methods Transforming i* Goal Models into Business Process Models," *In: Proc. 17th IEEE International Requirements Engineering Conference,* IEEE computer, August-Septemeber 2009, pp.15-23, doi: 10.1109/RE.2009.25.

Dijkman, R. Vanderfeesten, I. and Reijers, H. (2011). "The Road to a Business Process Architecture: An Overview of Approaches and their Use", Beta Publication.

Djouab, R. and Suryn, W. (2006) An ISO/IEC standards-based quality requirement definition approach: Applicative analysis of three quality requirements definition methods. *In: IEEE International Symposium on Industrial Electronics (ISIE 2006)*, Montreál 9-12 July 2006, pp. 3231-3239.

Dobson, G., Hall, S. and Kotonya, G. (2007) A Domain Independent Ontology for Non-Functional Requirements. *In: Proc. IEEE International Conference on e-Business Engineering (ICEBE 07)*, Hong Kong 24-26 October 2007, pp. 563-566.

Dobson, G., Lock, R. and Sommerville, I. (2005) QoSOnt : A QoS Ontology for Service-Centric Systems. *In: Proc. 31st EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 05)*, Porto 30 August – 3 September 2005, pp. 80-87.

Gebru, G. (2010) *Research Methods and Analysis for Business Studies: a holistic approach*. Saarbrucken, Germany: VDM Verlag Dr. Muller GmbH & Co.KG.

Eriksson, H. and Penker, M. (2000) Business Modelling with UML Canada: Wiley.

Eriksson, P. and Kovalainen, A. (2008) *Qualitative Methods in Business Research.* London: SAGE Publications Ltd.

Erl, T. (2007) *SOA: Principles of Service Design*. Upper Saddle River: Prentice Hall.

Galster, M. and Bucherer, E. (2008) A Taxonomy for Identifying and Specifying Non-functional Requirements in Service-Oriented Development. *In: IEEE Congress on Services Part I*, Honolulu 6-11 July 2008, pp. 345-352.

Ghauri, P. and Gronhaug, K. (2005) *Research Methods in Business Studies: A Practical Guide*. Third edition. Harlow, England: Prentice Hall

Gilb, T. (1997) Towards the Engineering of Requirements. *Requirements Engineering Journal*. 2(3), pp. 165-169.

Glinz, M. (2005) Rethinking the Notion of Non-Functional Requirements. *In: Proceedings of the Third World Congress for Software Quality*, Munich 26-30 September 2005, pp. 55-64.

Grau, G., Franch, X. and Maiden, N. (2008) PR*i*M: An *i**-based process reengineering method for information systems specification. *Information and Software Technology Journal*. 50(1-2), pp.76-100.

Green, S. and Ould, M. (2004) The Primacy of Process Architecture. *In: The 5th Workshop on Business Process Modeling, Development, and Support, in conjunction with the 16th Conference on Advanced Information Systems Engineering (CAiSE04)*, Riga.

Gruber, T.R (1993) A translation approach to portable ontologies. *Knowledge Acquisition*. 5(2), pp.199-220.

Hammer, M. and Champy, J. (1993) Re-engineering the Corporation; A Manifesto for Business Revolution. New York: Harper Business.

Heravizadeh, M., Mendling, J., Rosemann, M. (2009) Dimensions of Business Processes Quality (QoBP). In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. LNBIP, vol. 17, pp. 80–91. Springer, Heidelberg.

Hevner, A. and Chatterjee, S. (2010) Design Research in Information Systems Theory and Practice. Springer.

Horkoff, J. and Yu, E. (2011) Analyzing Goal Models – Different Approaches and How to Choose Among Them. *In: The SAC'11 Proceedings of the 2011 ACM Symposuim and Applied Computing*. Taiwan 21-24 March 2011, pp.675-682. DOI: 10.1145/1982185.1982334

Horrocks, I. (2008) Ontologies and the Semantic Web. *Communications of ACM*. 51 (12), pp.58-67.

IBM, Rational Unified Process (RUP) [online] Available from: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf . [Accessed 28 October 2013 at 16:04 pm]

Integrated Definition Method (IDEF). [online]. Available from:http://www.idef.com/. [Accessed 1 September 2011 at 12:30].

IEEE, 1993, IEEE Recommended Practice for Software Requirements Specification. IEEE standard 830-1993.

ISO/IEC. (2001) ISO/IEC 9126-1 Software Engineering Product Quality – Part 1: Quality model.

Ives, B., Jarvenpaa, S. and Mason, R. (1993) "Global business drivers: Aligning information technology to global business strategy" IBM Systems Journal. 32(1), pp.143-161.

Jaramillo, A. (2011) Non-Functional Requirements Elicitation from Business Process Models. In: the fifth international conference of Research Challenges in Information Science (RCIS). Gosier 19-21st May 2011, pp.1-7.

Kavakli, V. and Loucopoulos, P. (1999) Goal-Driven business process analysis application in electricity regulation. *Information Systems*. 4(3), pp.187-207.

Kavakli, E. and Loucopoulos, P. (2005) "Goal Modelling in Requirements Engineering: Analysis and Critique of Current Methods," Information Modeling Methods and

Methodologies: Advanced Topics in Database Research, Idea Group Publishing (IGP), 2005, pp.102-124, doi: 10.4018/978- 1-59140-375-3.ch006.

Khan, Z. (2009) Bridging the Gap between Business Process Models and Service Oriented Architecture with Reference to the Grid Environment, PhD thesis, University of the West of England.

Kotonya, G. and Sommerville, I. (1998) *Requirements Engineering: Processes and Techniques*. England: Jhon Wiley.

Lamsweerde, A. (2009*) Requirements Engineering: From System Goals to UML Models to software specifications*. England: Jhon Wiley.

Kueng, p., Bichler, P., Kawalek,P. and Schreft, M. (1996) How to compose an-object-oriented business process model. Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering: principles of method construction and tool support: principles of method construction and tool support. UK, London, Chapman and Hall. pp.94-100**.**

Luftman, J., Papp, R. and Brier, T. (1999) "Enablers and Inhibitors of Business-IT Alignment". Communications of the Association for Information Systems, 1(11).

Malinova, M., Leopold, H. and Mendling, J. (2013) An Empirical Investigation on the Design of the Process Architectures, *In: 11th International Conference on Wirtschaftsinformatik, Leipzig 27th February – 1st March 2013, pp.1-15.*

Markovic, I. and Kowalkiewicz, M. (2008) Linking Business Goals to Process Models in Semantic Business Process Modeling, *In: 12th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2008*), Munchen 15-19 September 2008, pp. 332-338.

Nurcan, S., Etien A., Kaab, R., and Zouka, I. (2005) "A Strategy Driven Business Process Modelling Approach". Journal of Business Process Management. 11(6), pp. 628-649.

O'brien, L., Merson, P., and Bass, L. (2007) Quality Attribute for Service-Oriented Architectures. *In: International Workshop on Systems Development in SOA Environment (SDSOA)*, Minneapolis 20-26 May 2007 pp.3,

Odeh, M. and Kamm, R. (2003) Bridging the Gap between Business Process Models and Systems. *Information and Software Technology, Special Edition on Modeling Organisational Processes*, 45(15), pp. 1053-1060.

OMG (2011) Business Process Model and Notation (BPMN) [online]. Available from: http://www.omg.org/spec/BPMN/2.0/PDF/. [Accessed 28 July 2011 at 3:36 am].

Ould, M. (1995) *Business Processes: Modelling and Analysis for Re-engineering*. Chichester: Wiley.

Ould, M. (2006) *Business Process Management A Rigorous Approach*. Swindon: The British Computer Society.

Papazoglou, M., Traverso, P., Dustdar, S. and Leymann, F. (2007) Service-Oriented Computing: State of the Art and Research Challenges. *IEEE Computer*. 40 (11), pp. 38-45.

Pavlovski, C. and Zou, J. (2008) Non-functional Requirements in Business Process Modelling. *In: 5th Asia-Pacific Conference on Conceptual Modeling (APCCM 2008)*, Wollongong 22-25 January 2008, pp.103-112.

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. (2006) The Design Science Research Process: A Model for Producing And Presenting Information Systems Research. Journal of Management Information Systems. 24(3) pp.45-77.

Pidd, M. (2003) Tools for Thinking: Modeling in Management Science. London: Wiley.

Plosch, R., Mayr, A. and Korner, C. (2010) Collecting Quality Requirements Using Quality Models and Goals. *In: Seventh International Conference on the Quality Information and Communications Technology (QUATIC)*, Porto 29 September – 2 October 2010, pp. 198-203.

Poels, G., Roelens, B. and Snoeck, M. (2013) Research Review: Investigating Goal-Oriented Requirements Engineering for Business Processes. Journal for Database Management. 24(2), pp. 35-71.

Protégé. [online] Available from: http://protege.stanford.edu/overview/protege-owl.html. [Accessed 23 October 2011 at 11:20 pm].

Roland, C., Souvey, C., and Ben Achour, C. (1998) Guiding Goal Modeling Using Scenarios. IEEE Transaction on Software Engineering, 24 (12), pp. 1055-1071.

Sancho, P., Juiz, C., Puigjaner, R., Chung, L. and Subramanian, N. (2007) An approach to ontology-aided Performance Engineering through NFR Framework. *In proceeding of: Proceedings of the 6th International Workshop on Software and Performance, WOSP 2007,* Buenes Aires, Argentina, February 5-8, pp.125-128.

Sekaran, U. and Bougie, B. (2013) *Research Methods for Business: A Skill Building Approach*. Sixth Edition. West Sussex, United Kingdom: John Wiley & Sons Ltd.

Silvius, A.J.G (2007) Business and IT Alignment in theory and practice. In: *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07).* 3rd-6th January, pp.2-11.

Silmane, A., Pinheiro, M. and Souveyet, C. (2009) Goal Reasoning for Quality Elicitation in the ISOA Approach. In: Third International Conference on Research Challenges in Information Science (RCIS 2009). 22-24 April, pp. 39-48.

Solaimani, S. and Bouwman, H. (2012) A framework for the alignment of business model and business processes: A generic model for trans-sector innovation. *Business Process Management Journal*, 18(4), pp.655-679.

Soffer, P. and Wand, Y. (2005) On the notion of soft-goals in business process modeling. *Business Process Management Journal*, 11(6), 2005, pp.663-679, doi: 10.1108/14637150510630837.

Sommerville, I. (2007) *Software Engineering*, 8th ed. England: Addison-Wesley.

Super Project (2008a), sBPMN and sEPC to BPMO Translation. Deliverable 4.5 [online]. Available from: http://www.ip-super.org [Accessed 30th October 2013at 6:13 am].

Super Project (2008b), main page [online]. Available from: http://www.ip-super.org [Accessed 30th October 2013at 6:13 am].

Valipour, M., Amirzafari, B., Maleki, K., and Daneshpour, N. (2009) A Brief Survey of Software Architecture Concepts and Service Oriented Architecture. *In: Proceeding the Second IEEE International Conference on Computer Science and Information Technology 2009 (ICCSIT 2009)*, Chengdu 8-11 August 2009, pp.34-38.

Volere Requirements Specification Template [online] Available from: http://www.volere.co.uk/template.htm [Accessed 16th August 2013 at 11:05 pm].

Wieringa, R. (2005) "Requirements Researchers: are we really doing research?," Requirements Engineering, vol.10, Aug. 2005, pp.304-306, doi:10.1007/s00766-005-0013-6.

Young, R. (2004) The Requirements Engineering Handbook, Artec House, USA. (Boston, London)

Yousef, R. (2010) *BPAOntoSOA: A Semantically Enriched Framework for Deriving SOA Candidates Software Services from Riva-based Business Process Architecture*. Ph.D. Thesis, University of the West of England.

Yu, E. (1995) *Modeling Strategic Relationships For Process Reengineering*, Ph.D. Thesis, University of Toronto.

Yu, E., Du Bois, Ph., Dubois, E. and Mylopoulos, J. (1995) From Organizational Models to System Requirements -- A "Cooperating Agents" Approach, *Proc. 3rd International Conference on Cooperative Information Systems--CoopIS-95*, Vienna, Austria, May 9-12, 1995, pp. 194-204.

Yu, E., Giorgini, P., Maiden, N. and Mylopoulos, J. (2010) Social Modelling for Requirements Engineering, MIT Press, USA.

Appendix A: CEMS Faculty of Administration Riva-based BPA

This appendix shows the Riva BPA of the CEMS Faculty of Administration in the University of the West of England (UWE). Ould and Green conducted the study as reported in (Ould and Green, 2004). The resulted diagrams are shown from Figure A.1- A.5.

CEMS Faculty administration

List of essential business entities, with units of work highlighted

Bracketed EBEs were agreed not to be UOWs for the reason given after the item

student
(field)
(school)
(award)
(examiner)
award definition
module definition
(inspection) *see the three types below*
(teacher)
(administration)
submission [exam, coursework, or project]
(exam assessment) *see 'submission'*
(coursework assessment) *see 'submission'*
(project assessment) *see 'submission'*
(assignment)
(assignment assessment)
(student record) *in 'student'*
meeting
(direct entrant) *type of 'student'*
(external examiner payment) *too small*
(the current teaching timetable) *output of programme planning*
(the planned teaching timetable) *output of programme planning*
(ISIS) *used by, but not the responsibility of CEMS*
(definitive document) *same as 'award' or 'module'*
(course road map) *in 'definitive document'*
(MAR) *something that constrains*
(the archive)
(placement)
(Blackboard)
the Programme Plan [maps awards onto students onto options onto module runs]
external examiner
(invigilator)
(student fee)
(exam result)
student withdrawal
(error)
student request to transfer award
student appeal
late submission
university academic review event
(benchmark) *an input*
Examining Board event
quality inspection event
validation event
accreditation event
extenuating circumstance
NAPP
(special need) *in 'student'*
(student at risk) *in 'student'*
assessment offence
(student fails to turn up) *in 'student'*
lost item of work
(option collection) *in 'student'*
(university requirement to change award/option) *in 'student'*
(letter)
student problem
(student exit profile)
(HESA return)
(intake target)
(graduating student) *part of 'student'*
(Graduation Day)
Induction Week
(referral)
Referral Day
(visiting lecturer)
(international student) *type of 'student'*
(professional body) *plays a role*
(module evaluation by students) *in 'module run'*
(module evaluation report) *in 'module'*
(the UWE Student Handbook)
Award Handbook
the Faculty Handbook
(Data Protection Act) *a constraint*
module run
(award results list) *same as 'Examining Board event'*
(module results list) *in 'module run'*
exam paper
assignment definition
(report) *see types of report*
(student request to change option) *in 'student'*
(monitoring and evaluation report)
(Exam Board report) *output from 'Exam Board event'*
Student Loan Company report
(Field Leader report)
(Award Director report)
(Programme report)
(Staff/Student Committee Meeting)
(committee minutes)
(external examiner report) *subsequent to External Board event*
(external examiner response)
(student complaint) *a sort of 'student problem'*
(ad hoc request)
(question/paper fails to turn up on time) *in 'module run'*
(mark fails to turn up on time) *in 'module run'*

VENICE CONSULTING LTD
p039r001 CEMS process architecture v2.vsd
16 May 2003
www.veniceconsulting.co.uk

Figure A.1: The EBEs and UoWs for CEMS Faculty of Administration [Source: (Green and Ould, 2004), Used with the author's permission]

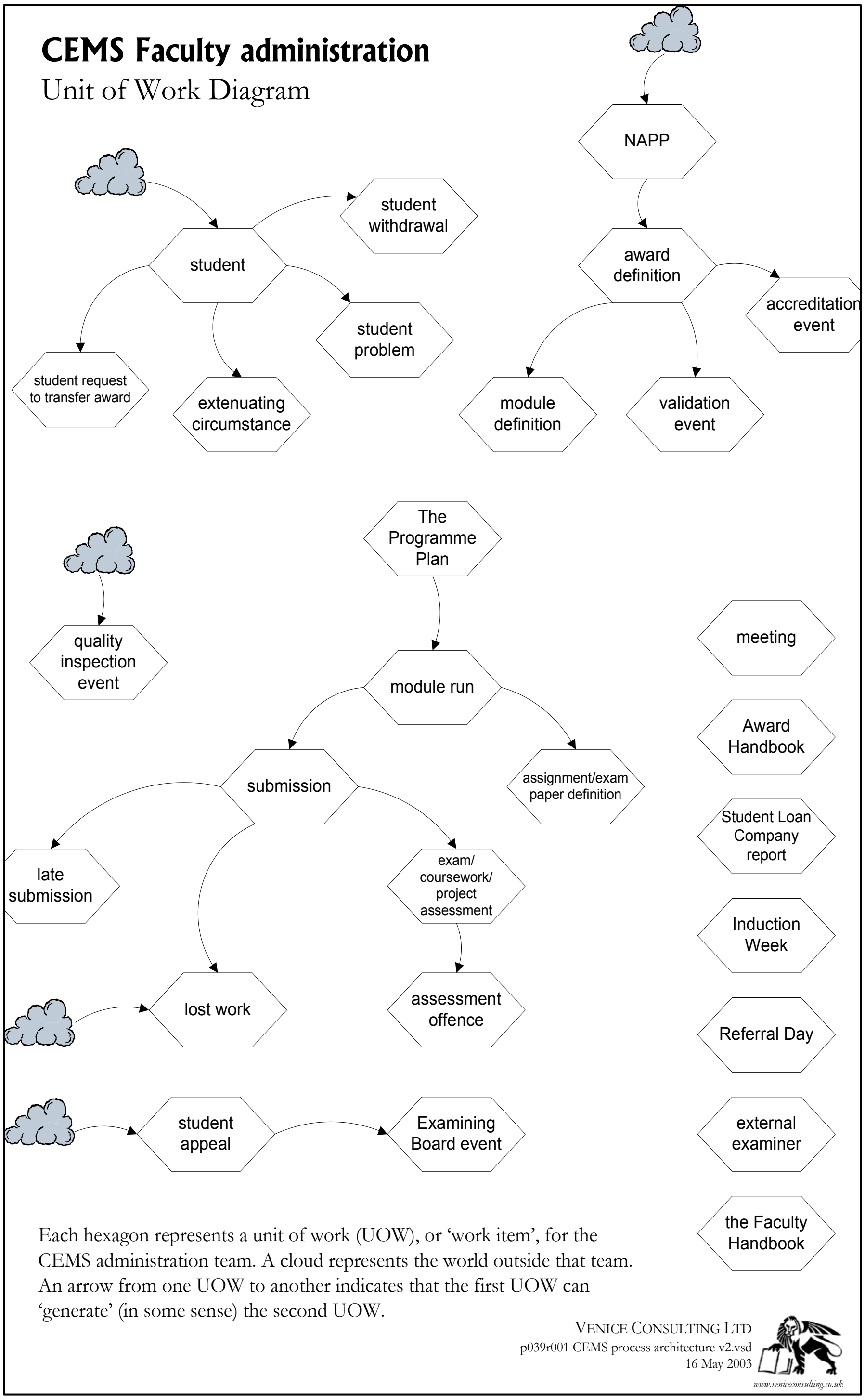


Figure A.2: The UoW Diagram for the CEMS Faculty of Administration [Source: (Green and Ould, 2004), Used with the author's permission]

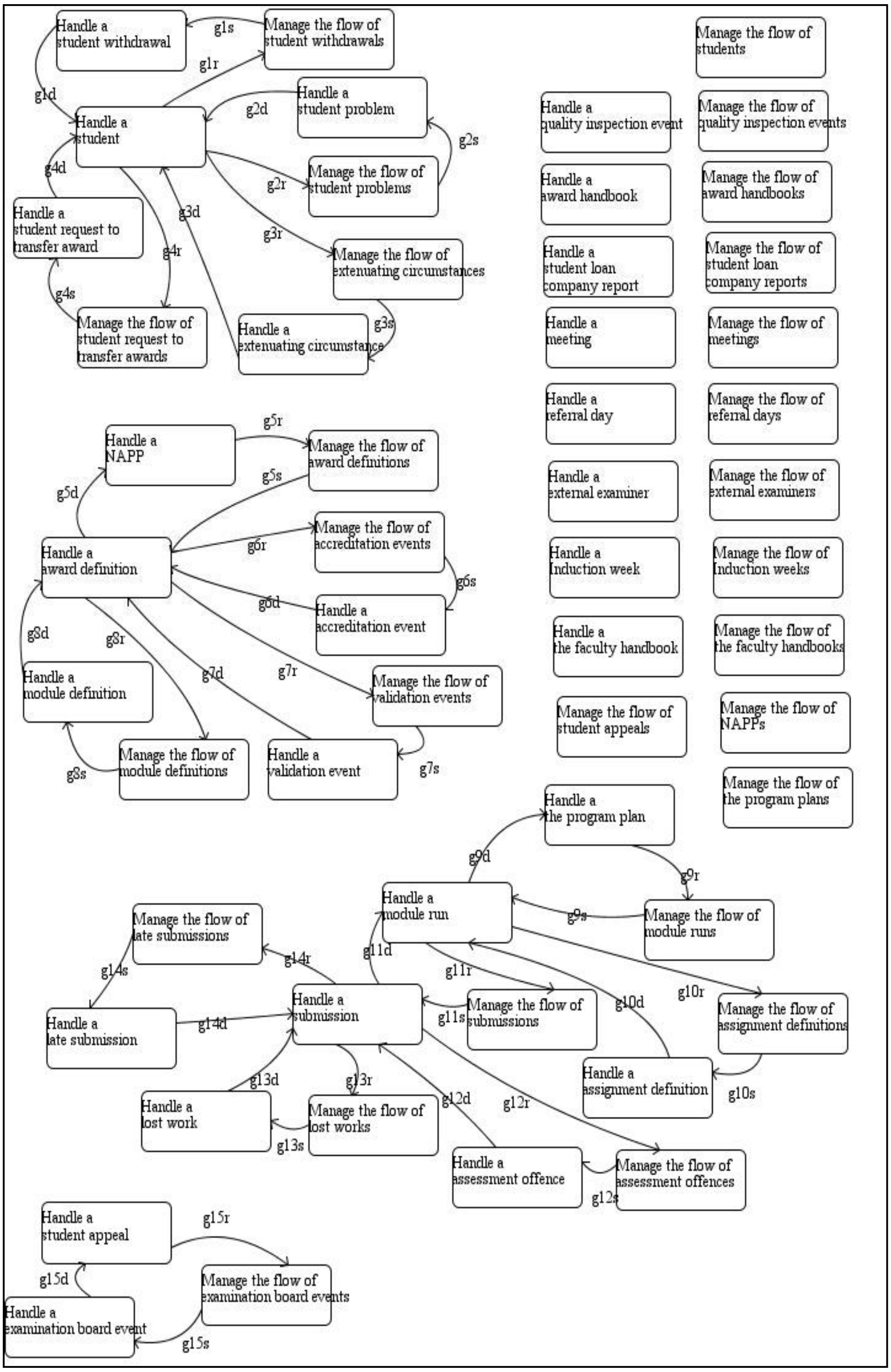


Figure A.3: The 1st Cut Architecture for the CEMS Faculty of Administration [Source: (Yousef, 2010), Used with the author's permission].

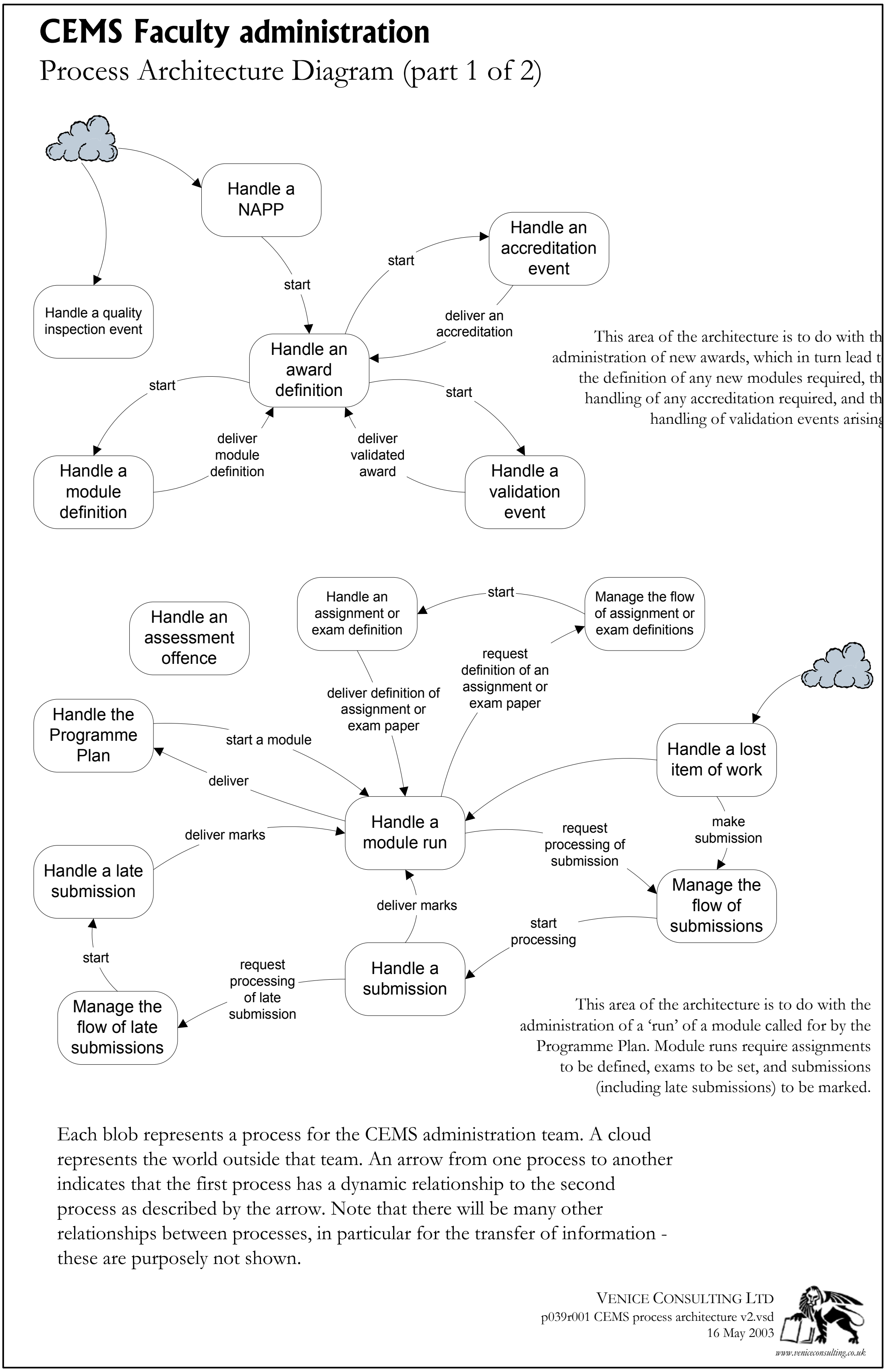


Figure A.4: The 2nd Cut Architecture for the CEMS Faculty Administration (Part 1 of 2) [Source: (Ould and Green, 2004), Used with the author's permission]

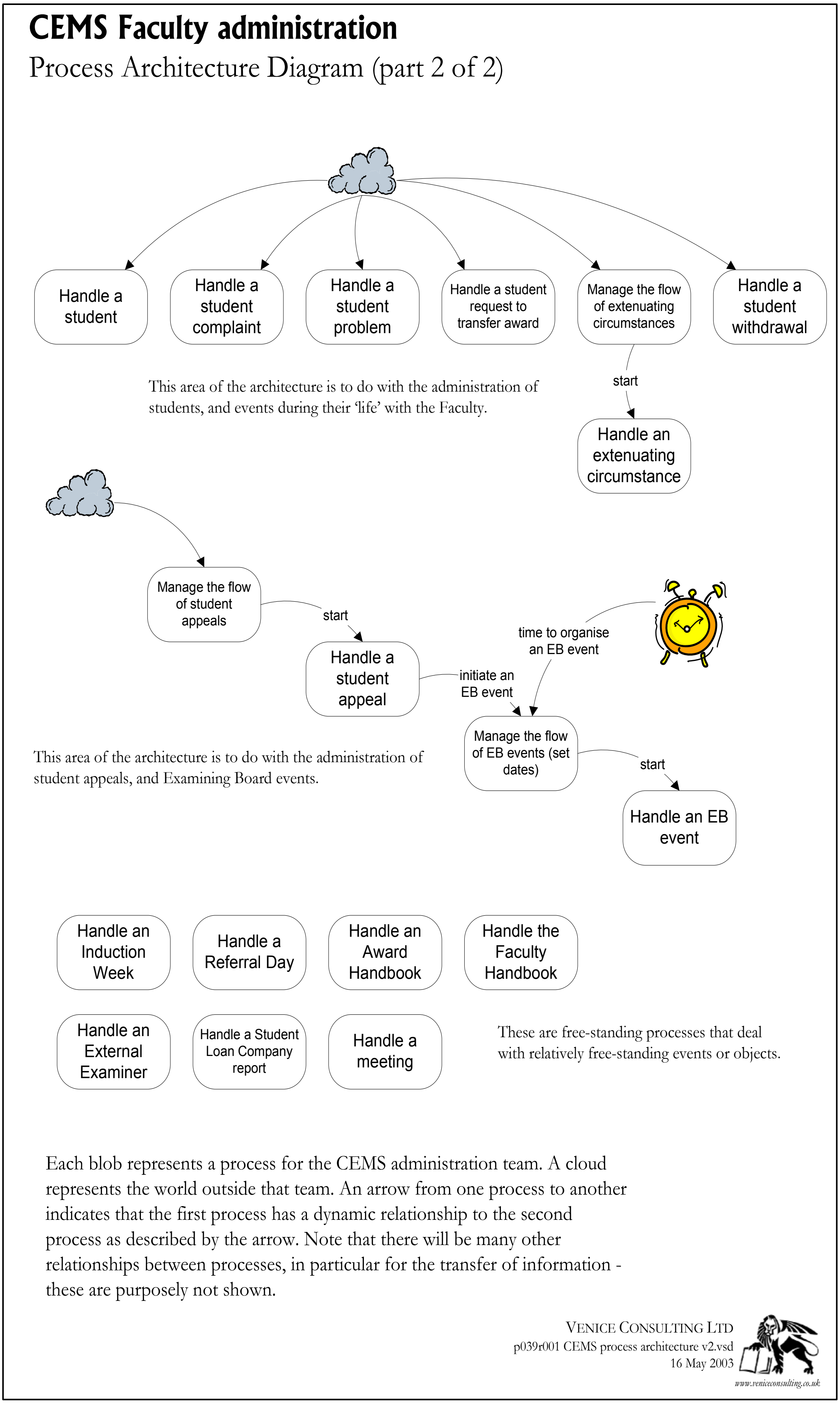


Figure A.5: The 2nd Cut Architecture for the CEMS Faculty Administration (Part 2 of 2) [Source: (Ould and Green, 2004), Used with the author's permission]

Appendix B: siGoal Ontology Classes and Associated Properties

This appendix presents siGoal ontology classes and associated attributes mentioned in Chapter 4 (Section 4.3.3.3). Table B.1 presents GO view concepts in a BSV as classes described with their associated properties.

Table B.4: The siGoal Ontology Concepts and Properties

Class	Description	Properties
CL	The canonical list of goals proposed by (Celements and Bass, 2010). Any business organisation must posses at least one of the goals in the CL as its highest business goal(s).	
BS_Model	The business strategy model, which is the first model required for the GO models within the business strategy view.	
Organisation	The agreed organisation for the business strategy view representation.	1- aimsForHBG of type HBG, 2- hasAimToRelation of type Aim_To and 3- belongsToBSModel only BS_Model.

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
HBG	The highest business goal that the agreed organisation aims to achieve. It is recognised within the goal network.	1- belongsToBSModel of type BS_Model, 2- belongsToGoalNetwork of type Goal_Network, 3- hasHBGIHGDecompositionRelation of type HBG_IHG_Goal_Decomposition, 4- hasSubIHG of type IH_G_Dependum, 5- matchesWithCL min 1 of type CL, and 6- isMatchedWithCL: Boolean.
Aim_To	The aim to relate to the BS model that sources from the organisation to its HBG(s). There is one aim to relation for each HBG.	1- hasHBGDestination of type HBG, 2- hasOrgSource of type Organisation and 3- belongsToBSModel of type BS_Model.
HSD_Model	The high strategic dependency, which is the second model required in the GO models within the business strategy view. It is considered as the first GO dependency model to appear in the interrelated GO models.	

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
HSD_Actor	The actor in HSD model. It is an active entity that holds intentions and/or abilities. A HSD actor in the agreed organisation is either a key role or a group of roles that collaborate to achieve a common goal in the HSD model.	1- hasIHGAbility of type IH_G_Dependum, 2- hasIHGIntention of type IH_G_Dependum, 3- hasIHSGAbility of type IH_SG_Dependum, 4- hasSubActor of type Actor, 5- isGroupOfActors: Boolean, 6- isKeyActor: Boolean, 7- hasIHSGIntention of type IH_SG_Dependum, 8- belongsToHSDModel only HSD_Model, and
IH_G_Dependum	The immediate highest sub goal dependum, which elaborate(s) a particular HBG parent. The IH-G is part of a dependency relation between the actors in the HSD model. It is recognised within the goal network	1- participatesInIHGDependency of type IH_G_Dependency, 2- belongsToHSDModel of type HSD_Model, 3- belongsToGoalNetwork of type Goal_Network, 4- hasIHGSDGDecompositionRelation of type IHG_SD_Goal_Decomposition and 5- hasSubSDGoal of type SD_Goal

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
IH_SG_Dependum	The immediate highest soft goal dependum, which constrain(s) an associated IH-G. The IH-SG participates as a part of a dependency relation between the actors in the HSD model	1- belongsToHSDModel of type HSD_Model, 2- constrainsIHG of type IH_G_Dependum, and
IH_G_Dependency	The goal dependency relation between two actors in the HSD model. The relation requires the participation of an IH-G dependum.	1- hasHSDDependeeDestination of type HSD_Actor, 2- hasHSDDependerSource of type HSD_Actor, 3- hasIHGDependum of type IH_G_Dependum and 4- belongsToHSDModel of type HSD_Model
IH_SG_Dependency	The soft goal dependency relation between two actors in the HSD model. The relation requires the participation of an IH-SG dependum.	1- hasHSDDependeeDestination of type HSD_Actor, 2- hasHSDDependerSource of type HSD_Actor, 3- hasIHSGDependum of type IH_SG_Dependum and 4- belongsToHSDModel of type HSD_Model.
SD_Model	The strategic dependency model in the *i** framework. It represents a network of goal and soft goal dependencies between the actors.	1- hasCorrespondingSRModel of type SR_Model.

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
SD_Goal	The main goal of the SD model. This goal represents the name of GQR-BPM. The SD goal is a goal itself and is recognised in the goal network.	1- belongsToGoalNetwork of type Goal_Network, 2- hasSubGoal of type Goal, 3- hasSDGDecompositionRelation of type SD_Goal_Decomposition, and 4- hasCorrespondingSDModel of type SD_Model.
Actor	An actor is the active entity that holds intentions or abilities to fulfill a dependum in the *i** framework.	1- belongsToSDModel of type SD_Model, 2- belongsToSRModel of type SR_Model 3- hasSubActor of type Actor, 4- hasGoalAbility of type Goal, 5- hasSoftGoalAbility of type Soft_Goal, 6- hasBoundary of type Actor_Boundary, 7- hasSoftGoalIntention of type Soft_Goal, 8- hasGoalIntention of type Goal, 9- addressesGoal of type Goal, 10- addressesSoftGoal of type Soft_Goal, 11- participatesInSoftGoalDependency Relation of type Soft_Goal_Dependency, and 12- participatesInGoalDependency of type Goal_Dependency.

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
Goal_Dependency	The goal dependency relation in the SD model between two actors, from a depender to a dependee.	1- belongsToSDModel of type SD_Model, 2- hasDependeeDestination of type Actor, 3- hasDependerSource of type Actor, and 4- hasGoalDependum of type Goal_Dependum.
Soft_Goal_Dependency	The soft goal dependency relation in the SD model between two actors, from a depender to a dependee.	1- belongsToSDModel of type SD_Model, and 2- hasDependeeDestination of type Actor, 3- hasDependerSource of type Actor, and 4- hasSoftGoalDependum of type Soft_Goal_Dependum.
Goal_Dependum	The goal dependum that participates in the goal dependency relation in the SD model. It is a condition or state in the *i** framework. It appears in both the SD and the SR models.	1- belongsToSDModel of type SD_Model, 2- belongsToSRModel of type SR_Model, and
Soft_Goal_Dependum	The soft goal dependum that participates in the soft goal dependency relation in the SD model.	1- belongsToSDModel of type SD_Model, 2- belongsToSRModel of type SR_Model, and

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
SR_Model	The strategic rationale model in the *i** framework. It represents the rationales beyond the SD's network of dependencies.	1- hasCorrespondingSDModel of type SD_Model.
Actor_Boundary	The actor boundary is the dotted area around the actor in the SR model. The boundary contains the SR components related to the actor of the boundary.	1- belongsToSRModel of type SR_Model, and 2- belongsToActorBoundary of type Actor_Boundary.
Goal	The goal component in the SR model within an actor boundary. It becomes a dependum if it participates in a dependency relationship in the SD model.	1- belongsToActorBoundary of type Actor_Boundary. 2- belongsToSRModel of type SR_Model. 3- isGoalDependum: Boolean, 4- hasSDGDecompositionRelation of type SD_Goal_Decomposition, 5- belongsToGoalNetwork of type Goal_Network, 6- belongsToSDModel of type SD_Model, 7- hasGGDecompositionRelation of type Goal_Goal_Decomposition, 8- hasTaskDecompositionRelationSubGoal of type Task_Decomposition_subGoal

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
Goal		9- hasSubGoal of type Goal, 10- participatesInGoalDependency of type Goal_Dependency, and 11- hasGTRelation of type GT_MeanEnd.
Soft_Goal	The goal component in the SR model within an actor boundary. It becomes a dependum if it participates in a dependency relationship.	1- belongsToActorBoundary of type Actor_Boundary, 2- belongsToSDModel of type SD_Model, 3- belongsToSRModel of type SR_Model, 4- participatesInSoftGoalDependencyRelation of type Soft_Goal_Dependency, 5- hasTaskDecompositionRelationSubSoftGoal of type Task_Decomposition_SoftGoalFor 6- constriansTask of type Task, 7- isSoftGoalDependum: Boolean, and 8- constrainsSDGoal of type SD_Goal
Resource	The resource component, that is physical or information, in the *i** framework.	1- hasTaskDecompositionRelationResourceFor of type Task_Decomposition_ResourceFor, 2- addressesTask of type Task, 3- hasRTRelation of type RT_MeanEnd, 4- belongsToSRModel of type SR_Model, and 5- belongsToActorBoundary of type Actor_Boundary.

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
Task	The task component, which is an operation or function in the i* framework.	1- belongsToSRModel of type SR_Model, 2- addressesGoal of type Goal, 3- addressesTask of type Task, 4- addressesSoftGoal of type Soft_Goal_Dependum, 5- hasSubTask of type Task, 6- hasSubResource of type Resource, 7- hasSubGoal of type Goal, 8- hasRTRelation of type RT_MeanEnd, 9- hasTaskDecompositionRelation SubGoal of type Task_Decomposition_subGoal 10- hasTaskDecompositionRelation ResourceFor of type Task_Decomposition_Resource For 11- hasTaskDecompositionRelation SubSoftGoal of type Task_Decomposition_SoftGoalF or 12- hasGTRelation of type GT_MeanEnd 13- hasTTRelation only TT_MeanEnd. 14- hasTaskDecompositionRelation SubTask of type Task_Decomposition_subTask, and 15- belongsToActorBoundary of type Actor_Boundary.

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
Task_Decomposition_subGoal	The task decomposition relationship, where the parent is a task and the goal is the sub part of the task.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasTaskSource of type Task, and 4- hasSubGoalDestination of type Goal_Dependum.
Task_Decomposition_subTask	The task decomposition relationship, where the parent is a task and the child is a sub task.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasTaskSource of type Task, and 4- hasSubTaskDestination of type Task.
Task_Decomposition_Resource For	The task decomposition relationship, where the parent is a task and the child is a resource as a sub part of the task.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasSubResourceDestination of type Resource, and 4- hasTaskSource of type Task.
Task_Decomposition_SoftGoal	The task decomposition relationship, where the parent is a task and the child is a soft goal for the task.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasTaskSource of type Task, and 4- hasSubSoftGoalDestination of type (Soft_Goal_Dependum or Soft_Goal).

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
TT_MeanEnd	This is the TT relationship in the SR model. In the TTLink relationship, the end and the mean are both tasks. The mean task fulfils the end task.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasTaskAsMean of type Task, and 4- hasTaskAsEnd of type Task.
GT_MeanEnd	In the GTLink relationship, the end is a goal and the mean is a task. The mean task fulfils the end goal.	1- belongsToSRModel of type SR_Model, 2- belongsToActorBoundary of type Actor_Boundary, 3- hasTaskAsMean of type Task, and 4- hasGoalAsEnd of type Goal.
Goal_Network	The goal network that represent the relation between the identified goals. The network comprises instances of the goal-related classes, that are HBG, IH_G_Dependum, SD_Goal and Goal, and the relations between them.	
HBG_IHG_Goal_Decomposition	The relation that represents the first-level between the goals in the goal network.	1- belongsToGoalNetwork of type Goal_Network, 2- hasHBGParent of type HBG, and 3- hasIHGOffspring only IH_G_Dependum

Table B.1 (Cont'd): The siGoal Ontology Concepts and Properties

Class	Description	Properties
IHG_SD_Goal_Decomposition	The relation that represents the second-level between the goals in the goal network.	1- belongsToGoalNetwork of type Goal_Network, 2- hasIHGParent of type IH_G_Dependum, 3- hasSDGoalOffspring of type SD_Goal
SD_Goal_Decomposition	The relation that represents the third-level between the goals in the goal network.	1- belongsToGoalNetwork of type Goal_Network, 2- hasGoalOffspring of type Goal, and 3- hasSDGoalParent of type SD_Goal
Goal_Goal_Decomposition	The relation that represents the fourth-level between the goals in the goal network.	1- belongsToGoalNetwork of type Goal_Network, 2- hasGoalOffspring of type Goal, and 3- hasGoalParent of type Goal.

In Table B.1, the **Organisation class** simply represents the agreed boundary of the organisation that is aiming to address a set of strategic objectives (i.e., HBGs) using the property *aimsforHBG*. The **HBG class** represents the highest business goals for an organisation where at least one of its instances should match or relate with an instance of the **CL class** (i.e., static class of the must possessing goals within an organisation) through the property "matchesWithCL min 1". This step requires the intervention of the analyst to decide if the identified HBGs for an organisation are a subset from the CL. Once the instances of the aforementioned classes are created and agreed, then an instance of the **BS_Model** is created in order to embody instances. In fact, the organisation must be related with the HBGs through the **Aim_To** relationship using the property *hasAimToRelation*. Accordingly, the Aim_To directed relation is characterised by its source (i.e., organisation) and its destination (i.e., HBGs) using the *hasOrgSource* and the *hasHBGDestination* properties respectively. The organisation and its related HBGs through the aim to relationships build the BS model using the *belongsToBSModel* property. The BS model is the first generated model in the GO part within the BSV.

The class **HSD_Actor** defines the actors that depend on each other's within the **HSD_Model** for a particular dependum. The dependum is either the immediate highest sub goal represented as the class **IH_G_Dependum** or the immediate highest soft goal represented in the class **IH_SG_Dependum**. The former dependum acts within its dependency relationship represented as the class **IH_G_Dependency** using the property *participatesInIHGDependency,* where the latter relates to the dependency relationship with class **IH_SG_Dependency** using the property *participatesInIHSGDependency*. The two dependencies are created after identifying the HSD Actor's IH-G and the IH-SG abilities and intentions, that requires the intervention of the analyst, using the object properties respectively: *hasIHGAbility, hasIHSGAbility, hasIHGIntention and hasIHGIntention*. Accordingly, the IH-G dependency relation is characterised using the object properties: *hasHSDDependeeDestination, hasHSDDependerSource* and *hasIHGDependum*. Similarly, the IH-SG dependency relation is characterised using the object properties: *hasHSDDependeeDestination, hasHSDDependerSource* and *hasIHSGDependum*. Hence, the HSD related classes constitute the HSD model for the GO part in the BSV via the property *belongsToHSDModel*. It necessary to highlight that IH-SG constrains associated relevant IH-G using the property *constrainsIHG*.

The **SD_Model** represents the third model that appears in siGoal ontology that is comprised of the classes **Actor, Goal_Dependency, Soft_Goal_Dependency, Goal_Dependum and Soft_Goal_Dependum**. The two dependums acts in their appropriate dependency relation respectively using the properties: *participatesInGoalDependency* and *participatesInSoftGoalDependency*. The actor in the SD model is identified using the object properties: *hasGoalAbility, hasGoalIntention, hasSoftGoalAbility* and *hasSoftGoalIntention*. Accordingly, the properties *hasDependeeDestination, hasDependerSource* and *hasGoalDependum* determine the class Goal_Dependency. The first and the second properties along with the property *hasSoftGoalDependum* are identified in order to set the class Soft_Goal_Dependency. The object property *belongsToSDModel* is defined in order to relate the class members (i.e., Actor, Goal_Dependency, Soft_Goal_Dependency, Goal_Dependum and Soft_Goal_Dependum) and the relations between them to the SD model. Although the notion of relating the classes in the SD model is similar to the work of the HSD model, the SD model differs from the HSD model in the abstraction level of the classes.

The SD model should have a corresponding SR model that elaborates the classes and the relations described above. Accordingly, the SD model related classes and relationships should exist for the SR model. Since those SD model-related classes and relationships have been already explained above, then it is desired to not explain them here again in order to shorten and ease the understanding of the SR-related classes and relationships that do not exist in the SD model (e.g., Actor_Boundary, Task, Resource, GT_MeanEnd, etc).

Since the *i** framework dependency relations in the proposed BSV are only in the form of goal and soft goal dependencies, their rationale and tactical elements address them within he SR model in away or another. Therefore, the classes **Actor_Boundary, Goal, Soft_Goal, Task, Resource**, and their relations presented in Table B.1 constitute the SR model. The SR model is interested in presenting the internal structure of the actor (i.e., represented using the class Actor) thereby; the internal boundary of the actor is represented using the class Actor_Boundary. Recalling that the actor boundaries are connected using the above description regarding the SD model.

The class **Goal_Network** binds some of the aforementioned classes in order to manifest the interrelationship between the four GO models. The actual binding is carried out from the goal concept point of view (i.e., HBG, IH-G, SD_Goal and Goal).

Appendix C: Ontologising Goal-Oriented View Using SWRL Rules

This appendix presents an implementation of the GO view with a BSV for an organisation. Table C.1 shows the steps proposed for designing GO view using SWRL rules. Each step is enriched with mentioning the related algorithm, related classes, associated classes' properties, class instances, related OWL restrictions and the designed SWRL rules.

Table C.1: The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

BS Model Step	**Step 1:** Identify the boundary of an agreed organisation and its associated ultimate goals, namely the Highest Business Goals (HBGs)
Related Algorithm	
Related Classes	**Organisation, HBG**
Class Properties	**aimsForHBG:** to set the ultimate goals (HBGs) required by the agreed organisation to address.
Class Instances	**Organisation** class is deigned to generate one instance that represents the agreed boundary of the organisation. The organisation instance should be identified and agreed by the stakeholders. In addition to the instances of the class HBG that represents the ultimate goals of the agreed organisation.
OWL Restrictions	-
SWRL Rules	-
BS Model Step	**Step 2:** Check whether at least one of the identified HBG instances in step 1 matches with an instance of the canonical list of goals in order to draw the BS model for the agreed organisation. This step is executed once.
Related Classes	**HBG, CL**
Class Properties	**isMatchedWithCL:** is set to True for a HBG instance matched with a particular instance of an ultimate goal in the agreed canonical list.
Class Instances	Instances of the class HBG represent the ultimate goals (highest business goals) for the agreed organisation. In this step, no new instances are created. This is because the already created HBG instances are checked here for their eligibility in the BS model, which is designed in the next step. If at least one of the created instances matches with the canonical list of goals, then the created HBG instances are considered eligible.
OWL Restrictions	**Cardinality restriction:** **HBG:** matchesWithCL **min 1** CL
SWRL Rules	**Rule_Investigating_HBG_in_CL:** CL(?cl) ^ isMatchedWithCL(?hbg, true) → HBG(?hbg).

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

BS Modelling Step	**Step 3:** Draw the BS model that represents the agreed organisation and its associated HBGs through the aim to relationships.
Related Classes	**Organisation, HBG, Aim_To, BS_Model**
Class Properties	**aimsForHBG:** to set the ultimate goals (HBGs) required by the agreed organisation to address. **hasAimToRelation**: Each agreed organisation instance should possess this property in order to explicitly denote to its HBGs aimed to address. **belongsToBSModel**: The generated instances for the BS model such as an instance of Organisation, instances of HBGs and their associated aiming-to relations should all belong to a particular instance BS model.
Class Instances	BS_Model instance, that represents the first goal-oriented model in the strategic view for an organisation, along with its related components' instances.
OWL Restrictions	**Organisation:** ∀ *belongsToBSModel only BS_Model* ∀ *aimsForHBG only HBG* ∀ *hasAimToRelation only Aim_To* **Aim_To:** ∀ *belongsToBSModel only BS_Model* ∀ *hasHBGDestination only HBG* ∀ *hasOrgSource only Organisation* **HBG:** ∀ *belongsToBSModel only BS_Model*
SWRL Rules	**Rule_Creating_BS_Model**: *Organisation(?org) ∧ aimsForHBG(?org, ?hbg) ∧ hasAimToRelation(?org, ?at) → HBG(?hbg) ∧ Aim_To(?at) ∧ hasOrgSource(?at, ?org) ∧ hasHBGDestination(?at, ?hbg) ∧ belongsToBSModel(?at, BS_Model) ∧ belongsToBSModel(?org, BS_Model) ∧ belongsToBSModel(?hbg, BS_Model)*
First Level of Goal Network Step	**Step 4:** Elaborate each HBG presented in the BS model into corresponding immediate highest sub goals and consider the elaboration as the first level of the goal network.
Related Classes	**HBG, IH_G_Dependum, Goal_Network, HBG_IHG_Goal_Decomposition.**
Class Properties	- **hasIHGOffspring**: to set the sub goals of the HBG identified in Step 3. -**hasHBGIHGDecompositionRelation**: Any HBG in the goal network that is decomposed into IHGs, will have this property. - **hasHBGParent and hasIHGOffspring**: Any decomposition relation in the first level of the goal network, will have these properties to determine the parent and the generated offsprings. -**belongsToGoalNetwork**: Any goal component in the goal network, will have this property.

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

<table>
<tr><td>Class Instances</td><td>Instances of the classes HBG, IH_G_Dependum and HBG_IHG_Goal_Decomposition. The instances are generated with their properties to identify the parent and the offspring of the decomposition relation. In particular, the first-level instance of the goal network is represented using the generated instances and the relationship between them.</td></tr>
<tr><td>OWL Restrictions</td><td>-HBG: ∀ hasSubIHG only IH_G_Dependum
∀ hasHBGIHGDecompositionRelation only HBG_IHG_Goal_Decomposition.
∀ belongsToGoalNetwork only Goal_Network

-IH_G_Dependum:
∀ belongsToGoalNetwork only Goal_Network

-HBG_IHG_Goal_Decomposition: ∀ hasHBGParent only HBG
∀ hasIHGOffspring only IH_G_Dependum
∀ belongsToGoalNetwork only Goal_Network</td></tr>
<tr><td>SWRL rules</td><td>Rule_Creating_First_Level_Goal_Network: hasSubIHG (?hbg, ?ihg) ∧ hasHBGIHGDecompositionRelation(?hbg, ?hidr) ∧ Goal_Network(?gnw) → HBG_IHG_Goal_Decomposition(?hidr) ∧ hasHBGParent(?hidr, ?hbg) ∧ hasIHGOffspring(?hidr, ?ihg) ∧ belongsToGoalNetwork(?hbg, ?gnw) ∧ belongsToGoalNetwork(?ihg, ?gnw) ∧ belongsToGoalNetwork(?hidr, ?gnw)</td></tr>
<tr><td>HSD Modelling Step</td><td>Step 5: Identify the high strategic actors along with their goal and soft goal abilities and intentions.

<table><tr><th></th><th>IH-G Abilities</th><th>IH-G Intentions</th><th>IH-SG Abilities</th><th>IH-SG Intentions</th></tr><tr><td>HSD Actor $_{x}$</td><td></td><td></td><td></td><td></td></tr><tr><td>HSD Actor $_{y}$</td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>Related Classes</td><td>HSD_Actor, IH_G_Dependum, IH_SG_Dependum.</td></tr>
<tr><td>Class Properties</td><td>isKeyActor: is a Boolean property that is set to true if the HSD actor is representing a key actor stemming from a key role.

isGroupOfActors: is Boolean type property set to "true" of the HSD actor is representing a set of actors.

hasIHGAbility and hasIHGIntention: to set the goal-based ability and the goal-based intention of the role and/or the HSD actor.

hasIHSGAbility and hasIHSGIntention: to set the soft goal-based ability and the soft goal-based intention of the role and/or the HSD actor.</td></tr>
</table>

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

Class Instances	Instances of the class HSD_Actor along with their properties with regard to the actor abilities and intentions.
OWL Restrictions	**HSD_Actor:** ∀ *hasIHGAbility only IH_G_Dependum* ∀ *hasIHGIntention only IH_G_Dependum* ∀ *hasIHSGAbility only IH_SG_Dependum.* ∀ *hasIHSGIntention only IH_SG_Dependum*
SWRL Rules	
HSD Modelling Step	**Step 6:** Draw the high strategic dependency (HSD) model that consists of the HSD actors, the immediate highest sub goal (IH-G) and soft goal (IH-SG) dependency relationships. This is the second goal-oriented model and as the first dependency model in the BSV.
Related Classes	**HSD_Model, HSD_Actor, IH_G_Dependum, IH_SG_Dependum, IH_G_Dependency, IH_SG_Dependency.**
Class Properties	**hasIHGAbility and hasIHGIntention:** to retrieve the goal-based ability and the goal-based intention of the HSD actor. **hasIHSGAbility and hasIHSGIntention:** to retrieve the soft goal-based ability and the soft goal-based intention of the HSD actor. **hasHSDDependeeDestination and hasHSDDependerSource:** is to set the depender and the dependee actors of the high strategic dependency relation that involves either a IH-G dependum or IH-SG depednum. **hasIHGDependum:** Any IH-G dependency relation should determine the participating IH-G dependum. **hasIHSGDependum:** Any IH-SG dependency relation should determine the participating IH-SG dependum. **belongsToHSDModel:** Any relevant component in a particular HSD model, will belong to the model.
Class Instances	HSD_Model instance that is interrelated with instances of the classes HSD_Actor, IH_G_Dependum, IH_SG_Dependum, IH_G_Dependency, IH_SG_Dependency.

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

OWL Restrictions	**HSD_Actor:** ∀ *hasIHGAbility only IH_G_Dependum* ∀ *hasIHGIntention only IH_G_Dependum* ∀ *hasIHSGAbility only IH_SG_Dependum* ∀ *hasIHSGIntention only IH_SG_Dependum* ∀ *belongsToHSDModel only HSD_Model* **IH_G_Dependum**: ∀ *belongsToHSDModel only HSD_Model* **IH_SG_Dependum**: ∀ *belongsToHSDModel only HSD_Model* **IH_G_Dependency**: ∀ *belongsToHSDModel only HSD_Model* ∀ *hasHSDDependeeDestination only HSD_Actor* ∀ *hasHSDDependerSource only HSD_Actor* ∀ *hasIHGDependum only IH_G_Dependum* **IH_SG_Dependency:** ∀ *belongsToHSDModel only HSD_Model* ∀ *hasHSDDependeeDestination only HSD_Actor* ∀ *hasHSDDependerSource only HSD_Actor* ∀ *hasIHSGDependum only IH_SG_Dependum*
SWRL Rules	**Rule_Creating_IHG_Dependency_Relation:** *HSD_Actor(?c) ∧ HSD_Actor(?e) ∧ hasIHGIntention(?c, ?ihg) ∧ hasIHGAbility(?e, ?ihg) ∧ HSD_Model(?hsdm) ∧ IH_G_Dependency(?ihgdr) → IH_G_Dependum(?ihg) ∧ hasIHGDependum(?ihgdr, ?ihg) ∧ hasHSDDependerSource(?ihgdr, ?c) ∧ hasHSDDependeeDestination(?ihgdr, ?e) ∧ belongsToHSDModel(?c, ?hsdm) ∧ belongsToHSDModel(?e, ?hsdm) ∧ belongsToHSDModel(?ihgdr, ?hsdm) ∧ belongsToHSDModel(?ihg, ?hsdm)* **Rule-Creating_IHSG_Dependency_Relation** *HSD_Actor(?c) ∧ HSD_Actor(?e) ∧ hasIHSGIntention(?c, ?ihsg) ∧ hasIHSGAbility(?e, ?ihsg) ∧ HSD_Model(?hsdm) ∧ IH_SG_Dependency(?ihsgdr) → IH_SG_Dependum(?ihsg) ∧ hasIHSGDependum(?ihsgdr, ?ihsg) ∧ hasHSDDependerSource(?ihsgdr, ?c) ∧ hasHSDDependeeDestination(?ihsgdr, ?e) ∧ belongsToHSDModel(?c, ?hsdm) ∧ belongsToHSDModel(?e, ?hsdm) ∧ belongsToHSDModel(?ihsgdr, ?hsdm) ∧ belongsToHSDModel(?ihsg, ?hsdm)*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

Second Level Goal Network Step	**Step 7:** Elaborate each IHG presented in the HSD model (in step 6) and in the goal network (in step 4) into corresponding sub-goals, that represent the goals of the SD models. The elaboration is considered as the second-level of the goal network.
Related Classes	**Goal_Network, IH_G_Dependum, SD_Goal, IHG_SD_Goal_Decomposition**
Class Properties	- **hasSubSDGoal**: the IH-G is elaborated into sub goals using this property. -**hasIHGSDGDecompositionRelation**: Any IHG in the goal network that is decomposed into sub goals will have this property. - **hasIHGParent and hasSDGoalOffspring**: Any decomposition relation in the second level of the goal network will have these properties to determine the parent and the generated offsprings. -**belongsToGoalNetwork**: Any component in the goal network will have this property.
Class Instances	Instances of the classes IH_G_Dependum, SD_Goal and IHG_SD_Goal_Decomposition. The instances are generated with their properties to determine the parent and the offspring of the IHG-SD decomposition relation. In addition, the second-level instance of the goal network is represented using the generated instances and the relationships between them.
OWL Restriction	**-IH_G_Dependum**: ∀ *hasSubSDGoal only SD_Goal* ∀ *hasIHGSDGDecompositionRelation only IHG_SD_Goal_Decomposition* ∀ *belongsToGoalNetwork only Goal_Network* **-SD_Goal:** ∀ *belongsToGoalNetwork only Goal_Network* **-IHG_SD_Goal_Decomposition**: ∀ *hasIHGParent only IH_G_Dependum* ∀ *hasSDGoalOffspring only SD_Goal* ∀ *belongsToGoalNetwork only Goal_Network*
SWRL Rules	**Rule-Creating_Second_Level_Goal_Network:** *IH_G_Dependum(?ihg) ∧ hasSubSDGoal (?ihg, ?sdg) ∧ hasIHGSDGDecompositionRelation(?ihg, ?isgdr) ∧ Goal_Network(?gnw) → SD_Goal(?sdg) ∧ IHG_SD_Goal_Decomposition(?isgdr) ∧ hasIHGParent(?isgdr, ?ihg) ∧ hasSDGoalOffspring(?isgdr, ?sdg) ∧ belongsToGoalNetwork(?ihg, ?gnw) ∧ belongsToGoalNetwork(?sdg, ?gnw) ∧ belongsToGoalNetwork(?isgdr, ?gnw)*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

<table>
<tr><td>SD Modelling Step</td><td>Step 8: Identify the SD actors' goal and soft goal abilities and intentions from the HSD actors identified in Step 5. The actor's goal and soft goal abilities and intentions are manually assigned. Some soft goals are automatically derived from the identified IH-SGs in step 5.

<table>
<tr><td></td><td>Goal Abilities</td><td>Goal Intentions</td><td>Soft Goal Abilities</td><td>Soft Goal Intentions</td></tr>
<tr><td>Actor $_x$</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Actor $_y$</td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Related Classes</td><td>HSD_Actor, IH_G_Dependum, IH_SG_Dependum, Actor, SD_Goal, Soft_Goal.</td></tr>
<tr><td>Class Properties</td><td>isKeyActor: is set to "true" for a HSD actor instance in the organisation in case it can represent a key actor.
isGroupOfActors: is set to "true" for a HSD actor instance in the HSD model in case it cannot represent a key actor and consists of a set of actors that have a common IH-G.

hasSubSDGoal: for a particular goal-based ability or intention of a HSD actor, there are sub SD goals that derive goal-based abilities and/or intentions using this property.

hasGoalAbility and hasGoalIntention: to set the goal-based ability and the goal-based intention of SD actor.

hasSoftGoalAbility and hasSoftGoalIntention: to set the soft goal-based ability and the soft goal-based intention of the SD actor.

constrainsIHG and contrainsSDGoal: to retrieve the IH-SG that constrains the IH-G and to identify the soft goal that constrains the goal of the SD model respectively.</td></tr>
<tr><td>Class Instances</td><td>Instances of the class Actor along with its related properties that are associated with the actor's goal abilities, intentions and soft goal abilities and intentions.</td></tr>
</table>

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

OWL Restrictions	**Actor:** ∀ *hasGoalAbility only Goal* ∀ *hasGoalIntention only Goal* ∀ *hasSoftGoalAbility only Soft_Goal* ∀ *hasSoftGoalIntention only Soft_Goal* **Soft_Goal:** ∀ *constrainsSDGoal only SD_Goal* **IH_G_Dependum:** ∀ *hasSubSDGoal only SD_Goal* **IH_SG_Dependum:** ∀ *constrainsIHG only IH_G_Dependum* **HSD_Actor***:* ∀ *hasSubSDActor only Actor.*
SWRL Rules	**Rule-Identifying_SD_Key_Actor:** *HSD_Actor(?hsda) ∧ isKeyActor(?hsda, true) → Actor(?hsda)* **Rule-Identifying_SD_Sub_Actors_From_HSD_Actor:** *HSD_Actor(?hsda) ∧ isGroupOfActors(?hsda, true) ∧ hasSubSDActor(?hsda, ?sda) → Actor(?sda)* **Rule-Deriving_Some_Corresponding_SoftGoals_From_IHSG:** *IH_SG_Dependum(?ihsg) ∧ constrainsIHG(?ihsg, ?ihg) ∧ SD_Goal(?sdg) ∧ hasSubSDGoal(?ihg, ?sdg) → Soft_Goal(?ihsg) ∧ constrainsSDGoal(?ihsg, ?sdg)*
Third and The Fourth Levels Goal Network Step	**Step 9**: Create the third-level in the goal network through elaborating the goal of the SD model into sub goals that are deduced in Step 8 in the form of goal abilities and intentions. In the same step, create the fourth level in the goal network through elaborating the goals into sub goals that some of them will participate in the dependency relations in step 10. This step produces all goals for the *i** framework models that are the SD and the SR models.
Related Classes	**SD_Goal, Goal, SDGoal_Goal_Decomposition, Goal_Goal_Decomposition, Goal_Network**
Class Properties	**- hasSDGoalParent:** to set the parent that is a goal of SD for a goal decomposition relation in the third level of the goal network. - **hasGoalOffspring**: to set the offspring goals of type Goal for goal decomposition relation in the third and fourth levels of the goal network. **-hasSDGDecompositionRelation**: this property is for goals that have sub goals in the third-level of the goal network **-hasGGDecompositionRelation:** this property is for goals that have sub goals in the fourth-level of the goal network. **-hasGoalParent:** to set up the parent of type goal in the Goal_Goal_Decomposition relation **-belongsToGoalNetwork**: Any component in the goal network, will have this property.

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

Class Instances	Instances of the classes that represent the third and the fourth levels in the goal network that are SD_Goal, Goal, SDGoal_Goal_Decomposition , Goal_Goal_Decomposition along with the properties between the classes that are related to the parents and off springs.
OWL Restrictions	**SD_Goal:** ∀ *hasSubGoal only Goal* ∀ *hasSDGDecompositionRelation only SD_Goal_ Decomposition* ∀ *belongsToGoalNetwork only Goal_Network* **Goal**: ∀ *hasSubGoal only Goal.* ∀ *belongsToGoalNetwork only Goal_Network* ∀ *hasGGDecompositionRelation only Goal_Goal_Decomposition* ∀ *hasSDGDecompositionRelation only SD_Goal_ Decomposition* **SD_Goal_ Decomposition:** ∀ *hasSDGoalParent only SD_Goal* ∀ *hasGoalOffspring only Goal* ∀ *belongsToGoalNetwork only Goal_Network* **Goal_Goal_Decomposition**: ∀*hasGoalParent only Goal.* ∀ *hasGoalOffspring only Goal* ∀ *belongsToGoalNetwork only Goal_Network*
SWRL Rules	**Rule-Creating_Third_Level_Goal_Network** *SD_Goal(?sdg) ⋀ Goal(?g) ⋀ SD_Goal_Decomposition(?sdgr) ⋀ hasSubGoal(?sdg, ?g) ⋀ hasSDGDecompositionRelation(?sgd, ?sdgr) ⋀ hasSDGDecompositionRelation(?g, ?sdgr) ⋀ Goal_Network(?gnw) → hasSDGoalParent(?sdgr, ?sdg) ⋀ hasGoalOffspring(?sdgr, ?g) ⋀ belongsToGoalNetwork(?sdg, ?gnw) ⋀ belongsToGoalNetwork(?g, ?gnw) ⋀ belongsToGoalNetwork(?sdgr, ?gnw)* **Rule-Creating_Fourth_Level_Goal_Network** *Goal(?ga) ⋀ Goal(?gb) ⋀ Goal_Goal_Decomposition(?ggdr) ⋀ hasSubGoal(?ga, ?gb) ⋀ hasGGDecompositionRelation(?ga, ?ggdr) ⋀ hasGGDecompositionRelation(?gb, ?ggdr) ⋀ Goal_Network(?gnw) → hasGoalParent(?ggdr, ?ga) ⋀ hasGoalOffspring(?ggdr, ?gb) ⋀ belongsToGoalNetwork(?ggdr, ?gnw) ⋀ belongsToGoalNetwork(?ga, ?gnw) ⋀ belongsToGoalNetwork(?gb, ?gnw)*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SD Modelling Step	**Step 10:** Draw the SD model that comprises of actors, goal dependencies and soft goal dependencies. This step is aided with the work in step 8.
Related Classes	**SD_Model, SD_Goal, Actor, Goal_Dependum, Soft_Goal_Depednum, Goal_Dependency, Soft_Goal_Dependency,**
Class Properties	**-isGoalDependum** and **isSoftGoalDependum:** these two Boolean properties are used to decide whether a goal and soft goal are dependums respectively. **- hasGoalIntention**: this property is related with the class Goal in order to retrieve the goal-based intentions, which have been already identified in step 8, for an actor in the SD model. **- hasSoftGoalIntention:** this properties is related with the class Soft_Goal in order to retrieve the soft goal-based intentions, which have been already identified in step 8, for an actor in the SD model. **-participatesInGoalDependency**: actors that has a goal intention or it is capable addressing a goal should have this property in order to create the goal dependency relation from the depender to the dependee. **-addressesGoal** and **addressesSoftGoal**: to match actors' abilities with goal-based intentions and soft goal-based intentions in one SD model in order to create goal and soft goal dependency relationships. **-participatesInSoftGoalDependency**: actors should use this property in order to create soft goal dependency relation from the depender to the dependee. **-hasGoalDependum**: this property involves a goal dependum instance in order to determine the dependency relation as a goal-oriented dependency relation. This happens similarly with the **hasSoftGoalDependum** property, yet with regard to the involvement of the soft goal dependum in order to draw a soft goal dependency relationship. **-hasDependerSource** and **hasDependeeDestination**: these two properties determine the source and the destination actors of a dependency relationship in the SD model. **- belongsToSDModel**: any instance of a related class with the SD model, will have this property to generate the SD model.
Class Instances	An instance of the SD model that is related through the properties with other instances that set up the SD model such as instances of the classes Actor, SD_Goal, Goal_Dependum, Soft_Goal_Dependum, Goal_Dependency and Soft_Goal_Dependency.

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

OWL Restrictions	**Actor:** ∀ *addressesGoal only Goal* ∀ *hasGoalIntention only Goal* ∀ *addressesSoftGoal only Soft_Goal* ∀ *hasSoftGoalIntention only Soft_Goal* ∀ *belongsToSDModel only SD_Model* ∀*participatesInGoalDependencyRelation only Goal_Dependency.* ∀*participatesInSoftGoalDependencyRelation only Soft_Goal_Dependency* **Goal_Dependum:** ∀ *belongsToSDModel only SD_Model* **Soft_Goal_Dependum:** ∀ *belongsToSDModel only SD_Model* **Goal_Dependency:** ∀ *hasDependerSource only Actor* ∀ *hasDependeeDestination only Actor* ∀ *hasGoalDependum only Goal_Dependum* ∀ *belongsToSDModel only SD_Model* **Soft_Goal_Dependency:** ∀ *hasDependerSource only Actor* ∀ *hasDependeeDestination only Actor* ∀ *hasSoftGoalDependum only Soft_Goal_Dependum* ∀ *belongsToSDModel only SD_Model*
SWRL Rules	**Rule-Corresponding_SD_Model** *SD_Goal(?sdg) ∧ hasCorrespondingSDModel(?sdg, ?sdm) → SD_Model(?sdm)* **Rule_Creating_SD_Goal_Dependency** *SD_Model(?sdm) ∧ Actor(?a) ∧ Actor(?b) ∧ Goal(?g) ∧ isGoalDependum(?g, true) ∧ Goal_Dependency(?gdr) ∧ participatesInGoalDependencyRelation(?a, ?gdr) ∧ participatesInGoalDependencyRelation(?b, ?gdr) ∧ participatesInGoalDependencyRelation(?g, ?gdr) ∧ hasGoalIntention(?a, ?g) ∧ addressesGoal(?b, ?g) ∧ belongsToSDModel(?a, ?sdm) ∧ belongsToSDModel(?b, ?sdm) ∧ belongsToSDModel(?g, ?sdm) ∧ belongsToSDModel(?gdr, ?sdm) → Goal_Dependum(?g) ∧ hasGoalDependum(?gdr, ?g) ∧ hasDependerSource(?gdr, ?a) ∧ hasDependeeDestination(?gdr, ?b) ∧ belongsToSDModel(?g, ?sdm)* **Rule_Creating_SD_SoftGoal_Dependency** *SD_Model(?sdm) ∧ Actor(?a) ∧ Actor(?b) ∧ Soft_Goal(?sg) ∧ isSoftGoalDependum(?sg, true) ∧ Soft_Goal_Dependency(?sgdr) ∧ participatesInSoftGoalDependencyRelation(?a, ?sgdr) ∧ participatesInSoftGoalDependencyRelation(?b, ?sgdr) ∧ participatesInSoftGoalDependencyRelation(?g, ?sgdr) ∧ hasSoftGoalIntention(?a, ?sg) ∧ addressesSoftGoal(?b, ?sg) ∧ belongsToSDModel(?a, ?sdm) ∧ belongsToSDModel(?b, ?sdm) ∧ belongsToSDModel(?sg, ?sdm) ∧ belongsToSDModel(?sgdr, ?sdm) → Soft_Goal_Dependum(?sg) ∧ hasSoftGoalDependum(?sgdr, ?sg) ∧ hasDependerSource(?sgdr, ?a) ∧ hasDependeeDestination(?sgdr, ?b) ∧ belongsToSDModel(?sg, ?sdm)*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SR Modelling Step	**Step 11:** Create a corresponding SR model for each SD model identified.
Related Classes	**SD_Model, SR_Model.**
Class Properties	**-hasCorrespondingSRModel**: to relate each SD model to at least one SR model. This property is not functional as one SD model instance is permitted to relate to at least one SR_Model instance using this property.
Class Instances	SR model instance that elaborate a corresponding SD model.
OWL Restrictions	**SD_Model**: ∀ *hasCorrespondingSRModel only SR_Model*
SWRL Rules	**Rule_Corresponding_SR_Model**: *SD_Model(?sdm) ∧ hasCorrespondingSRModel(?sdm, ?srm) → SR_Model(?srm)*
SR Modelling Step	**Step 12**: Create an actor boundary for each corresponding actor identified in the SD model (in step 8).
Related Classes	**SD_Model, SR_Model, Actor, Actor_Boundary**
Class Properties	**-hasBoundary:** by using this property, an actor in the SD model is shaped with a boundary in the SR model. This boundary comprises of the actor's SR related components. **-belongsToSRModel:** any SR model related component will belong to the SR model. A boundary of an actor must have this property as well. **-hasSubActor:** an actor may have a sub actor within the former actor boundary that is determined using this relation. **-belongsToActorBoundary**: GO components and relations that exist within an actor boundary should have this property in order to determine their boundary of usage under a particular actor. Also, an actor boundary that is sub of its parent actor boundary can use this object property. **-belongsToSDModel, hasCorrespondingSRModel.**
Class Instances	Actor_boundary instances that will be described by the SR modelling components.
OWL Restrictions	**SD_Model**: ∀ *hasCorrespondingSRModel only SR_Model* **Actor**: ∀ *hasBoundary only Actor_Boundary* ∀ *hasSubActor only Actor.* ∀ *belongsToSRModel only SR_Model* ∀ *belongsToSDModel only SD_Model* **Actor_Boundary**: ∀ *belongsToSRModel only SR_Model* ∀ *belongsToActorBoundary only Actor_Boundary.*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SWRL Rules	**Rule-Creating_Actor_Boundary:** *Actor(?x) ∧ hasBoundary(?x, ?bdr) ∧ belongsToSDModel(?x, ?sdm) ∧ SR_Model(?srm) ∧ hasCorrespondingSRModel(?sdm, ?srm) ∧ belongsToSRModel(?bdr, ?srm)→Actor_Boundary(?bdr) ∧ belongsToSRModel(?x, ?srm).* **Rule-Creating_Sub_Actor_Boundary: (if needed)** *Actor(?x1) ∧ hasBoundary(?x2, ?bdr2) ∧ hasSubActor(?x1, ?x2) ∧ hasBoundary(?x1, ?bdr1) ∧ belongsToSRModel(?bdr1, ?srm)→Actor_Boundary(?bdr2) ∧ Actor(?x2) ∧ belongsToActorBoundary(?bdr2, ?bdr1) ∧ belongsToActorBoundary(?x2, ?bdr1) ∧ belongsToSRModel(?bdr2, ?srm)*
SR Modelling Step	**Step 13:** Allocate each identified goal that is an ability of a particular actor to the actor's boundary and create the mean task relation that satisfies the end goal ability.
Related Classes	**Task, Goal, Actor_Boundary, GT_MeanEnd. SR_Model**
Class Properties	**hasGTRelation**: to bind a task mean instance to task mean goal end relation in the actor boundary in the SR model. **addressesGoal:** a task is involved in this property in order to determine which goal it will satisfy. **hasGoalAsEnd** and **hasTaskAsMean:** this is to set up the end goal and the mean task for the GT relation respectively. **belongsToActorBoundary:** tasks, goals, actor, and the GT mean end relation do all belong to the actor's boundary through using this object property. **hasGoalAbility, hasBoundary, belongsToSRModel.-** already explained above.
Class Instances	A task-mean goal-end relationship along with its assigned goal and task.
OWL Restrictions	**Task:** ∀ *hasGTRelation only GT_MeanEnd* ∀ *belongsToActorBoundary only Actor_Boundary* ∀ *belongsToSRModel only SR_Model* ∀ *addressesGoal only Goal* **Goal:** ∀ *belongsToActorBoundary only Actor_Boundary* ∀ *belongsToSRModel only SR_Model* ∀ *hasGTRelation only GT_MeanEnd* **Actor:** ∀ hasGoalAbility only Goal ∀ hasBoundary only Actor_Boundary. **Actor_Boundary:** ∀ *belongsToSRModel only SR_Model* **GT_MeanEnd:** ∀ *belongsToActorBoundary only Actor_Boundary* ∀ *belongsToSRModel only SR_Model* ∀ *hasGoalAsEnd only Goal* ∀ *hasTaskAsMean only Task*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SWRL Rules	**Rule_Creating_GT_Relation:** *Goal(?g) ⋀ Actor(?a) ⋀ Actor_Boundary(?bdr) ⋀ SR_Model(?srm) ⋀ hasGTRelation(?t, ?gtr) ⋀ hasGTRelation(?g, ?gtr) ⋀ hasGoalAbility(?a, ?g) ⋀ hasBoundary(?a, ?bdr) ⋀ belongsToSRModel(?a, ?srm) ⋀ belongsToSRModel(?bdr, ?srm) ⋀ belongsToActorBoundary(?g, ?bdr) ⋀ addressesGoal(?t, ?g) → Task(?t) ⋀ GT_MeanEnd(?gtr) ⋀ hasTaskAsMean(?gtr, ?t) ⋀ hasGoalAsEnd(?gtr, ?g) ⋀ belongsToActorBoundary(?g, ?bdr) ⋀ belongsToActorBoundary(?t, ?bdr) ⋀ belongsToActorBoundary(?gtr, ?bdr) ⋀ belongsToSRModel(?g, ?srm) ⋀ belongsToSRModel(?t, ?srm) ⋀ belongsToSRModel(?gtr, ?srm).*
SR Modelling Step	**Step 14:** Allocate the soft goal abilities, which have been already identified in step 8, of an actor to its boundary without constructing SIG as this will be in the part of the sQuality ontology in Section 4.3.4.
Related Classes	**Actor, Actor_Boundary, Soft_Goal, SR_Model**
Class properties	**hasSoftGoalAbility, hasBoundary, belongsToActorBoundary, belongsToSRModel,**
Class Instances	Instances of the class soft goal that represent a soft goal ability for a particular actor in the SR model considering that the soft goal abilities have been already identified in Step 8.
OWL restrictions	**Soft_Goal:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* **Actor:** ∀ hasSoftGoalAbility only Soft_Goal ∀ hasBoundary only Actor_Boundary. **Actor_Boundary**: ∀ *belongsToSRModel only SR_Model*
SWRL rules	**Rule-Allocating_Soft_Goal_Ability_in_Actor_Boundary:** *Actor(?a) ⋀ Soft_Goal(?sg) ⋀ hasSoftGoalAbility(?a, ?sg) ⋀ Actor_Boundary(?b) ⋀ hasBoundary(?a, ?b) ⋀ SR_Model(?srm) ⋀ belongsToSRModel(?b, ?srm) ⋀ belongsToSRModel(?a, ?srm) → belongsToActorBoundary(?sg, ?b) ⋀ belongsToSRModel(?sg, ?srm)*
SR Modelling Step	**Step 15:** Decompose a task into sub tasks or resources in the actor boundary.
Related Classes	**Task, Actor_Boundary, Actor, SR_Model, Resource, Task_Decomposition_subTask, Task_Decomposition_ResourceFor**

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

Class properties	**hasTaskSource, hasSubTaskDestination** and **hasSubResourceDestination:** the task decomposition relation that shows a decomposition of a task into sub tasks or into sub resources will use these properties in order to determine the task source that is decomposed into sub task or resource as destinations. **hasTaskDecompositionRelationSubTask:** a decomposable task in the actor boundary must have task decomposition relation. Also, the sub task should have this property to be related to the decomposable task. This task decomposition relation decomposes a task into a sub task. **hasTaskDecompositionRelationResourceFor**: this property relates a decomposable task and its sub resource. This task decomposition relationship decomposes a task into a sub resource. **belongsToActorBoundary, belongsToSRModel**
Class Instances	Instances of the classes Task_Decomposition_subTask along with its properties that are related to its source and destination tasks. In addition to instances of the class Task_Decomposition_ResourceFor along with its properties of task source and resource destination.
OWL restrictions	**Task:** ∀ *hasTaskDecompositionRelationSubTask only Task_Decomposition_subTask.* ∀ *hasTaskDecompositionRelationResourceFor only Task_Decomposition_ResourceFor* ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* **Task_Decomposition_subTask:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasSubTaskDestination only Task* ∀ *hasTaskSource only Task* **Resource:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ hasTaskDecompositionRelationResourceFor only Task_Decomposition_ResourceFor **Task_Decomposition_ResourceFor:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasSubResourceDestination only Resource* ∀ *hasTaskSource only Task*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SWRL rules	**Rule_Creating_Task_Decomposition_subTask_Relation:** *Task(?t1) ⋀ Task(?t2) ⋀ SR_Model(?srm) ⋀ Actor_Boundary(?bdr) ⋀ belongsToActorBoundary(?t1, ?bdr) ⋀ hasSubtask(?t1, ?t2) ⋀ hasTaskDecompositionRelationSubTask(?t1, ?tdtr) ⋀ hasTaskDecompositionRelationSubTask(?t2, ?tdtr) ⋀ belongsToSRModel(?bdr, ?srm) ⋀ belongsToSRModel(?t1, ?srm) → Task_Decomposition_subTask(?tdtr) ⋀ hasTaskSource(?tdtr, ?t1) ⋀ hasSubTaskDestination(?tdtr, ?t2) ⋀ belongsToActorBoundary(?t2, ?bdr) ⋀ belongsToActorBoundary(?tdtr, ?bdr) ⋀ belongsToSRModel(?t2, ?srm) ⋀ belongsToSRModel(?tdtr, ?srm)* ***Rule_Creating_Task_Decomposition_ResourceFor_Relation:*** *Task(?t) ⋀ Resource(?r) ⋀ hasSubResource(?t, ?r) ⋀ SR_Model(?srm) ⋀ Actor_Boundary(?bdr) ⋀ belongsToActorBoundary(?r, ?bdr) ⋀ belongsToSRModel(?bdr, ?srm) ⋀ hasTaskDecompositionRelationResourceFor(?t, ?tdrr) ⋀ hasTaskDecompositionRelationResourceFor(?r, ?tdrr) ⋀ belongsToSRModel(?t, ?srm) → Task_Decomposition_ResourceFor(?tdrr) ⋀ hasTaskSource(?tdrr, ?t) ⋀ hasSubResourceDestination(?tdrr, ?r) ⋀ belongsToActorBoundary(?t, ?bdr) ⋀ belongsToActorBoundary(?tdrr, ?bdr) ⋀ belongsToSRModel(?r, ?srm) ⋀ belongsToSRModel(?tdrr, ?srm)*
SR Modelling Step	**Step 16:** Creating TT-link (mean task, end task) and RT-link (mean task, end resource) relations.
Related Classes	**Task, Resource, SR_Model, Actor_Boundary, TT_MeanEnd, RT_MeanEnd**
Class properties	**hasTaskAsEnd:** this property is to make the end of a TT_Relation as a task. **hasResourceAsEnd:** this property is to set the end of an RT_Relation into a resource. **hasTTRelation** and **hasRTRelation:** a task can have TT-Relation or RT- Relation based on the type of the end. A resource is also involved in the **hasRTRelation**. **addressesTask**: a task or a resource that has hasTTRelation or RTRelation should be related to a task using this property. **hasTaskAsMean, belongsToActorBoundary, belongsToSRModel**

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

Class Instances	The TT link instances and the RT link instances along with their mean instances (i.e., tasks) and the their end instances (i.e., task and resource respectively) that are both determined using the appropriate object properties.
OWL restrictions	**Task:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasTTRelation only TT_MeanEnd* ∀ *hasRTRelation only RT_MeanEnd* ∀ addressesTask only Task **TT_MeanEnd**: ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasTaskAsMean only Task* ∀ *hasTaskAsEnd only Task* **Resource:** ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasRTRelation only RT_MeanEnd* ∀ addressesTask only Task **RT_MeanEnd**: ∀ *belongsToActorBoundary only Actor_Boundary.* ∀ *belongsToSRModel only SR_Model* ∀ *hasResourceAsEnd only Resource* ∀ *hasTaskAsMean only Task*
SWRL rules	**Rule_Creating_TT_Relation:** *Task(?t1) ∧ Task(?t2) ∧ addressesTask(?t2, ?t1) ∧ hasTTRelation(?t1, ?ttr) ∧ hasTTRelation(?t2, ?ttr) ∧ Actor_Boundary(?bdr) ∧ SR_Model(?srm) ∧ belongsToActorBoundary(?t1, ?bdr) ∧ belongsToSRModel(?bdr, ?srm) → TT_MeanEnd(?ttr) ∧ hasTaskAsMean(?ttr, ?t2) ∧ hasTaskAsEnd(?ttr, ?t1) ∧ belongsToActorBoundary(?t2, ?bdr) ∧ belongsToActorBoundary(?ttr, ?bdr) ∧ belongsToSRModel(?t2, ?srm) ∧ belongsToSRModel(?ttr, ?srm)* **Rule_Creating_RT_Relation:** *Task(?t) ∧ Resource(?r) ∧ hasSubResource(?t, ?r) ∧ SR_Model(?srm) ∧ Actor_Boundary(?bdr) ∧ belongsToActorBoundary(?r, ?bdr) ∧ belongsToSRModel(?bdr, ?srm) ∧ hasTaskDecompositionRelationResourceFor(?t, ?tdrr) ∧ hasTaskDecompositionRelationResourceFor(?r, ?tdrr) ∧ belongsToSRModel(?t, ?srm) → Task_Decomposition_ResourceFor(?tdrr) ∧ hasTaskSource(?tdrr, ?t) ∧ hasSubResourceDestination(?tdrr, ?r) ∧ belongsToActorBoundary(?t, ?bdr) ∧ belongsToActorBoundary(?tdrr, ?bdr) ∧ belongsToSRModel(?r, ?srm) ∧ belongsToSRModel(?tdrr, ?srm)*

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SR Modelling Step	**Step 17:** Decompose a task into sub goals and soft goals in the actor boundary. If any of the goals or soft goals is an intention for the actor, then it must participate in a dependency relation where the actor is the depender and the dependum is the goal or the soft goal.
Related Classes	**Task, Goal, Soft_Goal, SR_Model, Actor_Boundary, Task_Decomposition_subGoal, Task_Decomposition_SoftGoalFor**
Class properties	**-hasSubGoalDestination:** a goal and a task for task decomposition relation will have a goal as a destination (i.e., the goal is sub part of the task). **-hasSubGoal**: this property is to set goals that are part of a task. **-hasSubSoftGoalDestination:** a soft goal for task decomposition relation will have a soft goal as destination (i.e., the goal is sub part of the task). **-hasTaskDecompositionRelationSubGoal:** if a goal is sub of a task, then this property relates the task and goal with a relation of type Task_Decomposition_subGoal **-hasTaskDecompositionRelationSubSoftGoal:** if a soft goal is sub of a task, then this property relates the task and soft goal with a relation of type Task_Decomposition_SoftGoalFor **-addressesGoal** and **addressesSoftGoal**: to determine the tasks that are capable to satisfy a goal and soft goal dependums respectively. **-contrainsTask**: a soft goal that is sub from a task is treated as a constraint for this task using this property. **-hasTaskSource , belongsToActorBoundary, hasBoundary, belongsToSRModel, belongsToSDModel**
Class Instances	The task decomposition into sub goal relation instances along with their related instances of tasks and goals. Similarly, the task decomposition into sub soft goal relation instances along with their related instances of tasks and soft goals.

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

OWL restrictions	**Actor:** ∀ addressesGoal only Goal ∀ hasGoalIntention only Goal ∀ hasBoundary only Actor_Boundary **Goal:** ∀ belongsToActorBoundary only Actor_Boundary. ∀ belongsToSRModel only SR_Model ∀ hasTaskDecompositionRelationSubGoal only Task_Decomposition_subGoal **Soft_Goal:** ∀ belongsToActorBoundary only Actor_Boundary. ∀ belongsToSRModel only SR_Model ∀ constrainsTask only Task ∀ hasTaskDecompositionRelationSubSoftGoal only Task_Decomposition_SoftGoalFor **Task:** ∀ belongsToActorBoundary only Actor_Boundary. ∀ belongsToSRModel only SR_Model ∀ hasTaskDecompositionRelationSubGoal only Task_Decomposition_subGoal ∀ hasTaskDecompositionRelationSubSoftGoal only Task_Decomposition_SoftGoalFor ∀ addressesGoal only (Goal_Dependum or Goal) ∀ addressesSoftGoal only (Soft_Goal_Dependum or Soft_Goal) **Task_Decomposition_subGoal** ∀ belongsToActorBoundary only Actor_Boundary. ∀ belongsToSRModel only SR_Model ∀ hasTaskSource only Task ∀ hasSubGoalDestination only Goal_Dependum **Task_Decomposition_SoftGoalFor:** ∀ belongsToActorBoundary only Actor_Boundary. ∀ belongsToSRModel only SR_Model ∀ hasTaskSource only Task ∀ hasSubSoftGoalDestination only (Soft_Goal_Dependum or Soft_Goal).

Table C.1 (Cont'd): The Interrelated Goal-Oriented Models Proposed Steps Using OWL-DL

SWRL rules	**Rule_Creating_Task_Decomposition_subGoal_Relation**: *Goal(?g) ∧ Task(?t) ∧ hasSubGoal(?t, ?g) ∧ hasTaskDecompositionRelationSubGoal(?t, ?tdgr) ∧ hasTaskDecompositionRelationSubGoal(?g, ?tdgr) ∧ SR_Model(?srm) ∧ Actor_Boundary(?bdr) ∧ belongsToSRModel(?bdr, ?srm) ∧ belongsToActorBoundary(?t, ?bdr) ∧ belongsToActorBoundary(?g, ?bdr) → Task_Decomposition_subGoal(?tdgr) ∧ hasSubGoalDestination(?tdgr, ?g) ∧ hasTaskSource(?tdgr, ?t) ∧ belongsToActorBoundary(?tdgr, ?bdr) ∧ belongsToSRModel(?g, ?srm) ∧ belongsToSRModel(?tdgr, ?srm)* **Rule_Creating_Task_Decomposition_subSoftGoal_Relation**: *Soft_Goal(?sg) ∧ Task(?t) ∧ constrainsTask(?sg, ?t) ∧ hasTaskDecompositionRelationSubSoftGoal(?t, ?tdsgr) ∧ hasTaskDecompositionRelationSubSoftGoal(?sg, ?tdsgr) ∧ SR_Model(?srm) ∧ Actor_Boundary(?bdr) ∧ belongsToSRModel(?bdr, ?srm) ∧ belongsToActorBoundary(?t, ?bdr) ∧ belongsToActorBoundary(?sg, ?bdr) → Task_Decomposition_SoftGoalFor(?tdsgr) ∧ hasSubSoftGoalDestination(?tdsgr, ?sg) ∧ hasTaskSource(?tdsgr, ?t) ∧ belongsToActorBoundary(?tdsgr, ?bdr) ∧ belongsToSRModel(?sg, ?srm) ∧ belongsToSRModel(?tdsgr, ?srm)* **Rule-Addressing_Goal_Depednum_Using_A_Task:** *Actor(?a) ∧ Actor(?b) ∧ hasBoundary(?a, ?bdr1) ∧ hasBoundary(?b, ?bdr2) ∧ SR_Model(?srm) ∧ belongsToSRModel(?bdr1, ?srm) ∧ belongsToSRModel(?bdr2, ?srm) ∧ Goal(?g) ∧ Task(?t1) ∧ Task(?t2) ∧ hasGoalIntention(?a, ?g) ∧ addressesGoal(?b, ?g) ∧ belongsToActorBoundary(?t1, ?bdr1) ∧ belongsToActorBoundary(?t2, ?bdr2) ∧ hasTaskDecompositionRelationSubGoal(?t1, ?tdgr1) ∧ hasTaskDecompositionRelationSubGoal(?g, ?tdgr1) ∧ hasTaskDecompositionRelationSubGoal(?t2, ?tdgr2) ∧ hasTaskDecompositionRelationSubGoal(?g, ?tdgr2) → Task_Decomposition_subGoal(?tdgr1) ∧ Task_Decomposition_subGoal(?tdgr2) ∧ hasTaskSource(?tdgr1, ?t1) ∧ hasTaskSource(?tdgr2, ?t2) ∧ hasSubGoalDestination(?tdgr1, ?g) ∧ hasSubGoalDestination(?tdgr2, ?g) ∧ belongsToSRModel(?tdgr1, ?srm) ∧ belongsToSRModel(?tdgr2, ?srm) ∧*

	belongsToSRModel(?g, ?srm) **Rule-Addressing_Soft_Goal_Depednum_Using_A_Task:** *Actor(?a) ∧ Actor(?b) ∧ hasBoundary(?a, ?bdr1) ∧ hasBoundary(?b, ?bdr2) ∧ SR_Model(?srm) ∧ belongsToSRModel(?bdr1, ?srm) ∧ belongsToSRModel(?bdr2, ?srm) ∧ Soft_Goal(?sg) ∧ Task(?t1) ∧ Task(?t2) ∧ hasSoftGoalIntention(?a, ?sg) ∧ addressesSoftGoal(?b, ?sg) ∧ belongsToActorBoundary(?t1, ?bdr1) ∧ belongsToActorBoundary(?t2, ?bdr2) ∧ hasTaskDecompositionRelationSubSoftGoal(?t1, ?tdsgr1) ∧ hasTaskDecompositionRelationSubSoftGoal(?sg, ?tdsgr1) ∧ hasTaskDecompositionRelationSubSoftGoal(?t2, ?tdsgr2) ∧ hasTaskDecompositionRelationSubSoftGoal(?sg, ?tdsgr2) → Task_Decomposition_SoftGoalFor(?tdsgr1) ∧ Task_Decomposition_SoftGoalFor(?tdsgr2) ∧ hasTaskSource(?tdsgr1, ?t1) ∧ hasTaskSource(?tdsgr2, ?t2) ∧ hasSubSoftGoalDestination(?tdsgr1, ?sg) ∧ hasSubSoftGoalDestination(?tdsgr2, ?sg) ∧ belongsToSRModel(?tdsgr1, ?srm) ∧ belongsToSRModel(?tdsgr2, ?srm) ∧ belongsToSRModel(?sg, ?srm)*

Appendix D: The sQuality Ontology Classes and Properties

The aim of this appendix is to shows the implemented classes and associated properties that specify the sQuality ontology using OWL-DL. These classes and properties are shown in Table D.1.

The sQuality ontology is comprised of approximately 15 classes as shown in Table D.1. The **SIG_Diagram** class is designed in order to represent the NFR framework soft goal interdependency graph. In fact, the SIG diagram is comprised of the classes **NFRT_Type_SoftGoal, NFR_SoftGoal** and **Operationalisation_SoftGoal** that represent the three soft goal types NFR type (i.e., the root of the SIG), the NFR soft goal and the operationalization soft goal respectively.

The object properties: *hasNFRSGDecompositionRelation, hasSoftGoalDecompositionRelation* and *hasOperationalisationDecoRelation* relate the three soft goal classes above to their relevant decomposition relation classes that respectively are: **NFR_SoftGoal_Decomposition, SoftGoal_Decomposition** and **Operationalisation_Decomposition**. Each of those decomposition relation classes uses their functional string data type property *hasDecompositionType* in order to allow for the manual selection of the desired decomposition value (e.g., "AND" or "OR" but not both).

The **Explicit_Interdependency** class is constructed in order to represent the relation that is directed from the class **SG** (i.e., Operationalisation_SoftGoal class or the NFR_SoftGoal class) to the NFR_SoftGoal class or the NFR_Type_SoftGoal class using the object property *hasExplicitInterdependency*. The explicit interdependency relation class is characterised by the functional string data type property: *hasContributionValue*, which indicates the contribution value (e.g., "+", "++", "-" or "- -"). In addition, the interdependency relation class is described through identifying its offspring and parent using the two object properties respectively: *hasXOffspring* and *hasXParent*.

Similarly, the implicit interdependency relation among the soft goals is represented using the class **Implicit_Interdependency**. The essential object properties that characterises the implicit interdependency relation are: *hasiMean*, *hasiEnd* and the *hasContributionValue.* Using the first two properties, an implicit interdepdency relation instance is bonded by the

Operationalisation_SoftGoal class, as a mean, and by the NFR_Type_SoftGoal or the NFR_SoftGoal, as its end. The third object property has been already described above.

Finally, the functional string data type *hasLabelValue* (e.g., S, D, C and U) is defined for the evaluation step purpose that is normally carried out manually using the human-aided decisions.

Table D.1: The sQuality Ontology Classes and Properties

Class	Description	Attributes
NFR_Type_ SoftGoal	The main NFR soft goal in the SIG. It represents the root of the refined soft goals.	1- hastopic of type Topic, 2- hasLabelValue: String, 3- hasSubSoftGoal of type NFR_SoftGoal, 4- hasNFRSGDecompositionRelation of type NFR_SoftGoal_Decomposition, and 5- belongsToSIGDiagram of type SIG_Diagram.
Topic	The parameter in the NFR type soft goal.	1- belongsToSIGDiagram of type SIG_Diagram.
NFR_SoftGoal	This soft goal represents a refined kind of NFR type soft goal. It is similar to the one in the SR model.	1- belongsToSIGDiagram of type SIG_Diagram, 2- hasNFRSGDecompositionRelation of type NFR_SoftGoal_Decomposition, 3- hasExplicitInterdependency of type Explicit_Interdependency 4- hasLabelValue: String, 5- hasImplicitInterdependency of type Implicit_Interdependency 6- hasSoftGoalDecompositionRelation of type SoftGoal_Decomposition, and 7- hasSubSoftGoal of type NFR_SoftGoal.

Table D.1 (Cont'd): The sQuality Ontology Classes and Properties

Class	Description	Attributes
Operationalisation_Soft Goal	The operationalisation soft goal in the SIG diagram. This kind of soft goal is identified in order to show how it is possible to achieve a soft goal. In addition, they are involved in identifying the positive and the negative trade-offs among other relevant soft goals.	1- belongsToSIGDiagram of type SIG_Diagram, 2- isDynamic: Boolean, 3- isStatic: Booelan, 4- hasLabelValue: String 5- operateSoftGoal of type NFR_SoftGoal, 6- operatationalizesNFRTypeSoftGoal of type NFR_Type_SoftGoal, 7- correlatesWithSoftGoal of type NFR_SoftGoal, 8- correlatesWithSoftGoal of type NFR_Type_SoftGoal, 9- hasExplicitInterdependency of type Explicit_Interdependency 10- hasImplicitInterdependency of type Implicit_Interdependency, 11- hasOperationalisationDecoRelation of type Operationalisation_Decomposition, and 12- hasSubOperationalisation of type Operationalisation_SoftGoal.
Dynamic_ Operationalisation	The dynamic operationalisation that refers to a function or operation.	1- belongsToSIGDiagram of type SIG_Diagram
Static_ Operationalisation	The static operationalisation that refers to data or information.	1- belongsToSIGDiagram of type SIG_Diagram
NFR_ SoftGoal_Decompositi on	The decomposition relation when the parent is the NFR type soft goal and the offspring is a soft goal. If the decomposition type is **AND**, then this relation is addressed when **all** the offspring soft goals are addressed. And if the decomposition type is **OR**, then this relation is addressed when **at least one of** the offspring soft goals is addressed.	1- belongsToSIGDiagram of type SIG_Diagram, 2- hasDecompositionType: String, 3- hasNFRSoftGoalParent of type NFR_Type_SoftGoal, and 4- hasSoftGoalOffSpring of type NFR_SoftGoal.

Table D.1 (Cont'd): The sQuality Ontology Classes and Properties

Class	Description	Attributes
SoftGoal_ Decomposition	The decomposition relation between refined soft goals. The parent is the soft goal. The offspring is a soft goal too. If the decomposition type is AND then, this relation is addressed when **all** the offspring soft goals are addressed. If the decomposition type is OR then, this relation is addressed when **at least one of** the offspring soft goals is addressed.	1- belongsToSIGDiagram of type SIG_Diagram, 2- hasDecompositionType: String. 3- haSoftGoalParent of type NFR_SoftGoal, and 4- hasSoftGoalOffSpring of type NFR_SoftGoal
Operationalisation_Decomposition	The decomposition relation among operationalisation soft goals. The parent is an operationalisation and the offspring is an operationalisation too. If the decomposition type is **AND**, this relation is achieved when **all** the offspring operationalisation soft goals are addressed. If the decomposition type is **OR,** then this relation is achieved when **at least one of** the offspring operationalisation soft goals is addressed.	1- belongsToSIGDiagram of type SIG_Diagram., 2- hasDecompositionType: String. 3- hasOperationalisationOffSpring of type Operationalisation_SoftGoal, and 4- hasOperationalisationParent of type Operationalisation_SoftGoal
Explicit_Contribution	The explicit interdependency relation in the NFR framework.	1- belongsToSIGDiagram of type SIG_Diagram, 2- hasXOffspring of type Operationalisation_SoftGoal, 3- hasXParent of type SG, 4- hasXOffspring of type SG, 5- hasXParent of type (NFR_SoftGoal or NFR_Type_SoftGoal) , and 6- hasContributionValue: String.

Table D.1 (Cont'd): The sQuality Ontology Classes and Properties

Class	Description	Attributes
Implicit_Interdependency	The implicit interdependency relation in the NFR framework in order to explore the correlation and trade offs.	1- belongsToSIGDiagram of type SIG_Diagram, 2- hasiEnd only (NFR_SoftGoal or NFR_Type_SoftGoal), 3- hasiMean of type Operationalisation_SoftGoal, 4- hasPositiveCorrelation: Boolean, and 5- hasNegativeCorrelation: Boolean.
SIG_Diagram	The soft goal interdependency graph for a particular NFR type.	1- designsTheNFRSoftGoal of type NFR_Type_SoftGoal.

Appendix E: Ontologising the NFR Framework Modelling Steps

This appendix presents ontologising the steps of the NFR framework using OWL-DL language as shown in Table E.1. This implantation is employed in the sQuality ontology instantiator component. Table E.1 embodies about 75 SWRL rules. Each NFR framework step is semantically represented in Table E.1, where each concept within the steps is detected as a mandatory class for the sQuality ontology. The nature of the NFR framework, which is very rich with the quality-oriented concepts, requires a permanent user's intervention (i.e., highly depending on the human-based decisions). Therefore, the identified classes, the OWL-DL and the SWRL rule features work all together in managing the rich knowledge where the user is only requested to fill knowledge space that acts as a template. It is necessary to highlight that the rare decomposition and contribution cases in Section (4.3.4.4) are manually carried out in order to minimize ambiguity and confusion.

Table E.1: Ontologising the NFR Framework Steps Using OWL-DL

NFR Framework Step	**Step1**: Identify the main NFR soft goal (the root of the SIG) with its topic parameter that is comprised of related business process name along with the involved role/actor names.
Related Classes	**NFR_Type_SoftGoal, Topic, SIG_Diagram**
Class Properties	**hasTopic:** this property only associates the main NFR identified in order to bind it with the topic parameter in the further refinemnts within the SIG. **belongsToSIGDiagram:** all the identified concepts in this step must belong to the soft goal interdependency graph of the main NFR soft goal and with respect to the topic. **designsTheNFRSoftGoal**: this property charectrise a SIG diagram with the related main NFR soft goal.
Class Instances	Main NFR type that is the root of the SIG diagram along the topic parameter.
OWL Resrictions	**SIG_Diagram**: ∀ *designsTheNFRSoftGoal only NFR_Type_SoftGoal.* **NFR_Type_SoftGoal**: ∀ *hasTopic only Topic*. ∀ *belongsToSIGDiagram only SIG_Diagram*.

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

SWRL Rules	**Rule-Identifying_Main_NFR_Type_And_Topic:** NFR_Type_SoftGoal(?nfrtsg) ∧ SIG_Diagram(?sig) ∧ hasTopic(?nfrtsg, ?t) ∧ belongsToSIGDiagram(?nfrtsg, ?sig) → Topic(?t) ∧ designsTheNFRSoftGoal(?sig, ?nfrtsg)
NFR Framework Step	**Step 2:** Decompose the main NFR type identified in step 1 into sub NFR soft goals using AND or OR decomposition relationships.
Related Classes	**NFR_Type_SoftGoal, NFR_SoftGoal, NFR_SoftGoal_Decomposition, SIG_Diagram**
Class Properties	**hasSubSoftGoal:** to consider the sub soft goals of the identified main NFR type. **hasNFRSGDecompositionRelation:** to determine the required decomposition relation instance of the main NFR soft goal into its sub soft goals. This relation expresses that main NFR soft goal is decomposable. **hasDecompositionType**: this is a string functional property that allows to select a particular kind of decomposition , which is either the AND or the OR decomposition relation. **hasNFRSoftGoalParent** and **hasSoftGoalOffSpring:** the two properties are designed for the AND or the OR decomposition relations that must have the main NFR as the parent where its off springs are sub soft goals.
Class Instances	Instances from the classes that represent the decomposition of the main NFR (parent) into sub soft goals (offsprings).
OWL Restrictions	**NFR_Type_SoftGoal**: ∀ *hasNFRSGDecompositionRelation only NFR_SoftGoal_Decomposition.* ∀ *hasSubSoftGoal only NFR_SoftGoal.* ∀ *belongsToSIGDiagram only SIG_Diagram* **NFR_SoftGoal:** ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *hasNFRSGDecompositionRelation only NFR_SoftGoal_Decomposition.* **NFR_SoftGoal_Decomposition**: ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *hasNFRSoftGoalParent only NFR_Type_SoftGoal.* ∀ *hasSoftGoalOffSpring only NFR_SoftGoal.*

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

SWRL Rules	**Rule-Elaborating_NFR_Type_To_Soft_Goals_Using_AND_Relation**: *NFR_Type_SoftGoal(?nfrt) ⋀ hasSubSoftGoal(?nfrt, ?sg1) ⋀ hasSubSoftGoal(?nfrt, ?sg2) ⋀ hasDecompositionType(?andr1, "AND") ⋀ hasDecompositionType(?andr2, "AND") ⋀ belongsToSIGDiagram(?andr1, ?sig) ⋀ belongsToSIGDiagram(?andr2, ?sig) ⋀ hasNFRSGDecompositionRelation(?nfrt, ?andr1) ⋀ hasNFRSGDecompositionRelation(?nfrt, ?andr2) ⋀ hasNFRSGDecompositionRelation(?sg1, ?andr1) ⋀ hasNFRSGDecompositionRelation(?sg2, ?andr2) → NFR_SoftGoal(?sg1) ⋀ NFR_SoftGoal(?sg2) ⋀ NFR_SoftGoal_Decomposition(?andr1) ⋀ NFR_SoftGoal_Decomposition(?andr2) ⋀ hasNFRSoftGoalParent(?andr1, ?nfrt) ⋀ hasNFRSoftGoalParent(?andr2, ?nfrt) ⋀ hasSoftGoalOffSpring(?andr1, ?sg1) ⋀ hasSoftGoalOffSpring(?andr2, ?sg2) ⋀ belongsToSIGDiagram(?sg1, ?sig) ⋀ belongsToSIGDiagram(?sg2, ?sig) ⋀ belongsToSIGDiagram(?nfrt, ?sig)* **Rule-Elaborating_NFR_Type_To_Soft_Goals_Using_OR_Relation:** *NFR_Type_SoftGoal(?nfrt) ⋀ hasSubSoftGoal(?nfrt, ?sg1) ⋀ hasSubSoftGoal(?nfrt, ?sg2) ⋀ hasNFRSGDecompositionRelation(?nfrt, ?or1) ⋀ hasNFRSGDecompositionRelation(?nfrt, ?or2) ⋀ hasNFRSGDecompositionRelation(?sg1, ?or1) ⋀ hasNFRSGDecompositionRelation(?sg2, ?or2) ⋀ hasDecompositionType(?or1, "OR") ⋀ hasDecompositionType(?or2, "OR") ⋀ belongsToSIGDiagram(?or1, ?sig) ⋀ belongsToSIGDiagram(?or2, ?sig) → NFR_SoftGoal(?sg1) ⋀ NFR_SoftGoal(?sg2) ⋀ NFR_SoftGoal_Decomposition(?or1) ⋀ NFR_SoftGoal_Decomposition(?or2) ⋀ hasNFRSoftGoalParent(?or1, ?nfrt) ⋀ hasNFRSoftGoalParent(?or2, ?nfrt) ⋀ hasSoftGoalOffSpring(?or1, ?sg1) ⋀ hasSoftGoalOffSpring(?or2, ?sg2) ⋀ belongsToSIGDiagram(?sg1, ?sig) ⋀ belongsToSIGDiagram(?sg2, ?sig) ⋀ belongsToSIGDiagram(?nfrt, ?sig)*
NFR Framework Step	**Step 3**: Decompose the generated soft goals in step 2 into sub-soft goals.
Related Classes	**NFR_SoftGoal, SIG_Diagram, SoftGoal_Decomposition,**
Class Properties	**hasSubSoftGoal**: any soft goal that is decomposable into sub soft goals must use this property in order to determine the soft goal off springs. **hasSoftGoalDecompositionRelation** : a decomposable soft goal into sub soft goals must be related to the class SoftGoal_Decomposition using this object property. **hasSoftGoalParent** and **hasSoftGoalOffSpring**: the two properties must aasociate the AND or the OR decomposition relation between soft goals in order to indicate the parent and its offspring soft goals. **hasDecompositionType, belongsToSIGDiagram**.
Class Instances	Instances from the classes that represent the decomposition between soft goals (parent and off springs).

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

OWL Restrictions	**NFR_SoftGoal:** ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *hasSoftGoalDecompositionRelation only SoftGoal_Decomposition.* ∀ *hasSubSoftGoal only NFR_SoftGoal*. **SoftGoal_Decomposition**: ∀ *belongsToSIGDiagram only SIG_Diagram* ∀ *hasSoftGoalOffSpring only NFR_SoftGoal* ∀ *hasSoftGoalParent only NFR_SoftGoal*
SWRL Rules	**Rule_Decomposing_Soft_Goal_To_Soft_Goals_Using_AND_Relation**: NFR_SoftGoal(?sgp) ∧ hasSubSoftGoal(?sgp, ?sg1) ∧ hasSubSoftGoal(?sgp, ?sg2) ∧ hasDecompositionType(?andr1, "AND") ∧ hasDecompositionType(?andr2, "AND") ∧ belongsToSIGDiagram(?andr1, ?sig) ∧ belongsToSIGDiagram(?andr2, ?sig) ∧ hasNFRSGDecompositionRelation(?sgp, ?andr1) ∧ hasNFRSGDecompositionRelation(?sgp, ?andr2) ∧ hasNFRSGDecompositionRelation(?sg1, ?andr1) ∧ hasNFRSGDecompositionRelation(?sg2, ?andr2) → NFR_SoftGoal(?sg1) ∧ NFR_SoftGoal(?sg2) ∧ NFR_SoftGoal_Decomposition(?andr1) ∧ NFR_SoftGoal_Decomposition(?andr2) ∧ hasNFRSoftGoalParent(?andr1, ?sgp) ∧ hasNFRSoftGoalParent(?andr2, ?sgp) ∧ hasSoftGoalOffSpring(?andr1, ?sg1) ∧ hasSoftGoalOffSpring(?andr2, ?sg2) ∧ belongsToSIGDiagram(?sg1, ?sig) ∧ belongsToSIGDiagram(?sg2, ?sig) ∧ belongsToSIGDiagram(?sgp, ?sig) **Rule_Decomposing_Soft_Goal_To_Soft_Goals_Using_OR_Relation**: NFR_SoftGoal(?sgp) ∧ hasSubSoftGoal(?sgp, ?sg1) ∧ hasSubSoftGoal(?sgp, ?sg2) ∧ hasNFRSGDecompositionRelation(?sgp, ?or1) ∧ hasNFRSGDecompositionRelation(?sgp, ?or2) ∧ hasNFRSGDecompositionRelation(?sg1, ?or1) ∧ hasNFRSGDecompositionRelation(?sg2, ?or2) ∧ hasDecompositionType(?or1, "OR") ∧ hasDecompositionType(?or2, "OR") ∧ belongsToSIGDiagram(?or1, ?sig) ∧ belongsToSIGDiagram(?or2, ?sig) → NFR_SoftGoal(?sg1) ∧ NFR_SoftGoal(?sg2) ∧ NFR_SoftGoal_Decomposition(?or1) ∧ NFR_SoftGoal_Decomposition(?or2) ∧ hasNFRSoftGoalParent(?or1, ?sgp) ∧ hasNFRSoftGoalParent(?or2, ?sgp) ∧ hasSoftGoalOffSpring(?or1, ?sg1) ∧ hasSoftGoalOffSpring(?or2, ?sg2) ∧ belongsToSIGDiagram(?sg1, ?sig) ∧ belongsToSIGDiagram(?sg2, ?sig) ∧ belongsToSIGDiagram(?sgp, ?sig)

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

NFR Framework Step	**Step 4:** Draw the explicit interdependency relations between operatioanalisation soft goals and NFR soft goals. This kind of relations by default has the some positive contribution that is either help (+) or make (++) but not both. However, nothing is possible to prevent identifying explicit interdependency relations with a negative contribution. The explicit relation is represented using a solid directed line from the mean (operationalization) to the end (NFR soft goal)
Related Classes	**NFR_ SoftGoal, Operationalisation_SoftGoal, Explicit_Interdependency.**
Class Properties	**operateSoftGoal**: this property sets the operationalization soft goal that is designed in order to address the targeted NFR soft goal. It is required to use this property before identifying the explicit interdependency relation. **operationaliseNFRTypeSoftGoal**: this is illustrate the rare contribution case from an operationalization soft goal to NFR type soft goal. This property is manually fulfilled. **hasExplicitInterdependency**: any operationalization soft goal identified to achieve a soft goal must both have this property in order to identifiy its explicit interdependency relationship. **hasXParent** and **hasXOffspring**: the explicit interdependency relation indicates its parent and its offspring using the two properties respectively. **hasContributionValue**: an explicit interdependency relation must be charecterised with a contribution value. This property is identified as a string functional where the allowed entries are: "++" to refer to make, "+" refers to help, "-" refers to hurt and "- -" refers to break. **belongsToSIGDiagram**
Class Instances	The resulted instances from this step is the explicit interdependency relations that source from operationalization soft goal to the NFR soft goal destination in order to present the operationalisations that achieve the NFR soft goal.
OWL Restriction	**Operationalisation_SoftGoal**: ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *operateSoftGoal only NFR_SoftGoal* ∀ *operationaliseNFRTypeSoftGoal only NFR_Type_SoftGoal* ∀ *hasExplicitInterdependency only Explicit_Interdependency* **Explicit_Interdependency**: ∀ *belongsToSIGDiagram only SIG_Diagram* ∀ *hasXParent only Operationalisation_SoftGoal* ∀ *hasXOffspring only (NFR_SoftGoal* ***or*** *NFR_Type_SoftGoal)* **NFR_SoftGoal:** ∀ *belongsToSIGDiagram only SIG_Diagram* ∀ *hasExplicitInterdependency only Explicit_Interdependency*

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

<table>
<tr><td>SWRL Rules</td><td>Rule-Identifying_Possible_Make_ Operationslisations_For_NFR_SoftGoal

NFR_SoftGoal(?sgo) ∧ belongsToSIGDiagram(?xr, ?sig) ∧ hasExplicitInterdependency(?sgo, ?xr) ∧ operateSoftGoal(?opsg, ?sgo) ∧ hasExplicitInterdependency(?opsg, ?xr) ∧ hasContributionValue(?xr, "++") → Operationalisation_SoftGoal(?opsg) ∧ Explicit_Interdependency(?xr) ∧ hasXOffspring(?xr, ?opsg) ∧ hasXParent(?xr, ?sgo) ∧ hasContributionValue(?xr, "++") ∧ belongsToSIGDiagram(?opsg, ?sig) ∧ belongsToSIGDiagram(?sgo, ?sig)

Rule-Identifying_Possible_Help_ Operationalisation_For_NFR_SoftGoal:

NFR_SoftGoal(?sgo) ∧ belongsToSIGDiagram(?xr, ?sig) ∧ operateSoftGoal(?opsg, ?sgo) ∧ hasExplicitInterdependency(?opsg, ?xr) ∧ hasExplicitInterdependency(?sgo, ?xr) ∧ hasContributionValue(?xr, "+") → Operationalisation_SoftGoal(?opsg) ∧ Explicit_Interdependency(?xr) ∧ hasXOffspring(?xr, ?opsg) ∧ hasXParent(?xr, ?sgo) ∧ hasContributionValue(?xr, "+") ∧ belongsToSIGDiagram(?opsg, ?sig) ∧ belongsToSIGDiagram(?sgo, ?sig)

Rule-Identifying_Possible_Hurt_Operationalisation_For_NFR_SoftGoal:

NFR_SoftGoal(?sgo) ∧ belongsToSIGDiagram(?xr, ?sig) ∧ operateSoftGoal(?opsg, ?sgo) ∧ hasExplicitInterdependency(?opsg, ?xr) ∧ hasExplicitInterdependency(?sgo, ?xr) ∧ hasContributionValue(?xr, "-") → Operationalisation_SoftGoal(?opsg) ∧ Explicit_Interdependency(?xr) ∧ hasXOffspring(?xr, ?opsg) ∧ hasXParent(?xr, ?sgo) ∧ hasContributionValue(?xr, "-") ∧ belongsToSIGDiagram(?opsg, ?sig) ∧ belongsToSIGDiagram(?sgo, ?sig)

Rule-Identifying_Possible_Break_Operationalisation_For_NFR_SoftGoal :

NFR_SoftGoal(?sgo) ∧ belongsToSIGDiagram(?xr, ?sig) ∧ operateSoftGoal(?opsg, ?sgo) ∧ hasExplicitInterdependency(?opsg, ?xr) ∧ hasExplicitInterdependency(?sgo, ?xr) ∧ hasContributionValue(?xr, "--") → Operationalisation_SoftGoal(?opsg) ∧ Explicit_Interdependency(?xr) ∧ hasXOffspring(?xr, ?opsg) ∧ hasXParent(?xr, ?sgo) ∧ hasContributionValue(?xr, "--") ∧ belongsToSIGDiagram(?opsg, ?sig) ∧ belongsToSIGDiagram(?sgo, ?sig)</td></tr>
<tr><td>NFR Framework Step</td><td>Step 5: Decompose an operatioanlisation into sub operationalisations using the AND or the OR operationalization decomposition relations.</td></tr>
<tr><td>Related Classes</td><td>Operationalisation_SoftGoal, SIG_Diagram, Operationalisation_Decomposition.</td></tr>
</table>

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

Class Properties	**hasSubOperationalisation**: this property tells that a decomposable operationalization soft goal has sub operationalization soft goals. **hasOperationalisationDecoRelation**: a decomposable operationlaisation soft goal must use this object property in order to identify its decomposition relation. **hasOperationalisationParent** and **hasOperationalisationOffSpring**: the two properties are required with the AND or the OR operationalization decomposition relation to indicate the parent and the off springs. **hasDecompositionType, belongsToSIGDiagram.**
Class Instances	AND or OR operationalization decomposition relation instances along with instances of the associated properties that determine the parent and the offspring of the relation.
OWL Restrictions	**Operationalisation_SoftGoal**: ∀ *belongsToSIGDiagram only SIG_Diagram*. ∀ *hasOperationalisationDecoRelation only Operationalisation_Decomposition* ∀ *hasSubOperationalisation only Operationalisation_SoftGoal* **Operationalisation_Decomposition:** ∀ *hasOperationalisationOffSpring only Operationalisation_SoftGoal* ∀ *hasOperationalisationParent only Operationalisation_SoftGoal* ∀ *belongsToSIGDiagram only SIG_Diagram*.

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

SWRL Rules	**Rule-Decomposing_Operationalisation_To_Operationalisation_Using_AND_Relation:** Operationalisation_SoftGoal(?optp) ∧ hasSubOperationalisation(?optp, ?op1) ∧ hasSubOperationalisation(?optp, ?op2) ∧ hasOperationalisationDecoRelation(?optp, ?aoptr1) ∧ hasOperationalisationDecoRelation(?op1, ?aoptr1) ∧ hasDecompositionType(?aoptr1, "AND") ∧ hasOperationalisationDecoRelation(?optp, ?aoptr2) ∧ hasOperationalisationDecoRelation(?op2, ?aoptr2) ∧ hasDecompositionType(?aoptr2, "AND") ∧ SIG_Diagram(?sig) ∧ belongsToSIGDiagram(?aoptr1, ?sig) ∧ belongsToSIGDiagram(?aoptr2, ?sig) → Operationalisation_SoftGoal(?op1) ∧ Operationalisation_SoftGoal(?op2) ∧ Operationalisation_Decomposition(?aoptr1) ∧ Operationalisation_Decomposition(?aoptr2) ∧ hasOperationalisationParent(?aoptr1, ?optp) ∧ hasOperationalisationOffSpring(?aoptr1, ?op1) ∧ hasOperationalisationParent(?aoptr2, ?optp) ∧ hasOperationalisationOffSpring(?aoptr2, ?op2) ∧ belongsToSIGDiagram(?op1, ?sig) ∧ belongsToSIGDiagram(?op2, ?sig) ∧ belongsToSIGDiagram(?optp, ?sig) **Rule-Decomposing_Operationalisation_To_Operationalisations_Using_OR_Relation** Operationalisation_SoftGoal(?optp) ∧ hasSubOperationalisation(?optp, ?op1) ∧ hasSubOperationalisation(?optp, ?op2) ∧ hasOperationalisationDecoRelation(?optp, ?oroptr1) ∧ hasOperationalisationDecoRelation(?op1, ?oroptr1) ∧ hasDecompositionType(?oroptr1, "OR") ∧ hasOperationalisationDecoRelation(?optp, ?oroptr2) ∧ hasOperationalisationDecoRelation(?op2, ?oroptr2) ∧ hasDecompositionType(?oroptr2, "OR") ∧ SIG_Diagram(?sig) ∧ belongsToSIGDiagram(?oroptr1, ?sig) ∧ belongsToSIGDiagram(?oroptr2, ?sig) → Operationalisation_SoftGoal(?op1) ∧ Operationalisation_SoftGoal(?op2) ∧ Operationalisation_Decomposition(?oroptr1) ∧ Operationalisation_Decomposition(?oroptr2) ∧ hasOperationalisationParent(?oroptr1, ?optp) ∧ hasOperationalisationOffSpring(?oroptr1, ?op1) ∧ hasOperationalisationParent(?oroptr2, ?optp) ∧ hasOperationalisationOffSpring(?oroptr2, ?op2) ∧ belongsToSIGDiagram(?op1, ?sig) ∧ belongsToSIGDiagram(?op2, ?sig) ∧ belongsToSIGDiagram(?optp, ?sig)

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

NFR Framework Step Adopted From (Cysnerio,)	**Step 6**: Classify the operationalisations that occupy the place of the leafs in the SIG into dynamic or static.
Related Classes	**Operationalisation_SoftGoal**
Class Properties	**isStatic**: is to set true for an operationalization soft goal if it represents data or information. **isDynamic**: is to set true for an operationalization soft goal if it represents function,action or activity.
Class Instances	Static and dynamic opertaionalisation soft goals in the SIG.
OWL Restrictions	-
SWRL Rules	**Rule-Determining_Static_Operationalisation**: Operationalisation_SoftGoal(?op) $\wedge$ isStatic(?op, true) → Static_Operationalisation(?op) **Rule-Determining_Dynamic_Operationalisation**: Operationalisation_SoftGoal(?op) $\wedge$ isDynamic(?op, true) → Dynamic_Operationalisation(?op)
NFR Framework Step	**Step** 7: Dealing with positive and negative implicit interdependencies among soft goals.
Related Classes	**Operationalisation_SoftGoal, Implicit_Interdependency, NFR_SoftGoal.**
Class Properties	**correlatesWithSoftGoal**: this relation detects the operationalisations that positively or negatively correlate with soft goals. **hasImplicitInterdependency**: Any operationalisation that is detected with a correlation with a soft goal must have this relation. A single operationalisation soft goal may have more than one interdependency relation. **hasiMean** *and* **hasiEnd**: an implicit interdependency between soft goals must be identified along with its mean and end. Besides, the implicit interdependency is charectrised with the some contribution. The two relations are charectirised as functional. **belongsToSIGDiagram**
Class Instances	The resulted instances are the implicit interdependency relations among soft goals along with their related properties that sets the source (mean) the destination (end), and the positive/negative contribution.

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

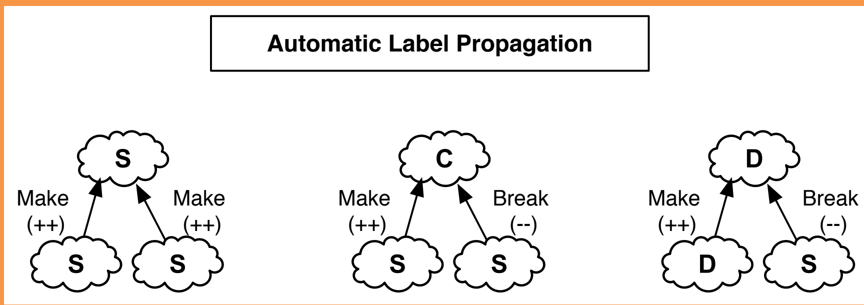
OWL Restrictions	**Operationalisation_SoftGoal**: ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *correlatesWithSoftGoal only NFR_SoftGoal* ∀ *correlatesWithSoftGoal only NFR_Type_SoftGoal* ∀ *hasImplicitInterdependency only Implicit_Interdependency* **Implicit_Interdependency:** ∀ *belongsToSIGDiagram only SIG_Diagram* ∀ *hasiEnd only (NFR_SoftGoal or NFR_Type_SoftGoal)* ∀ *hasiMean only Operationalisation_SoftGoal* **NFR_SoftGoal** ∀ *hasImplicitInterdependency only Implicit_Interdependency* ∀ *belongsToSIGDiagram only SIG_Diagram*
SWRL Rules	**Rule-Dealing_With_Positive_Implicit_Interdependencies_Among_SoftGoals**: Operationalisation_SoftGoal(?opip) ∧ NFR_SoftGoal(?sgi) ∧ belongsToSIGDiagram(?ir, ?sig) ∧ correlatesWithSoftGoal(?opip, ?sgi) ∧ hasImplicitInterdependency(?opip, ?ir) ∧ hasImplicitInterdependency(?sgi, ?ir) ∧ hasPositiveCorrelation(?ir, true) → Implicit_Interdependency(?ir) ∧ hasiMean(?ir, ?opip) ∧ hasiEnd(?ir, ?sgi) ∧ belongsToSIGDiagram(?opip, ?sig) ∧ belongsToSIGDiagram(?sgi, ?sig) ∧ hasContributionValue(?ir, "+") **Rule-Dealing_With_Negative_Implicit_Interdependencies_Among_SoftGoals:** Operationalisation_SoftGoal(?opin) ∧ NFR_SoftGoal(?sgi) ∧ belongsToSIGDiagram(?ir, ?sig) ∧ correlatesWithSoftGoal(?opin, ?sgi) ∧ hasImplicitInterdependency(?opin, ?ir) ∧ hasImplicitInterdependency(?sgi, ?ir) ∧ hasNegativeCorrelation(?ir, true) → Implicit_Interdependency(?ir) ∧ hasiMean(?ir, ?opin) ∧ hasiEnd(?ir, ?sgi) ∧ belongsToSIGDiagram(?sgi, ?sig) ∧ belongsToSIGDiagram(?opin, ?sig) ∧ hasContributionValue(?ir, "-")

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

The NFR Framework Step	**Step 8:** Generalise NFR and operationalization soft goals into soft goals. Then, select among alternatives and evaluate the individual impact of decisions. Besides, perform automatic evaluation using the label propagation technique.

Evaluation Catalogue

Individual impact of **offspring** with **label**:	Upon **parent label**, given offspring-parent contribution type: Break (--)	Some -	Hurt (-)	?	Help (+)	Some+	Make (++)	=
D	W+	W+	W+	U	W-	W-	D	D
C	C	C	C	U	C	C	C	C
U	U	U	U	U	U	U	U	U
S	D	W-	W-	U	W+	W+	S	S



Related Classes	**SG, Explicit_Interdependency**
Class Properties	**hasLabelValue**: This property is functional string property for the operationalization soft goals and the NFR soft goals in order to manually determine the selection. The impact of the selection may be detrmined automatically or manually based on the existence of relevant rule. **hasExplicitInterdependency, hasXParent, hasXOffspring, hasContributionValue**
Class Instances	Filled soft goals, which are from the kind of the operationalization and the NFR, that are related with explicit interdependency relations either as a parent or as an offspring.
OWL Restrictions	**SG: (this class refers to any soft goal)** ∀ *belongsToSIGDiagram only SIG_Diagram.* ∀ *hasExplicitInterdependency only Explicit_Interdependency.* **Explicit_Interdependency:** ∀ *hasXOffspring only SG* ∀ *hasXParent only SG*
SWRL Rules	**Rule-generalising_NFR_Soft_Goal:** NFR_SoftGoal(?sg) → SG(?sg) **Rule-generalising_operationalisation_soft_goal** Operationalisation_SoftGoal(?sg) → SG(?sg) --- **Rule-Evaluating_Indivisual_Impact_COffspring_Break_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "--") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr,

	?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C"). **Rule-Evaluating_Indivisual_Impact_COffspring_Equal_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "=") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C"). **Rule-Evaluating_Indivisual_Impact_COffspring_Help_Relation**: SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C") **Rule-Evaluating_Indivisual_Impact_COffspring_Hurt_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C") **Rule-Evaluating_Indivisual_Impact_COffspring_Make_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "++") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C") **Rule-Evaluating_Indivisual_Impact_COffspring_SomeNegative_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C") **Rule-Evaluating_Indivisual_Impact_COffspring_SomePositive_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "C") **Rule-Evaluating_Indivisual_Impact_COffspring_Unknown_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "?") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "C") → hasLabelValue(?p, "U") **Rule-Evaluating_Indivisual_Impact_DOffspring_Break_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "--") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "S") **Rule-Evaluating_Indivisual_Impact_DOffspring_Equal_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "=") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_Indivisual_Impact_DOffspring_Help_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_DOffspring_Hurt_Relation:**

	SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_DOffspring_Make_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "++") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_Indivisual_Impact_DOffspring_SomeNegative_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_DOffspring_SomePositive_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_DOffspring_Unknow_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "?") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "D") → hasLabelValue(?p, "U") **Rule-Evaluating_Indivisual_Impact_SOffspring_Break_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "--") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "D") **Rule-Evaluating_Indivisual_Impact_SOffspring_Equal_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "=") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_Indivisual_Impact_SOffspring_Help_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_SOffspring_Hurt_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_SOffspring_Make_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "++") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_Indivisual_Impact_SOffspring_SomeNegative_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some-") ∧ hasXParent(?xr, ?p) ∧

	hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_SOffspring_SomePositive_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_SOffspring_Unknown_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "?") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "S") → hasLabelValue(?p, "U"). **Rule-Evaluating_Indivisual_Impact_UOffspring_Any_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "U") → hasLabelValue(?p, "U") **Rule-Evaluating_Indivisual_Impact_WminusOffspring_Equal_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "=") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W-") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_WminusOffspring_SomeNegative_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W-") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_WminusOffspring_SomePositive_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W-") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_WplusOffspring_Equal_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "=") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W+") → hasLabelValue(?p, "W+") **Rule-Evaluating_Indivisual_Impact_WplusOffspring_SomeNegative_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some-") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W+") → hasLabelValue(?p, "W-") **Rule-Evaluating_Indivisual_Impact_WplusOffspring_SomePositive_Relation:** SG(?p) ∧ SG(?os) ∧ hasExplicitInterdependency(?p, ?xr) ∧ hasContributionValue(?xr, "some+") ∧ hasXParent(?xr, ?p) ∧ hasXOffspring(?xr, ?os) ∧ hasLabelValue(?os, "W+") → hasLabelValue(?p, "W+")

<table>
<tr>
<td></td>
<td>

Rule-Automatic_Label_Propagation_DMake_SBreak:

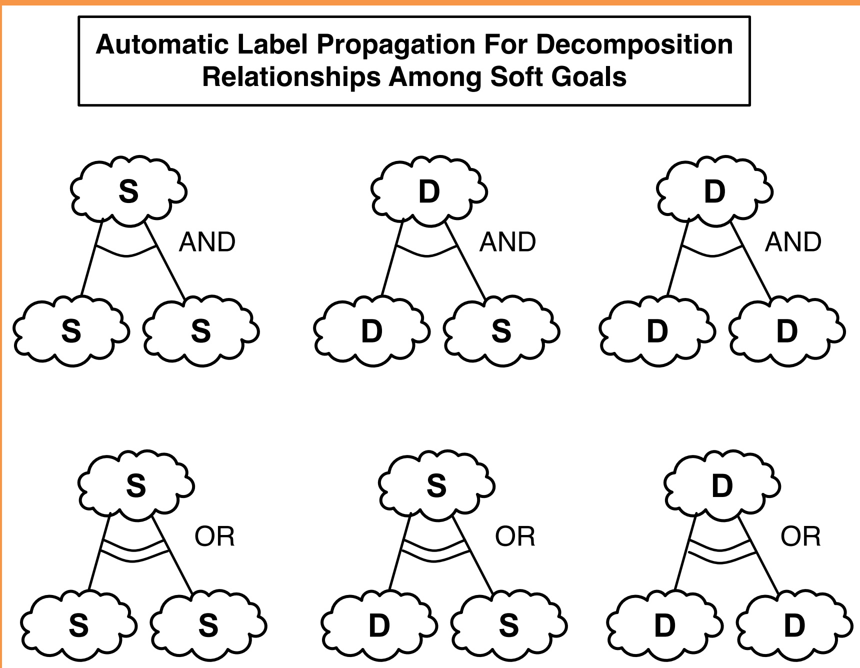
SG(?p) ∧ SG(?o1) ∧ SG(?o2) ∧ hasExplicitInterdependency(?p, ?xr1) ∧ hasXParent(?xr1, ?p) ∧ hasXOffspring(?xr1, ?o1) ∧ hasContributionValue(?xr1, "++") ∧ hasLabelValue(?o1, "D") ∧ hasExplicitInterdependency(?p, ?xr2) ∧ hasXParent(?xr2, ?p) ∧ hasXOffspring(?xr2, ?o2) ∧ hasContributionValue(?xr2, "--") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "D")

Rule-Automatic_Label_Propagation_SMake_SBreak:
SG(?p) ∧ SG(?o1) ∧ SG(?o2) ∧ hasExplicitInterdependency(?p, ?xr1) ∧ hasXParent(?xr1, ?p) ∧ hasXOffspring(?xr1, ?o1) ∧ hasContributionValue(?xr1, "++") ∧ hasLabelValue(?o1, "S") ∧ hasExplicitInterdependency(?p, ?xr2) ∧ hasXParent(?xr2, ?p) ∧ hasXOffspring(?xr2, ?o2) ∧ hasContributionValue(?xr2, "--") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "C")

Rule-Automatic_Label_Propagation_SMake_SMake:
SG(?p) ∧ SG(?o1) ∧ SG(?o2) ∧ hasExplicitInterdependency(?p, ?xr1) ∧ hasXParent(?xr1, ?p) ∧ hasXOffspring(?xr1, ?o1) ∧ hasContributionValue(?xr1, "++") ∧ hasLabelValue(?o1, "S") ∧ hasExplicitInterdependency(?p, ?xr2) ∧ hasXParent(?xr2, ?p) ∧ hasXOffspring(?xr2, ?o2) ∧ hasContributionValue(?xr2, "++") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S")

</td>
</tr>
<tr>
<td>NFR Framework Step</td>
<td>

Step 9: Perform top-down semi-automation evaluation for the decomposition relations among the three kinds of soft goals. Filling the label value of the off springs is manual where the resulted label results from the propagation are automatic.



</td>
</tr>
</table>

Table E.1 (Cont'd): Ontologising the NFR Framework Steps Using OWL-DL

Related Classes	**Operationalisation_SoftGoal, NFR_SoftGoal, NFR_Type_SoftGoal**
Class Properties	**hasLabelValue** **hasOperationalisationDecoRelation, hasOperationalisationParent, hasOperationalisationOffSpring.** **hasSoftGoalDecompositionRelation, hasNFRSoftGoalParent, hasNFRSoftGoalOffSpring.** **hasNFRSGDecompositionRelation, hasNFRSoftGoalParent, hasSoftGoalOffSpring**
Class Instances	The resulted label value from different soft goal decomposition relations.
OWL Restrictions	**-**
SWRL Rules	**Rule-Evaluating_AND_Decomposition_OperaSG_DD:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧ hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_OperaSG_DS:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧ hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_OperaSG_SD:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧

	hasOperationalisationDecoRelation(?p, ?dr1) ∧ hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_OperaSG_SS:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_OperaSG_DD:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_OR_Decomposition_OperaSG_DS:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2,

	?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_OperaSG_SD:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_OperaSG_SS:** Operationalisation_SoftGoal(?p) ∧ Operationalisation_SoftGoal(?o1) ∧ Operationalisation_SoftGoal(?o2) ∧ hasOperationalisationDecoRelation(?p, ?dr1) ∧hasOperationalisationDecoRelation(?o1, ?dr1) ∧ Operationalisation_Decomposition(?dr1) ∧ hasOperationalisationParent(?dr1, ?p) ∧ hasOperationalisationOffSpring(?dr1, ?o1) ∧ hasOperationalisationDecoRelation(?p, ?dr2) ∧ hasOperationalisationDecoRelation(?o2, ?dr2) ∧ Operationalisation_Decomposition(?dr2) ∧ hasOperationalisationParent(?dr2, ?p) ∧ hasOperationalisationOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S")
	Rule-Evaluating_AND_Decomposition_NFRSG_DD: NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRSG_DS:** NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧

	SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRSG_SD:** NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRSG_SS:** NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRSG_DS**: NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRSG_SD:** NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRSG_SS:**

	NFR_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasSoftGoalDecompositionRelation(?p, ?dr1) ∧ hasSoftGoalDecompositionRelation(?o1, ?dr1) ∧ SoftGoal_Decomposition(?dr1) ∧ hasSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasSoftGoalDecompositionRelation(?p, ?dr2) ∧hasSoftGoalDecompositionRelation(?o2, ?dr2) ∧ SoftGoal_Decomposition(?dr2) ∧ hasSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") ——————————————————— **Rule-Evaluating_AND_Decomposition_NFRTypeSG_DD**: NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧ hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRTypeSG_DS:** NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRTypeSG_SD**: NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧ hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_AND_Decomposition_NFRTypeSG_SS:** NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p,

	?dr2) ∧hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "AND") ∧ hasDecompositionType(?dr2, "AND") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRTypeSG_DD:** NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "D") **Rule-Evaluating_OR_Decomposition_NFRTypeSG_DS**: NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧ hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "D") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRTypeSG_SD:** NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧ hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "D") → hasLabelValue(?p, "S") **Rule-Evaluating_OR_Decomposition_NFRTypeSG_SS:** NFR_Type_SoftGoal(?p) ∧ NFR_SoftGoal(?o1) ∧ NFR_SoftGoal(?o2) ∧ hasNFRSGDecompositionRelation(?p, ?dr1) ∧ hasNFRSGDecompositionRelation(?o1, ?dr1) ∧ NFR_SoftGoal_Decomposition(?dr1) ∧ hasNFRSoftGoalParent(?dr1, ?p) ∧ hasSoftGoalOffSpring(?dr1, ?o1) ∧ hasNFRSGDecompositionRelation(?p, ?dr2) ∧hasNFRSGDecompositionRelation(?o2, ?dr2) ∧ NFR_SoftGoal_Decomposition(?dr2) ∧ hasNFRSoftGoalParent(?dr2, ?p) ∧ hasSoftGoalOffSpring(?dr2, ?o2) ∧ hasDecompositionType(?dr1, "OR") ∧ hasDecompositionType(?dr2, "OR") ∧ hasLabelValue(?o1, "S") ∧ hasLabelValue(?o2, "S") → hasLabelValue(?p, "S")

Appendix F: The CCR Processes Case Study

This appendix aims at presenting the original RAD BPMs and the associated NFR framework models that represent the CCR case study employed for the evaluation purposes of this research. The CCR process comprises six processes, where three are only selected for the evaluation. This is because the three appeared representative enough as shown in Chapter 7 (Section 7.3.1). The six CCR processes were originally designed in the PhD thesis work in (Aburub, 2006). The patient reception process, the cancer detection process and the cancer treatment process along with their associated NFR framework models are shown in Section F.1, F.2 and F.3 respectively.

F.1 The Patient Reception Process

This section shown the RAD model of the patient reception process in Figure F.1, where its associated NFR framework models are shown in Figures F.2- F.5.

F.2 The Cancer Detection Process

This section shown the RAD model of the cancer detection process in Figure F.6, where its associated NFR framework models are shown in Figures F.7- F.12.

F.3 The Cancer Treatment Process

This section shown the RAD model of the cancer treatment process in Figure F.13, where its associated NFR framework models are shown in Figures F.14- F.18.

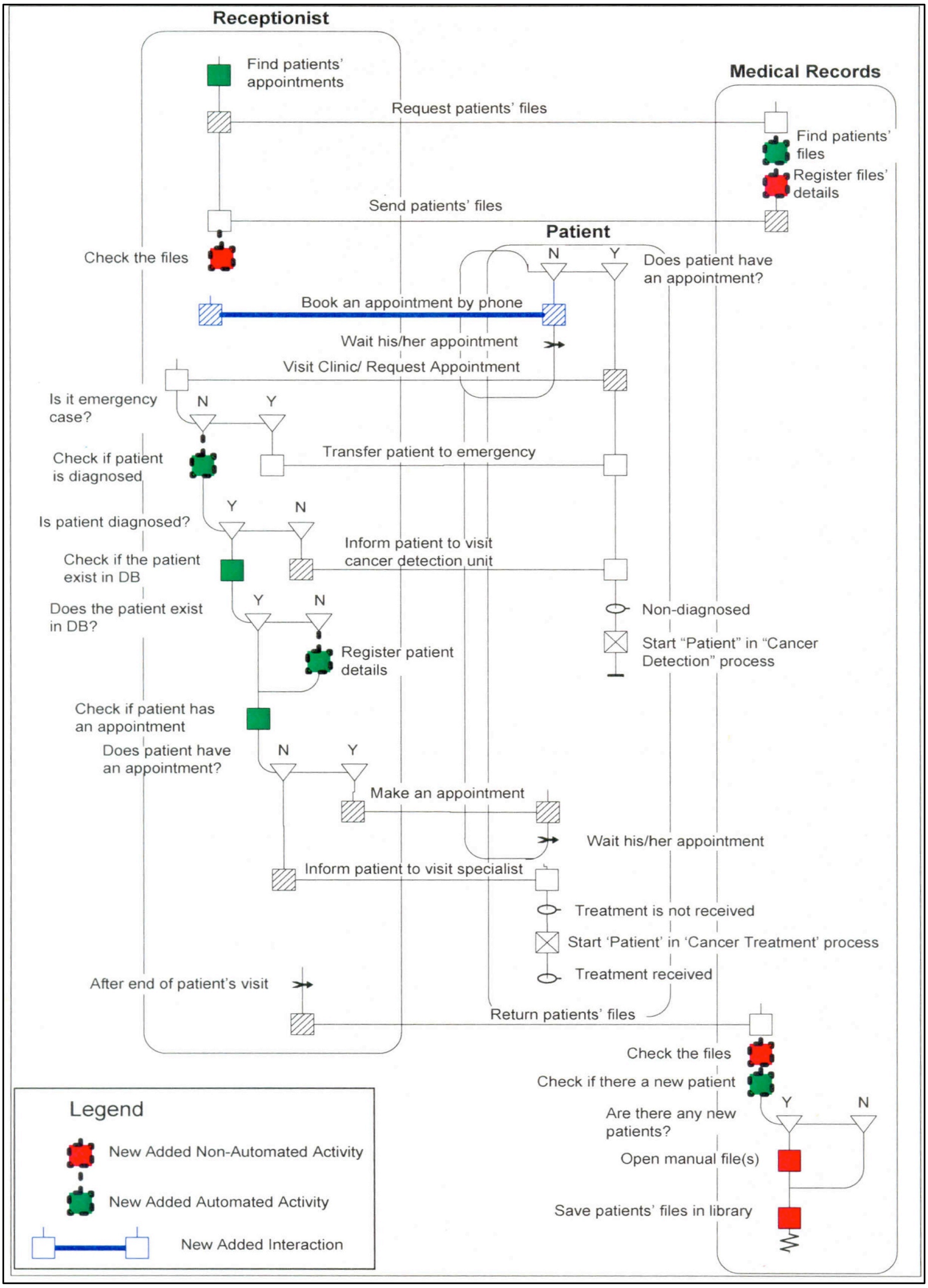


Figure F.1: The Patient Reception Process Designed Using RAD [Source: (Aburub, 2006), Used with the author's permission].

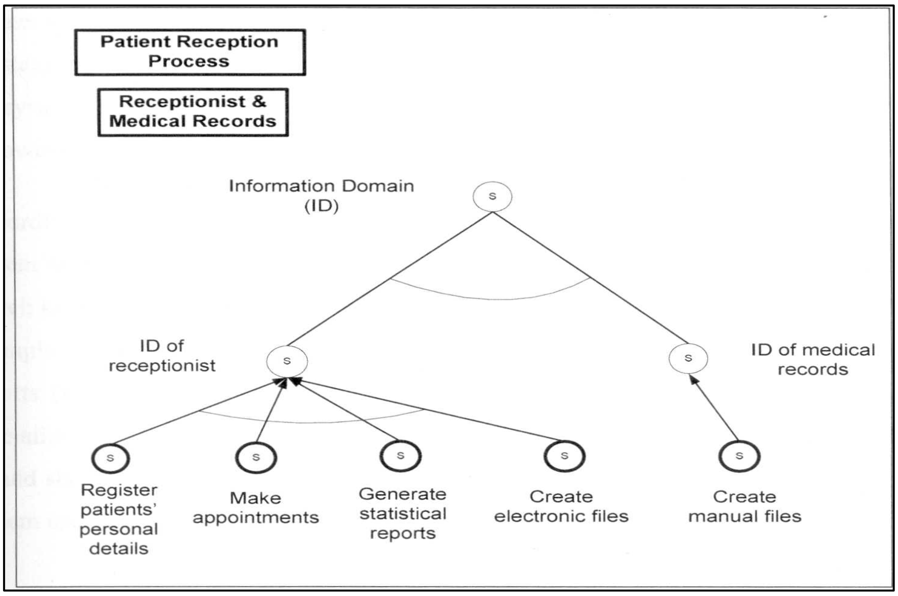


Figure F.2: The Information Domain NFR Framework for the Patient Reception Process having Ref no.1b [Source: (Aburub,2006), Used with the author's permission].

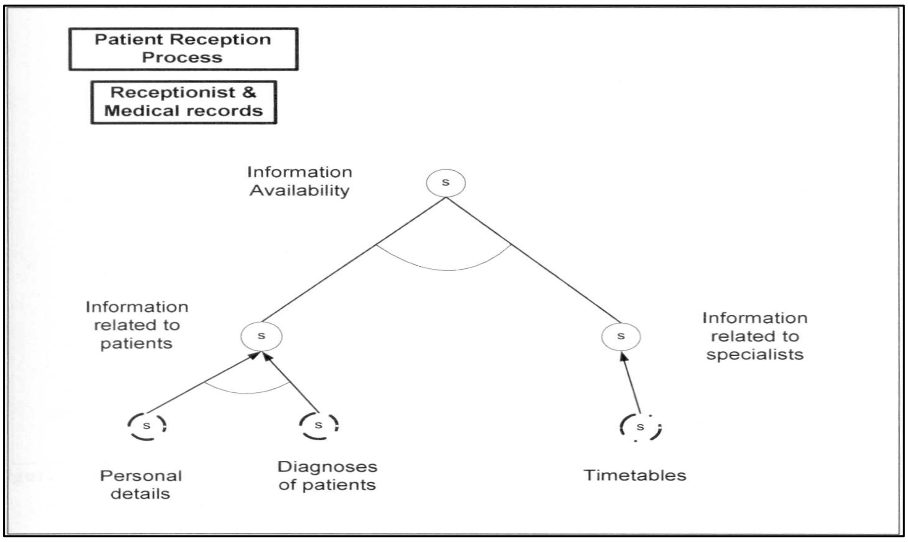


Figure F.3: The Information Availability NFR Framework for the Patient Reception Process having Ref no.1c [Source: (Aburub,2006), Used with the author's permission].

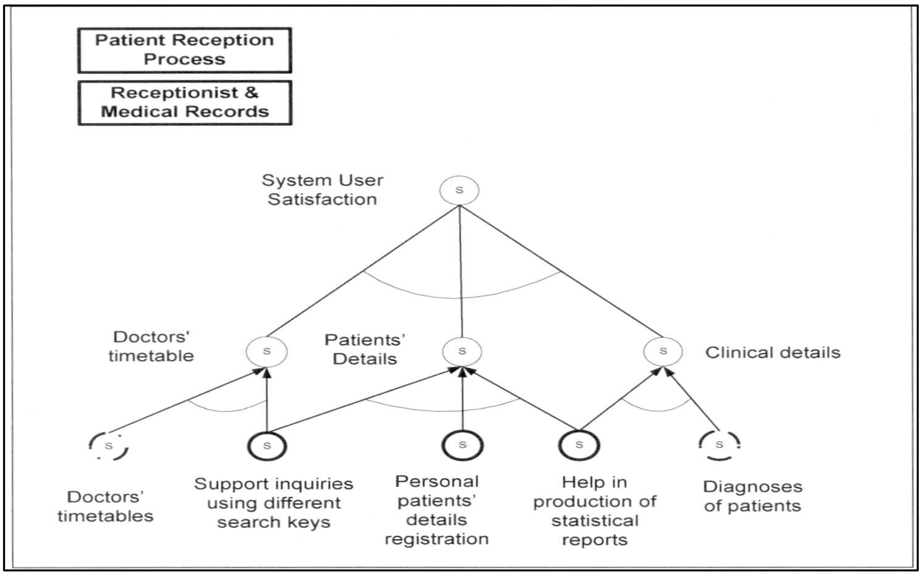


Figure F.4: The System User Satisfaction NFR Framework for the Patient Reception Process having Ref no.1d [Source: (Aburub, 2006), Used with the author's permission].

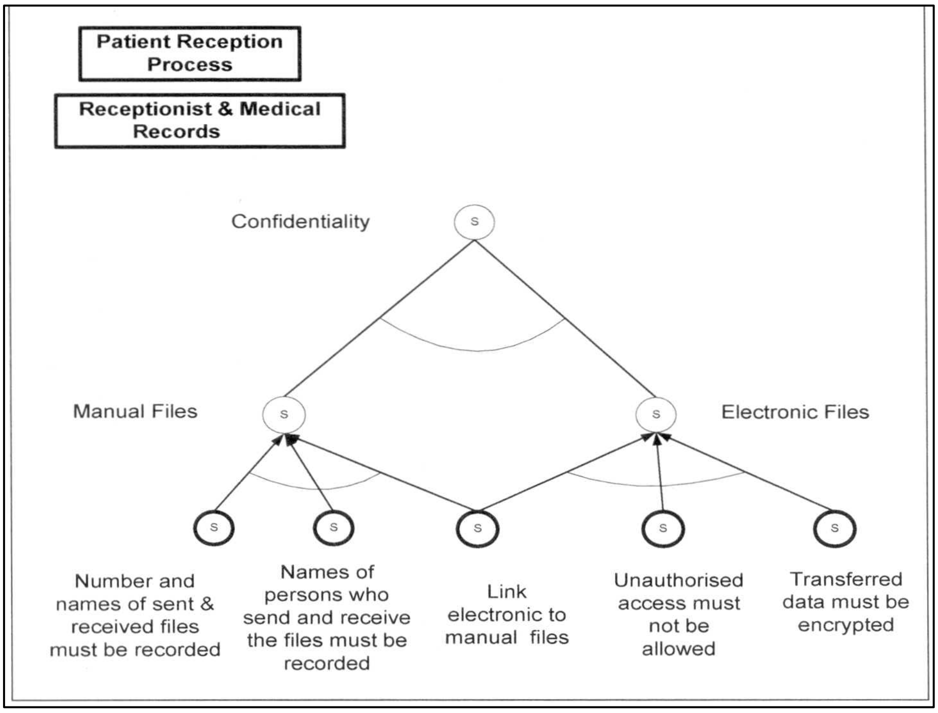


Figure F.5: The Confidentiality NFR Framework for the Patient Reception Process having Ref no.1e [Source: (Aburub, 2006), Used with the author's permission].

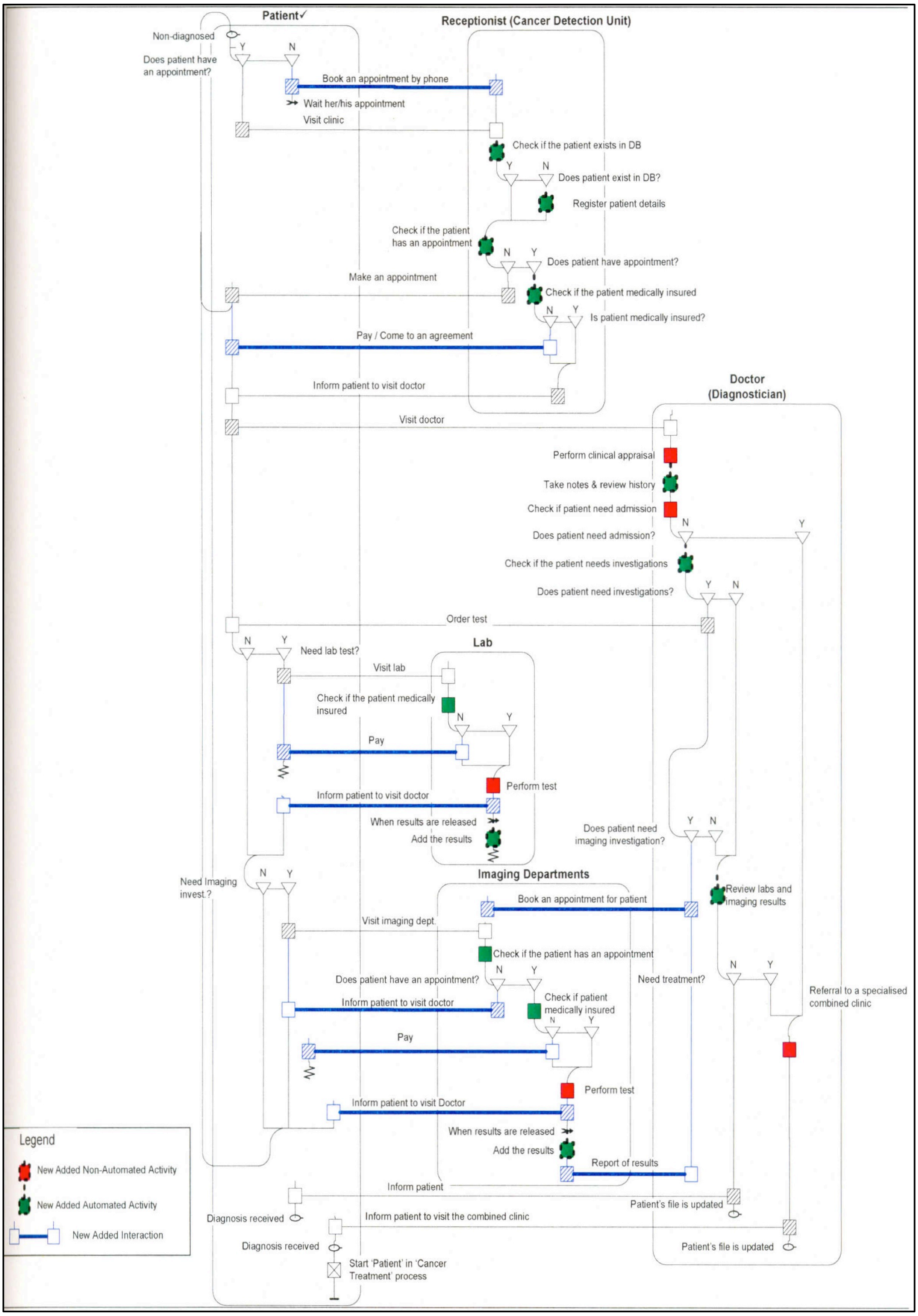


Figure F.6: The Cancer Detection Process Designed Using RAD [Source: (Aburub, 2006), Used with the author's permission].

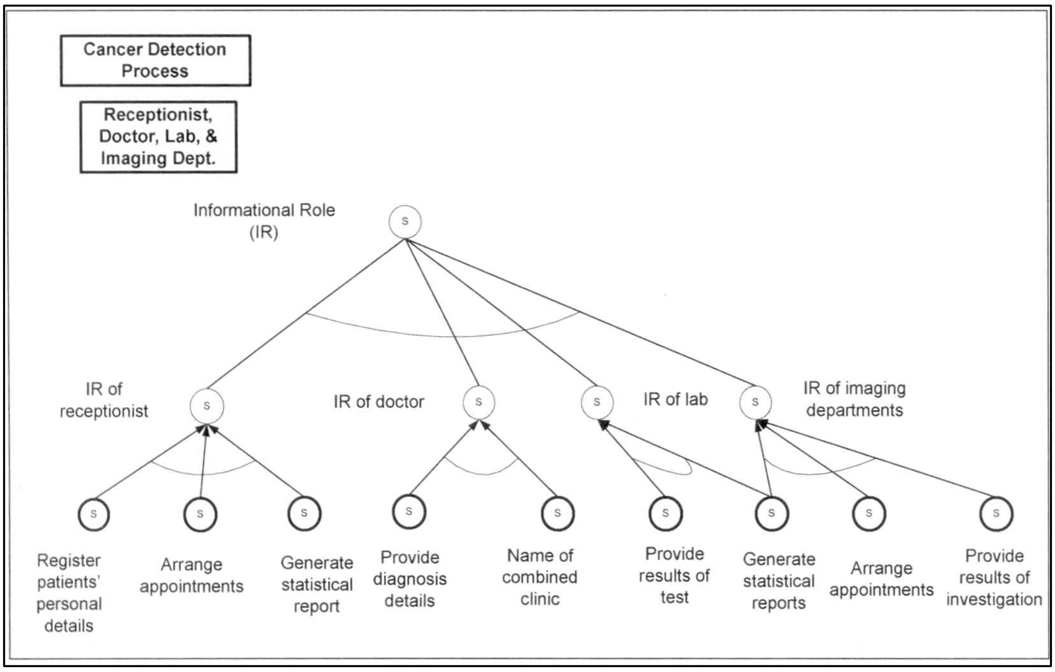


Figure F.7: The Information Domain NFR Framework for the Cancer Detection Process having Ref no.2b [Source: (Aburub,2006), Used with the author's permission].

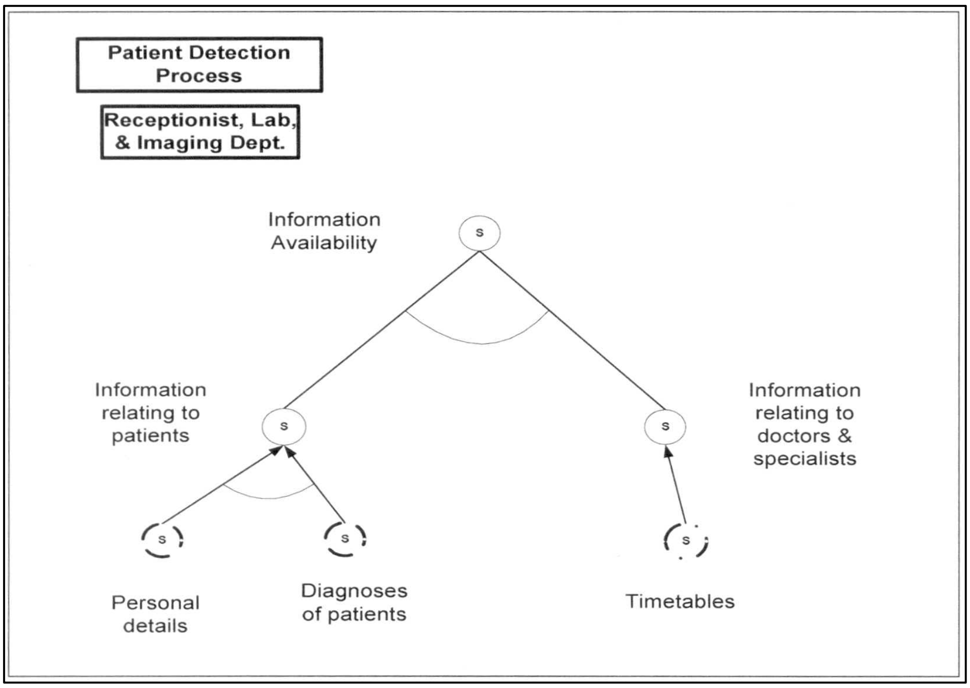


Figure F.8: The Information Availability NFR Framework for the Cancer Detection Process having Ref no.2c[Source: (Aburub,2006), Used with the author's permission].

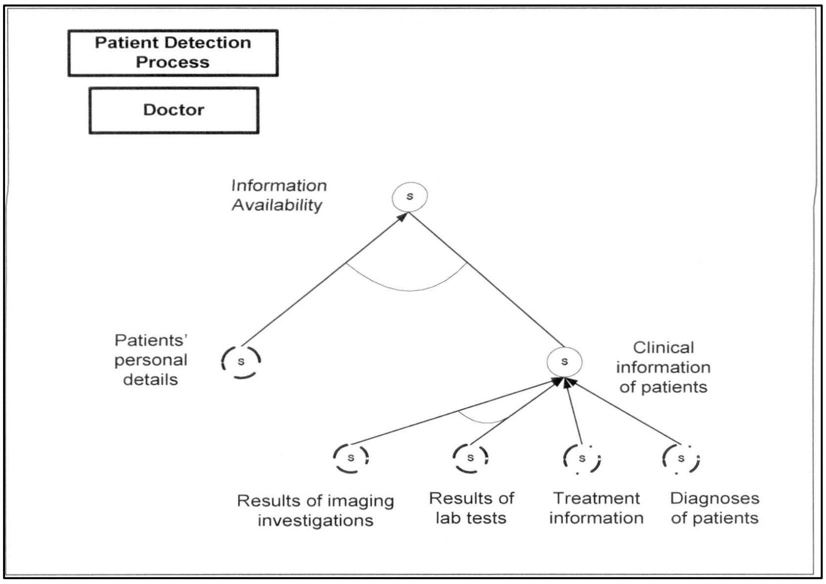


Figure F.9: The Information Availability NFR Framework for the Cancer Detection Process having Ref no.2d[Source: (Aburub,2006), Used with the author's permission].

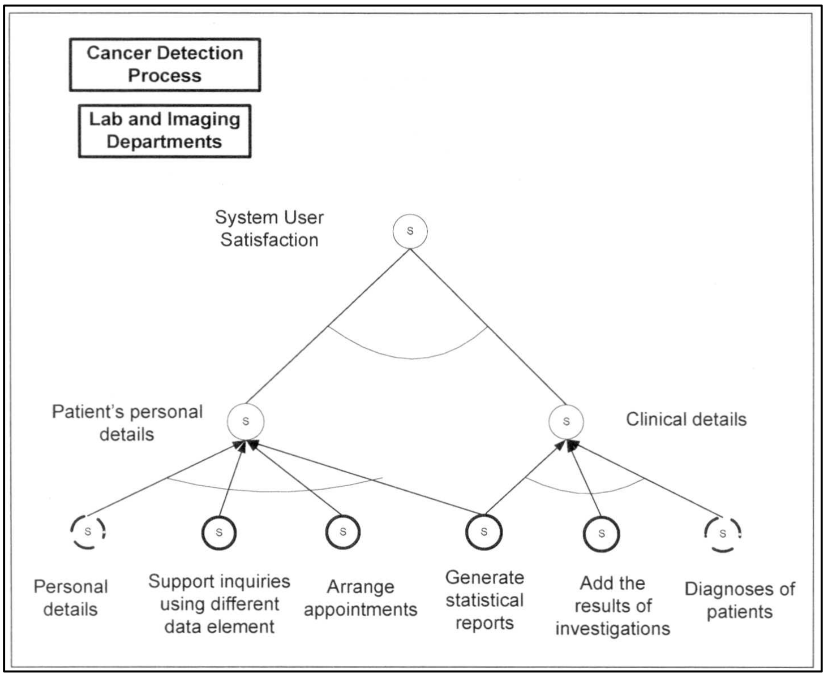


Figure F.10: The System User Satisfaction NFR Framework for the Cancer Detection Process having Ref no.2e [Source: (Aburub,2006), Used with the author's permission].

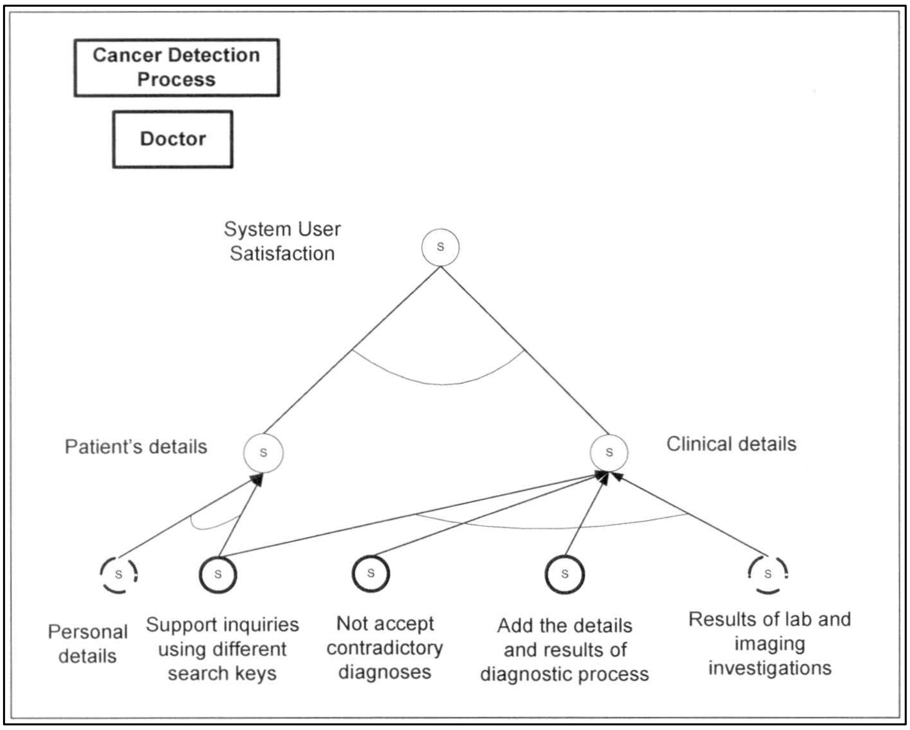


Figure F.11: The System User Satisfaction NFR Framework for the Cancer Detection Process having Ref no.2f [Source: (Aburub,2006), Used with the author's permission].

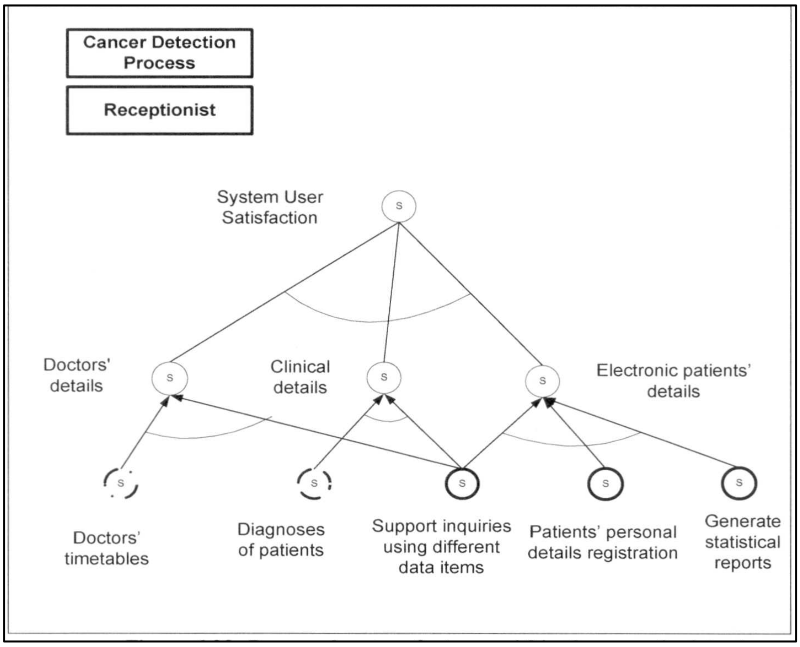


Figure F.12: The System User Satisfaction NFR Framework for the Cancer Detection Process having Ref no.2g [Source: (Aburub,2006), Used with the author's permission].

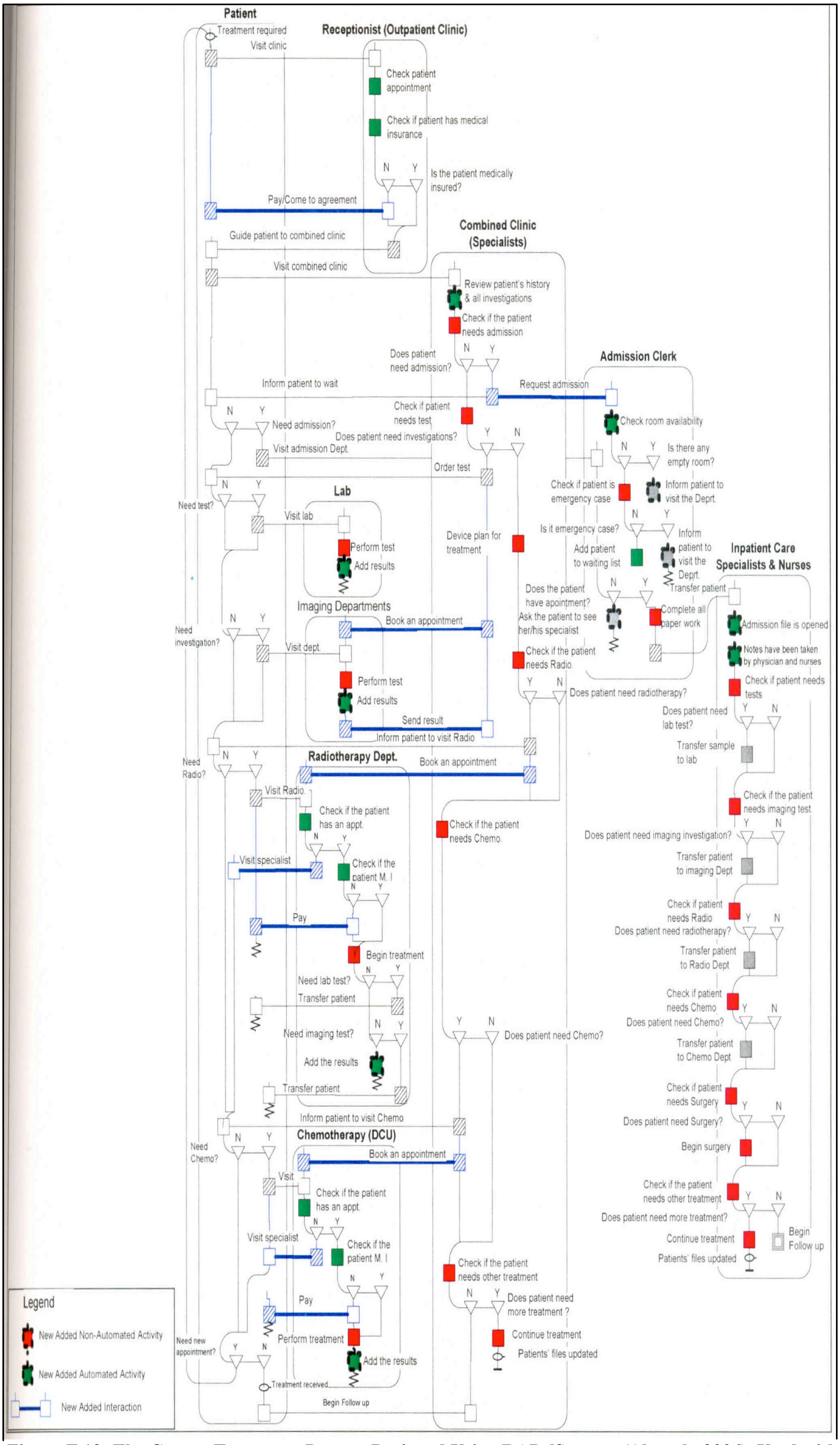


Figure F.13: The Cancer Treatment Process Designed Using RAD [Source: (Aburub, 2006), Used with the author's permission].

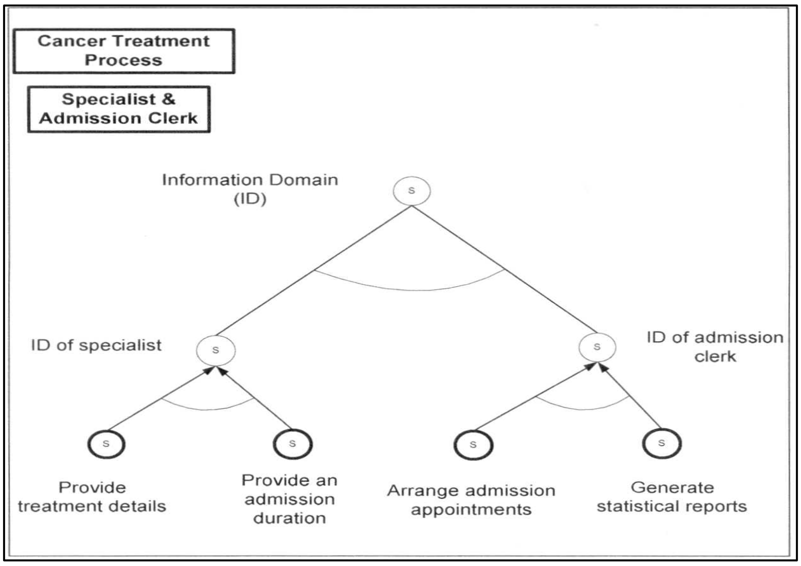


Figure F.14: The Information Domain NFR Framework for the Cancer Treatment Process having Ref no.3b [Source: (Aburub,2006), Used with the author's permission].

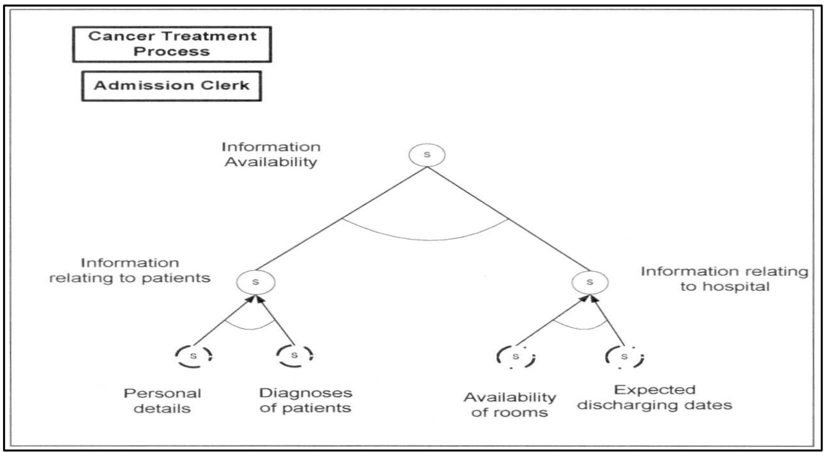


Figure F.15: The Information Availability NFR Framework for the Cancer Treatment Process having Ref no.3c [Source: (Aburub,2006), Used with the author's permission].

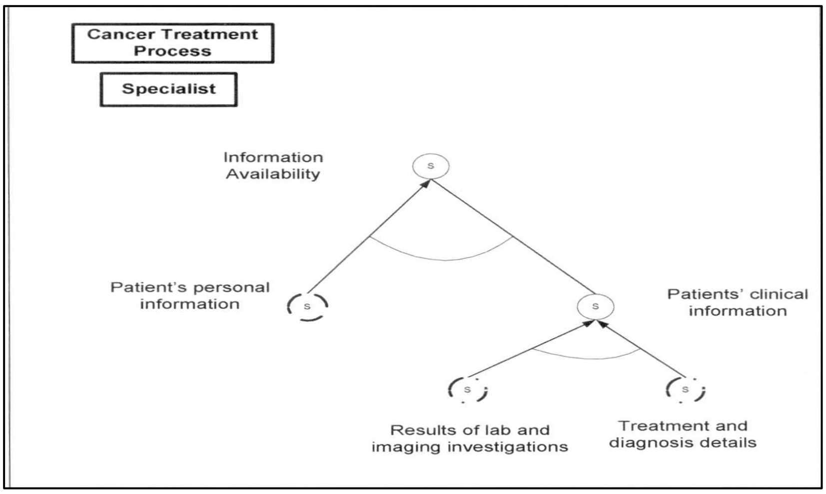


Figure F.16: The Information Availability NFR Framework for the Cancer Treatment Process having Ref no.3d [Source: (Aburub,2006), Used with the author's permission].

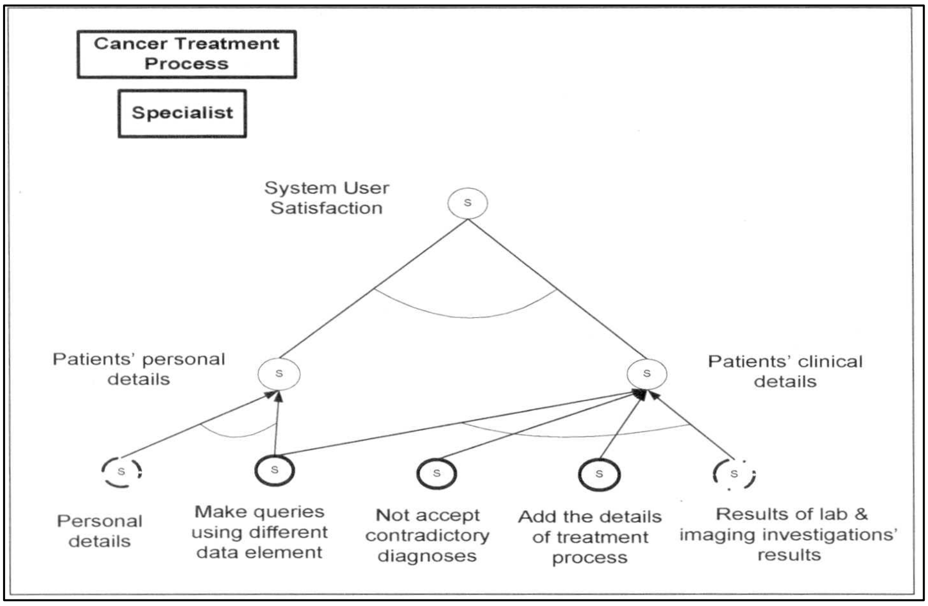


Figure F.17: The System User Satisfaction NFR Framework for the Cancer Treatment Process having Ref no.3e [Source: (Aburub,2006), Used with the author's permission].

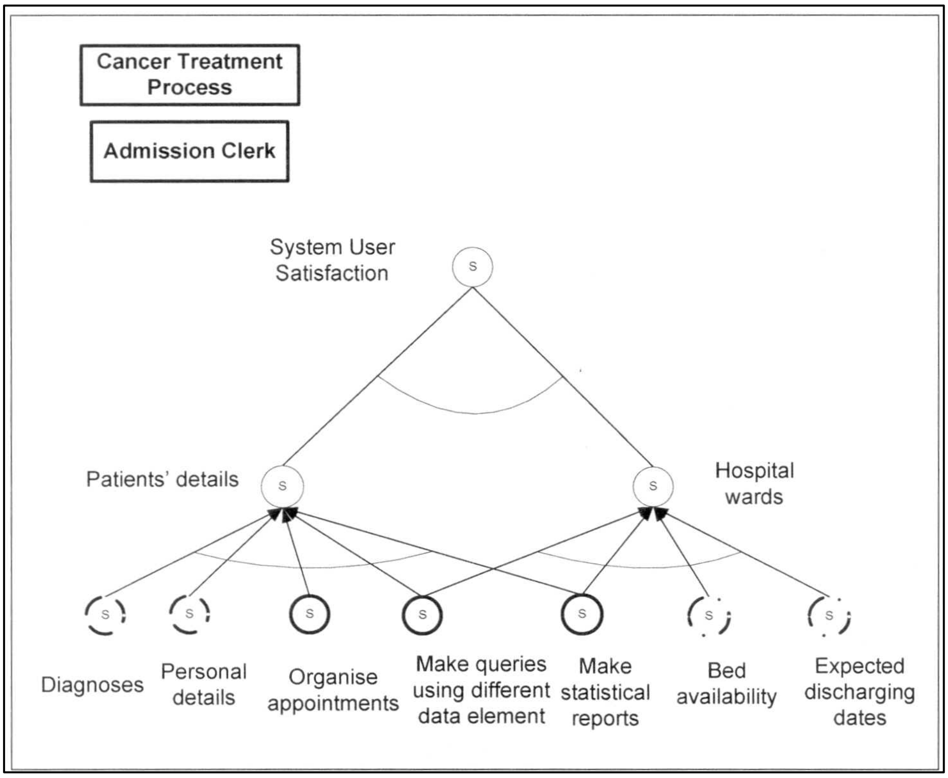


Figure F.18: The System User Satisfaction NFR Framework for the Cancer Treatment Process having Ref no.3f [Source: (Aburub,2006), Used with the author's permission].

Appendix G: Instantiating the GQOnt Using a Pilot Case Study

This appendix contents are designed for the work of Chapter 4 Section 4.4 that concerns with the instantiation of the GQOnt ontology using the patient reception process example, where its models are attached in Appendix F (Section F.1).

G.1 The siGoal Ontology Instantiation Using the Patient Reception Process

The instantiation of the siGoal ontology is carried out with the help of the work attached in Appendix C. Moreover, the GO view of the patient reception process is derived using the algorithms presented in Section 4.3.3.1.

G.1.1 the BS Model of the Patient Reception Process

The BS model for the patient reception process within the KHCC appears in Figure G.1. The model is generated using the algorithm 4.2 "Derive BS model from the G-BPMs". In Figure G.1, the HBG appears as improving the cancer care registration business process. This HBG is represented after a successful matching with at least one goal in the canonical list of goals that is improving the business process (Clements and Bass, 2010). The canonical list goal instances are created in the individuals tab in Protégé as shown in Figure G.2.

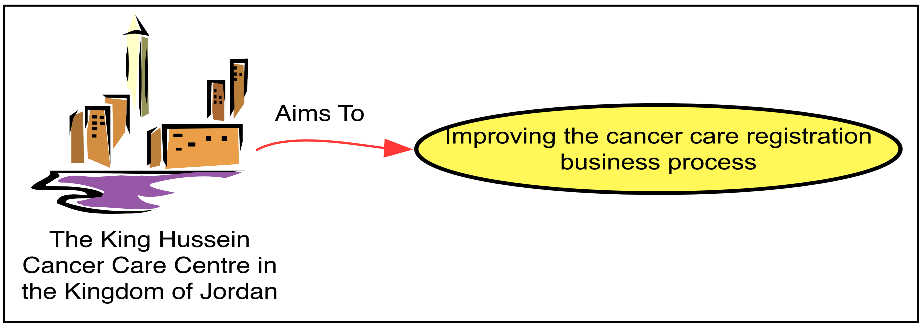


Figure G.1: The BS Model for the Patient Reception Process within the KHCC.

Figure G.3 shows a snapshot from the siGoal ontology individual tab Protégé window while creating the aim to relation instance and its required associated attributes. In Figure G.3, the circled red area acts as an automatic user-friendly template the guide the user's filling with the required information for the created instance or individual (i.e., at1). Figure G.4 shows the enabling and running the Jess engine of the above second SWRL rule.

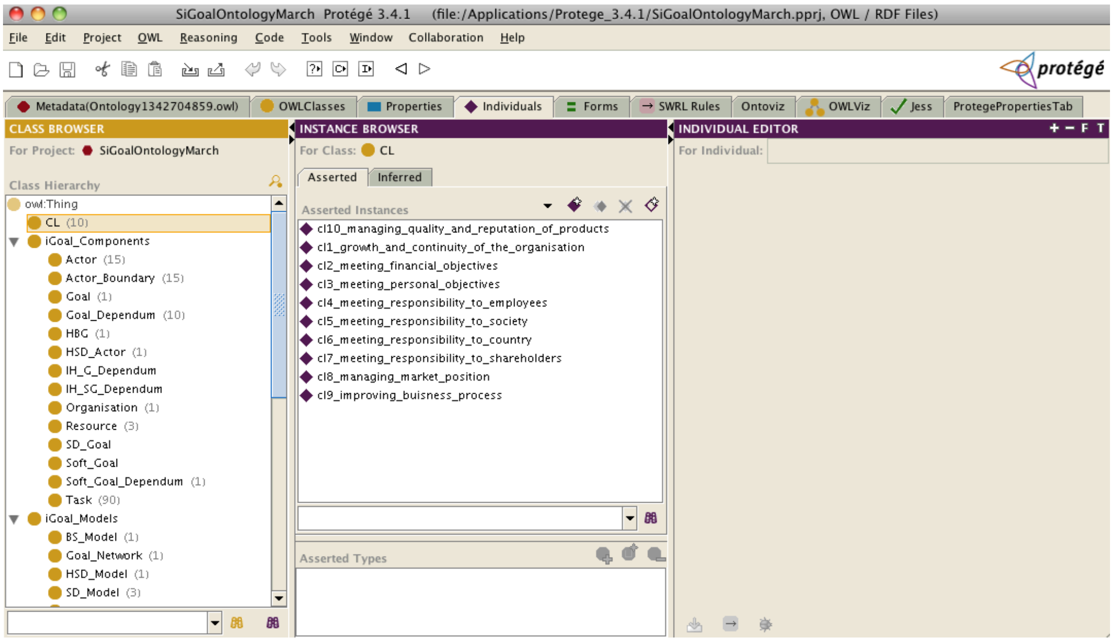


Figure G.2: A Snapshot for the CL Instances Creation within the Individuals Tab in the Protégé Window.

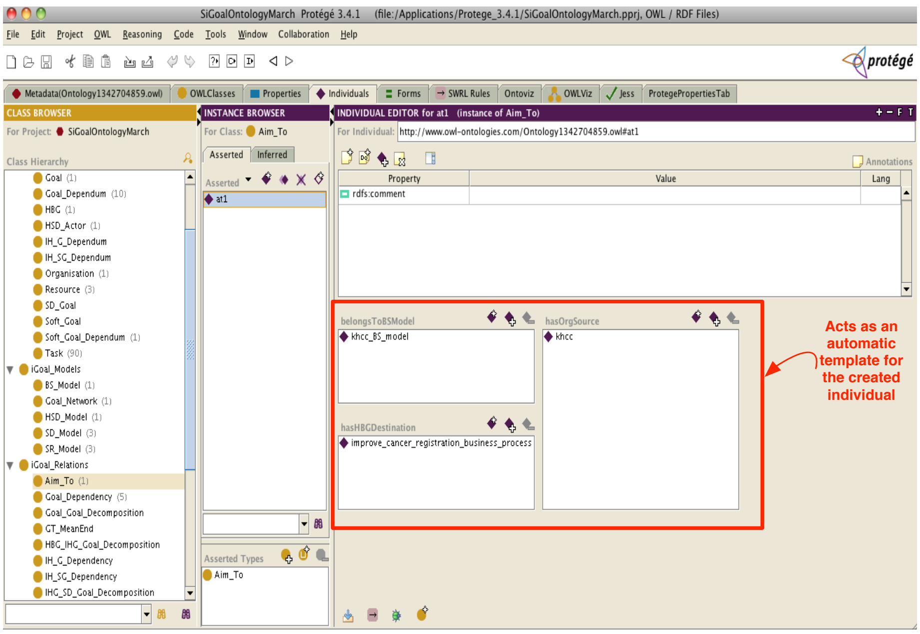


Figure G.3: A Snapshot for the Aim to Relation Class Instance From the Protégé Editor.

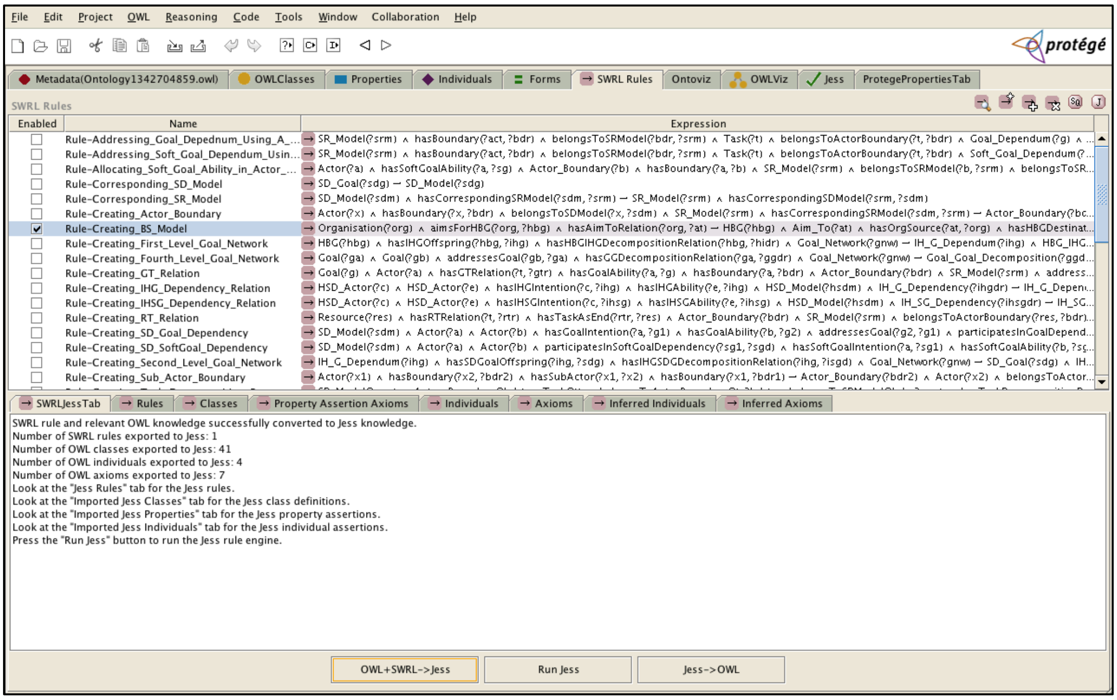


Figure G.4: Snapshot for the SWRL Rule tab within the Protégé Editor.

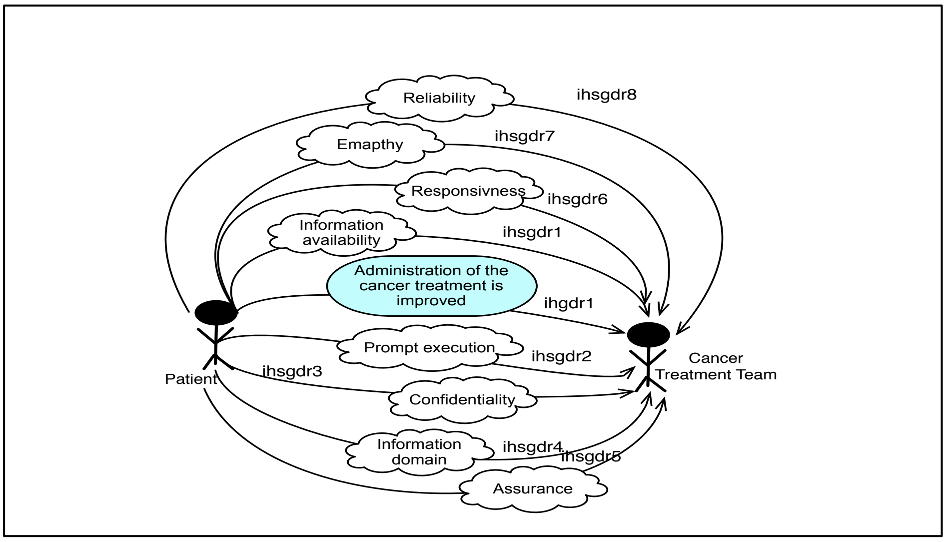


Figure G.5: The HSD Model for the Patient Reception Process (Generated Using Omni Graffle Professional Tool)

G.1.2 the HSD Model for the Patient Reception Process

The HSD model that is related to the patient reception process is generated using the algorithm 4.3 in Section 4.3.3.1 and is shown in Figure G.5. Two HSD actors are identified that are the patient as a key HSD actor, where the cancer treatment team is group of roles. Figure G.5 shows eight IH-SGs and one IH-G along with their dependency relations directed from the patient as a depender to the cancer treatment team as a depndee. The ninth IH-SG that is the system user satisfaction does not participate in any dependency relation however; it must be instantiated in order to derive the further soft goals. The IH-G and the IH-SG dependency relations are created using the two SWRL rules: "Rule-Creating_IHG_Dependency_Relation" and "Rule-Creating_IHSG_Dependency_Relation". A snapshot from the Protégé editor that depicts the instantiation of the IH-SG dependency relations is shown in Figure G.6.

In Figure G.5, the IH-G is elaborated into further sub goals that represent the goals of the SD models where one of the goals of SD model is the patient general reception. This is derived using the object property *hasSubSDGoal*.

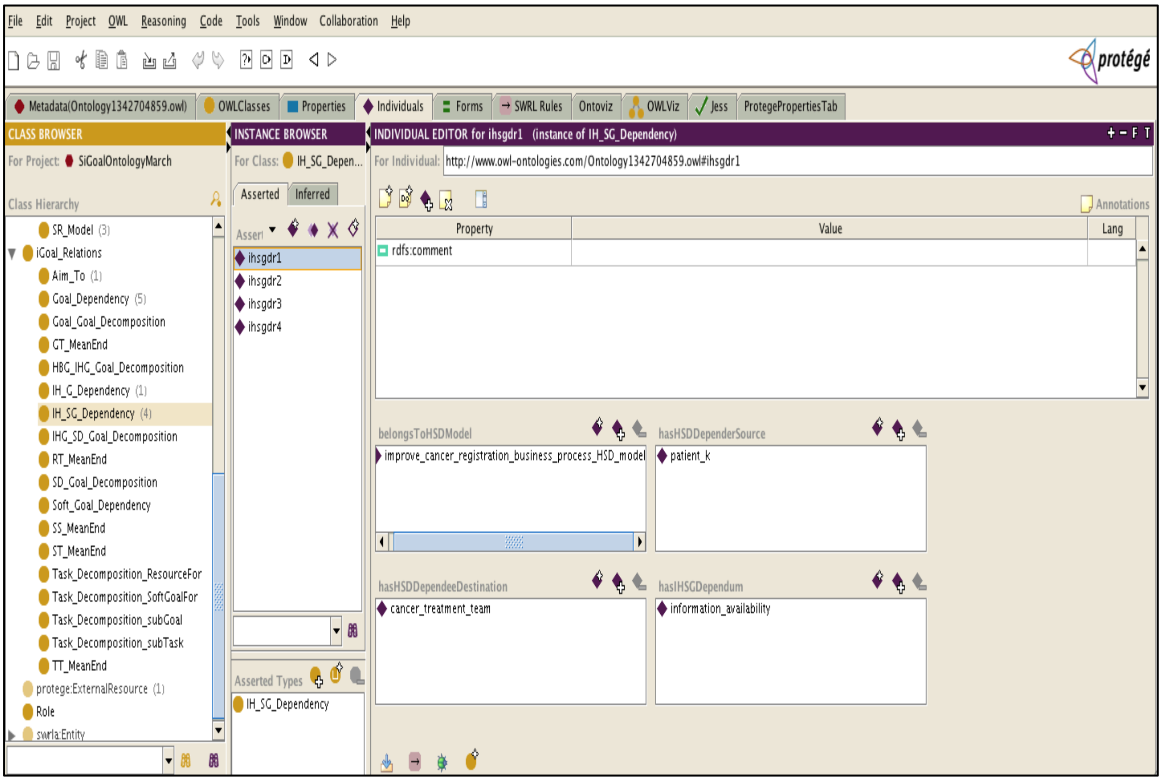


Figure G.6: Example of Creating Instances of the class IH_SG_Dependency and relating them to their HSD Actor Dependers and Depdnees.

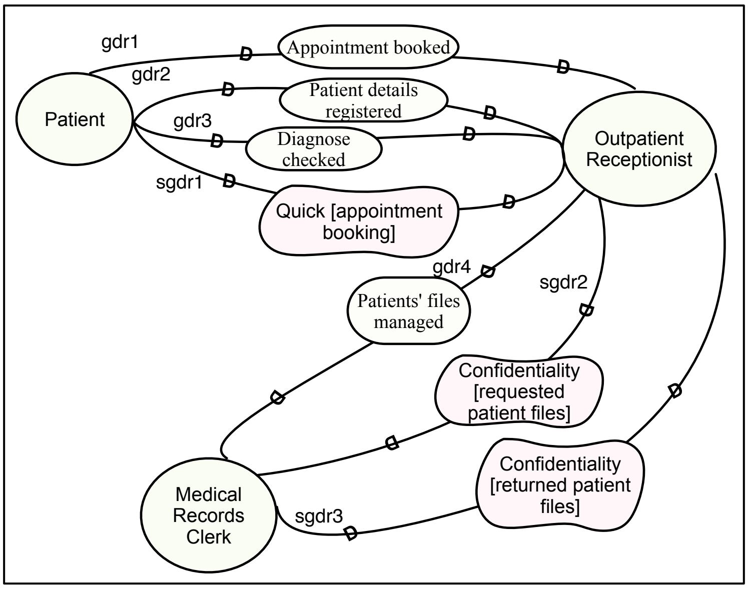


Figure G.7: The SD Model for the General Patient Reception.

G.1.3 the SD Model for the Patient Reception Process

According to the algorithms presented in Section 4.3.3.1, the related algorithm to the SD model comes at the end. However, its order must be after the HSD model in order to represent the interrelation between the GO models. The patient general reception SD model is depicted in Figure G.7. The patient (i.e., key actor), the medical records clerk and the outpatient receptionist actor's goal abilities, soft goal abilities, goal intentions and soft goal intentions are manually instantiated in the Protégé environment. Figure G.8 shows part of the instances creation for goal dependency relatioships for the SD model in Figure G.7.

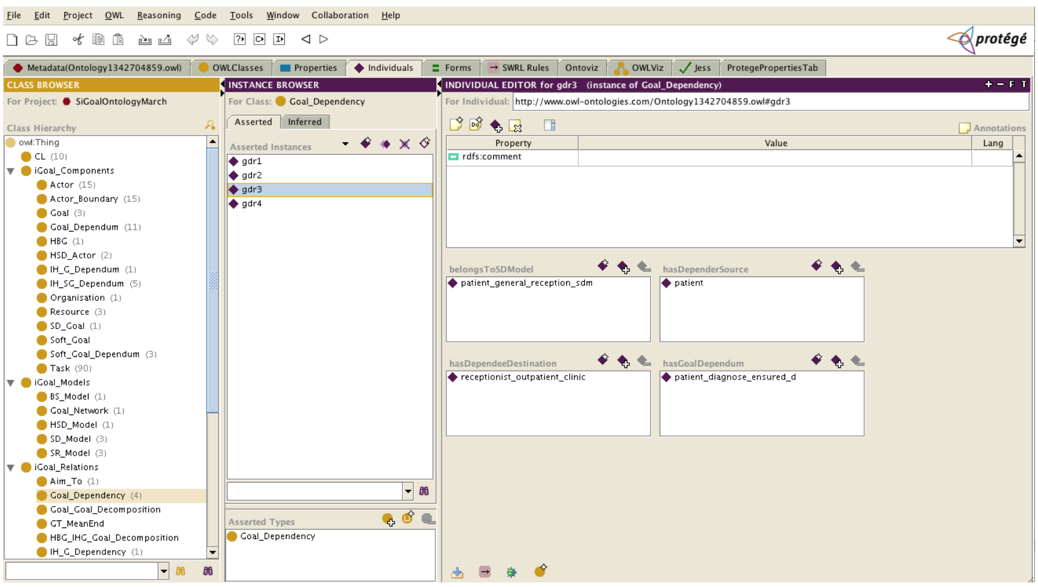


Figure G.8: Example of the Goal Dependency Instantiation for the Patient Reception within siGoal Ontology

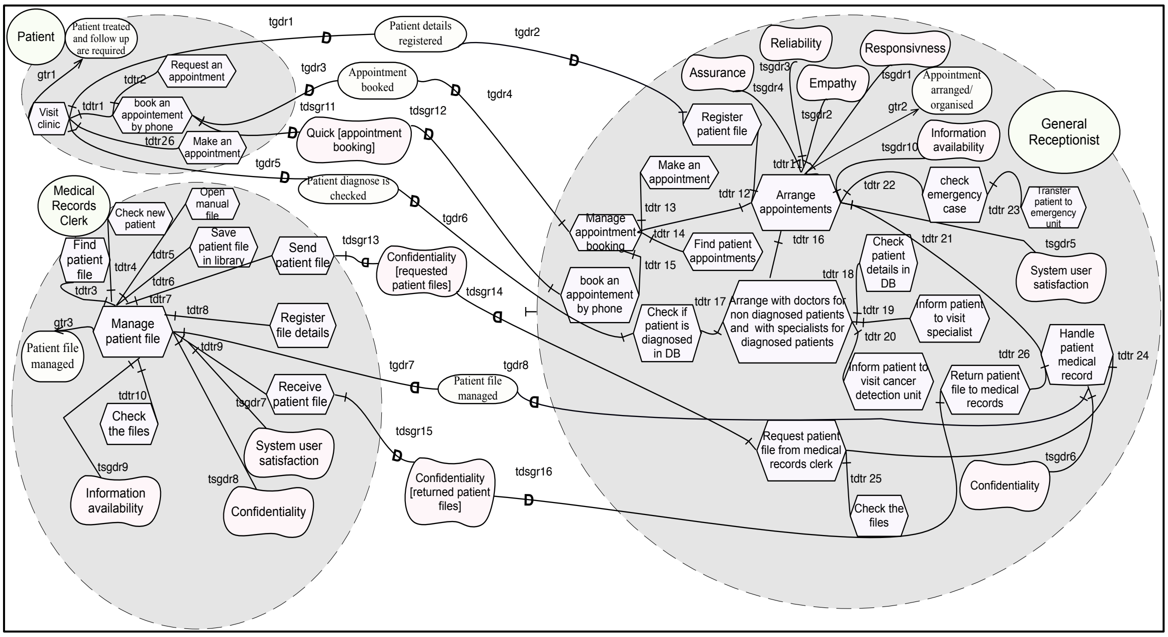


Figure G.9: The instantiated SR model within the GO view for the patent reception BSV (Designed using Omni Graffle Professional Tool).

G.1.4 the SR Model for the Patient Reception Process

This section represents the instantiation of the final and the finest GO model within the GO view for the patient reception process. The SR model of the corresponding general reception SD model is generated using the algorithm 4.5 "Derive the SR Model" presented in Section 4.3.3.1 and it appears in Figure G.9.

Part of the instances, and particularly the ones from the class GT_MeanEnd, in the SR model that appears in Figure G.9 are shown in the captured snapshot from the Protégé editor window in Figure G.10.

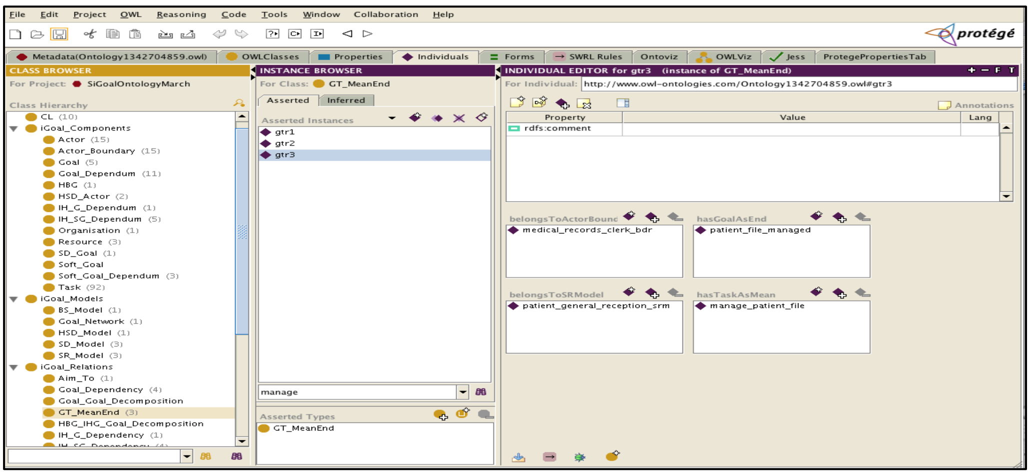


Figure G.10: Example of the instantiation of the GT-Mean end relation class within the siGoal Ontology.

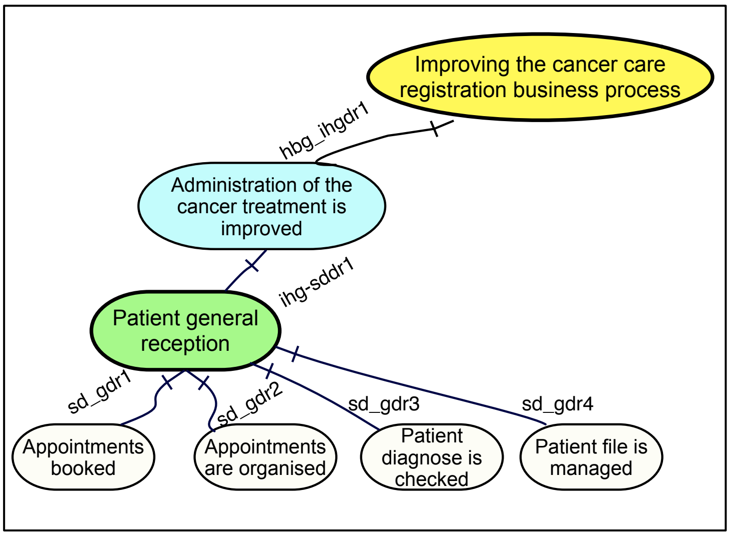


Figure G.11: The GO Network for the Patient Reception Process

G.1.5 The Goal Network for the Patient Reception Process

The GO network related to the patient reception process is illustrated in Figure G.11. The creation of the first-, second-, third- and fourth-levels are carried out using the implemented SWRL as follows where a snapshot for the third level of the GO network from the ontology editor is shown in Figure G.12.

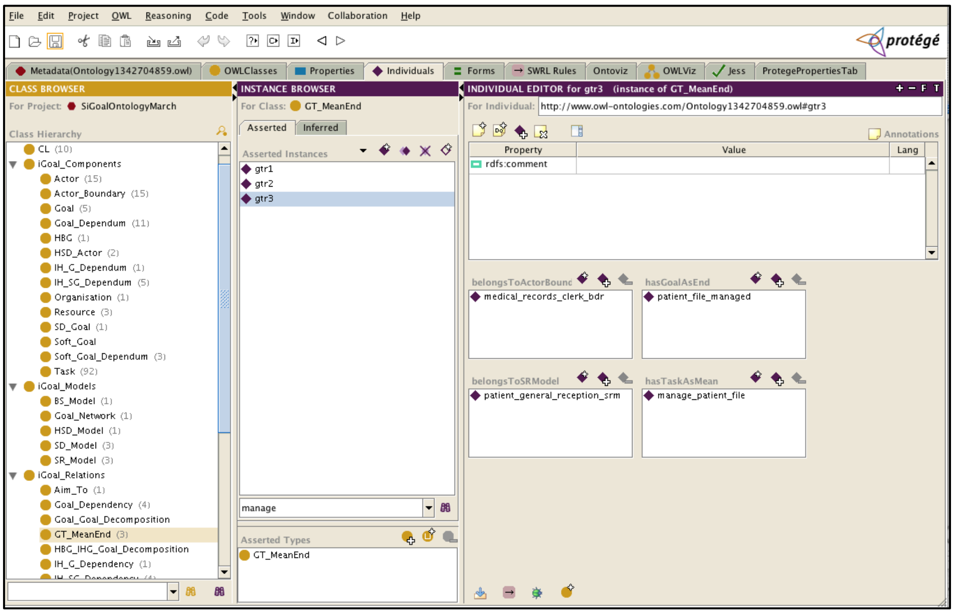


Figure G.12: Example of The correct instantiation of the third level in the GO Network

G.2 The sQuality Ontology Instantiation Using the Patient Reception Process

The sQuality ontology is based on the NFR framework models, where their instantiation is carried out using the work attached in Appendix E. The pilot study NFR framework original models are attached in Appendix F (Section F.1). The confidentiality is one of NFR frameworks for the patient reception process as shown in Figure G.13. A snapshot of a consistent instantiation is shown in Figure G.14.

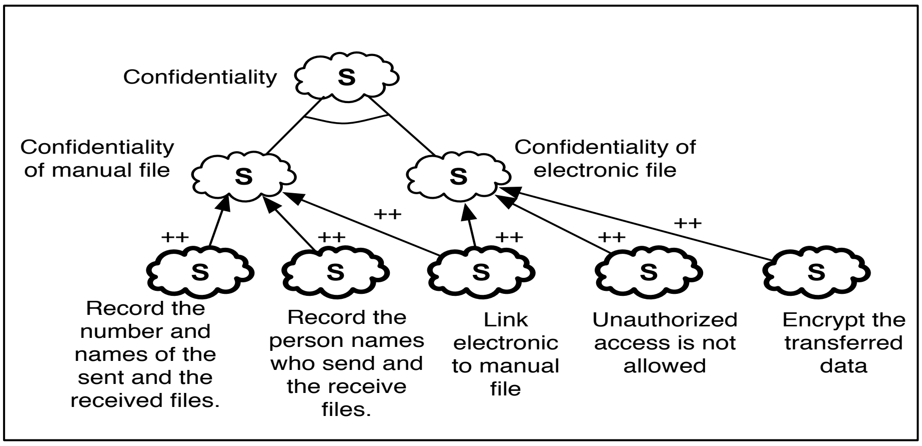


Figure G.14: Representing the Confidentiality NFR Framework for the Entire CCR Process.

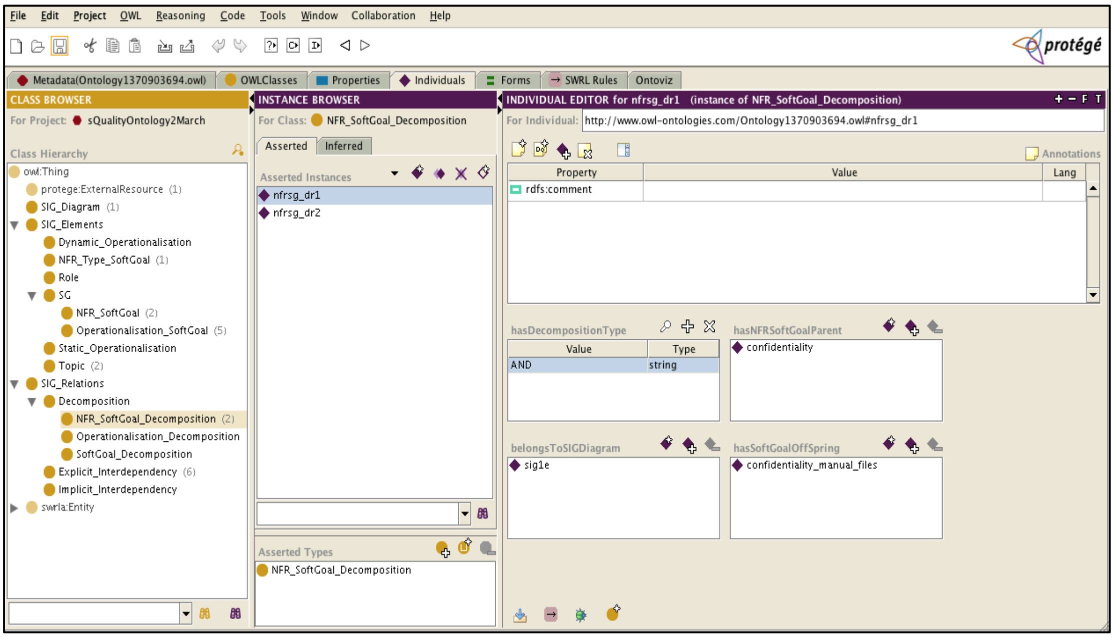


Figure G.14: An Example of the Correct Instantiation of the decomposition relation between NFR Type Soft Goal and NFR Soft Goal

G.3 The siGoal-sQuality Ontology Linker Instantiation Using the Patient Reception Process Pilot Study

Table G.1 shows the linking between the GO view and the Quality view in order to bring the ultimate BSV for the patient reception process. . In this pilot study, the second (i.e., when an operationalization matches with a task in the SR model) and the third (i.e., when a NFR type soft goal is sub from a task in the SR model) linking rules presented in Section 4.3.1.5 have fit with the desired linking. A snapshot from the Protégé Editor that shows how the two ontologies are imported is depicted in Figure G.15. In the figure, the classes from the both ontologies are linked to each others using the proposed linking rules in Section 4.3.5. Figure G.16 illustrates the patient reception process SR model-aided with the NFR framework representation.

Table G.1: Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ↗	**The Linking Rule within the i* Framework**	**NFR type**	**NFR soft goal**	**Operationalization soft goal**
NFR type (Information availability)	Information availability **sub of** arrange appointments			
NFR soft goal		Information related to patients **is sub of** information availability		
Operationalization soft goal (<u>S</u>tatic: data **/<u>D</u>ynamic:** function**)**			Personal details (<u>S</u>) and 2- diagnoses details (<u>S</u>) **contribute to** information related to patients	
			Specialists timetables (<u>S</u>) **contributes to** information related to specialists	
NFR type (Information domain)				
NFR soft goal				

Table G.1 (Cont'd): Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ↗	**The Linking Rule within the i* Framework**	**NFR type**	**NFR soft goal**	**Operationalization soft goal**
Operationalisation soft goal (Static: data /**Dynamic:** function**)**	Make an appointment (**D**) **is sub of** manage booking	- Make an appointment (**D**), 2- create e-file (**D**), 3- register patient personal details (**D**) and 4- generate statistical reports (**D**) **contribute to** information domain of receptionist		
	Register patient personal details (**D**) **is sub of** manage booking			
NFR type (System user satisfaction)	System user satisfaction **sub of** arrange appointments			
NFR soft goal		Doctor timetables **is sub of** system user satisfaction		
		Patient details **is sub of** system user satisfaction		
		Clinical details **is sub of** system user satisfaction		
Operationalisation soft goal (Static: data /**Dynamic:** function**)**			1-Doctor timetables (**S**) and 2- support inquiries using different search keys (**D**) **contribute to** doctor timetables	
			- Support inquiries using different search keys (**D**), 2- register personal details (**D**), 3- help in generating statistical reports (**D**) **contribute to** patients' details	

Table G.1 (Cont'd): Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ↗	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
Operationalisation soft goal (Static: data **/D**ynamic: function**)**			1- Help in generating statistical reports (**D**) and 2-diagnoses of patient (**S**) **contribute to** clinical details	
NFR type (Confidentiality)	Confidentiality of patients' files **is sub of** handle patients' medical records			
NFR soft goal		Confidentiality [manual files] is **sub of** confidentiality NFR type Confidentiality [electronic files] is **sub of** confidentiality NFR type		
Operationalisation soft goal (Static: data **/D**ynamic: function**)**			1- Number and names of sent & received files must recorded, 2- names of persons who sent receive files recorded and 3-link electronic to manual files **contribute to** Confidentiality [manual files]	
			1-Link electronic to manual files, 2-unauthorised access is not allowed and 3-transferred data must be encrypted **contribute to** Confidentiality [electronic files]	

Table G.1 (Cont'd): Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ➚	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
NFR type (Information availability)	Information availability **sub of** manage patients' files			
NFR soft goal		Information related to patients **is sub of** information availability Information related to specialists **is sub of** information availability		
Operationalisation soft goal (<u>S</u>tatic: data /<u>**D**</u>ynamic: function**)**			1- Personal details (<u>S</u>) and 2- diagnoses details (<u>S</u>) **contribute to** information related to patients Specialists timetables (<u>S</u>) **contributes to** information related to specialists	

Table G.1 (Cont'd): Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ↗	**The Linking Rule within the i* Framework**	**NFR type**	**NFR soft goal**	**Operationalization soft goal**
NFR type (Information domain)				
NFR soft goal				
Operationalisation soft goal (<u>S</u>tatic: data /**<u>D</u>ynamic:** function**)**	Open manual file **is sub of** manage patients' files	Open manual file **contributes to** information domain.		
NFR type (System user satisfaction)	System user satisfaction **sub of** mange patients' files			
NFR soft goal		Doctor timetables **is sub of** system user satisfaction		
		Patient details **is sub of** system user satisfaction		
		Clinical details **is sub of** system user satisfaction		

Table G.1 (Cont'd): Linking the sQuality Ontology into the siGoal Ontology using the Patient Reception Process

Optional] Sub of/ Contributes ➚	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
Operationalisation soft goal (<u>S</u>**tatic:** data /<u>D</u>**ynamic:** function**)**			1-Doctor timetables (<u>S</u>) and 2- support inquiries using different search keys (<u>D</u>) **contribute to** doctor timetables	
			1- Support inquiries using different search keys (<u>D</u>), 2- register personal details (<u>D</u>), 3- help in generating statistical reports (<u>D</u>) **contribute to** patients' details	
			1- Help in generating statistical reports (<u>D</u>) and 2-diagnoses of patient (<u>S</u>) **contribute to** clinical details	

[Optional] Sub of/ Contributes ➚	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
NFR type (Confidentiality)	Confidentiality of patients' files **is sub of** manage patients' files			
NFR soft goal		Confidentiality [manual files] is **sub of** confidentiality NFR type		
		Confidentiality [electronic files] is **sub of** confidentiality NFR type		

[Optional] Sub of/ Contributes ➚	The Linking Rule within the i* Framework	NFR type	NFR soft goal	Operationalization soft goal
Operationalisation soft goal (<u>S</u>tatic: data /<u>D</u>ynamic: function**)**			1- Number and names of sent & received files must recorded, 2- names of persons who sent receive files recorded and 3-link electronic to manual files **contribute to** Confidentiality [manual files]	
			1-Link electronic to manual files, 2- unauthorized access is not allowed and 3- transferred data must be encrypted **contribute to** Confidentiality [electronic files]	

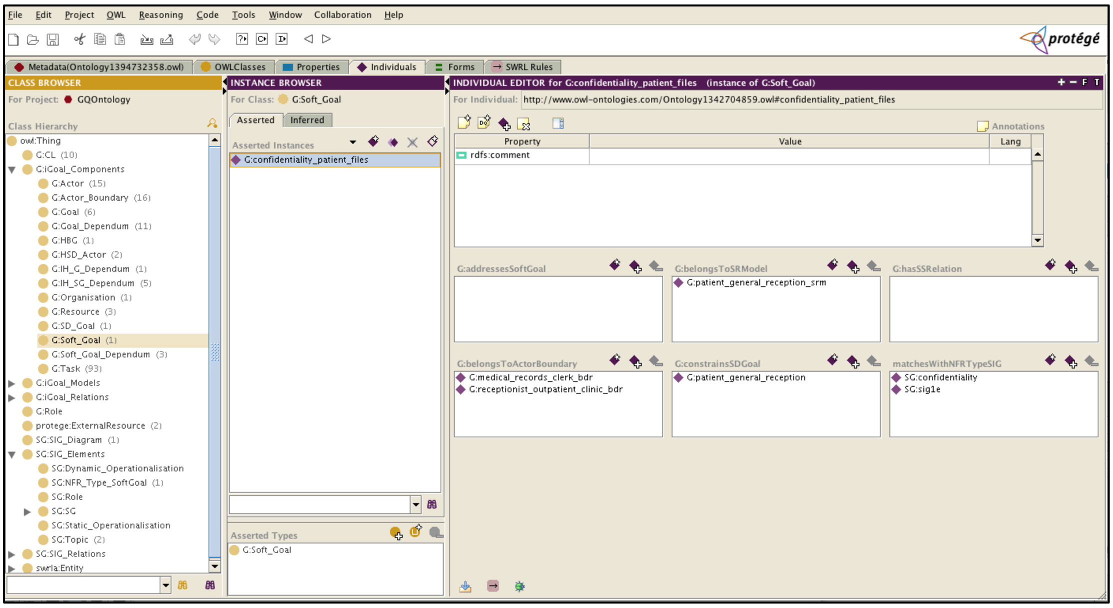


Figure G.15: Example from the GQOnt Instance Properties for the Patient Reception Process

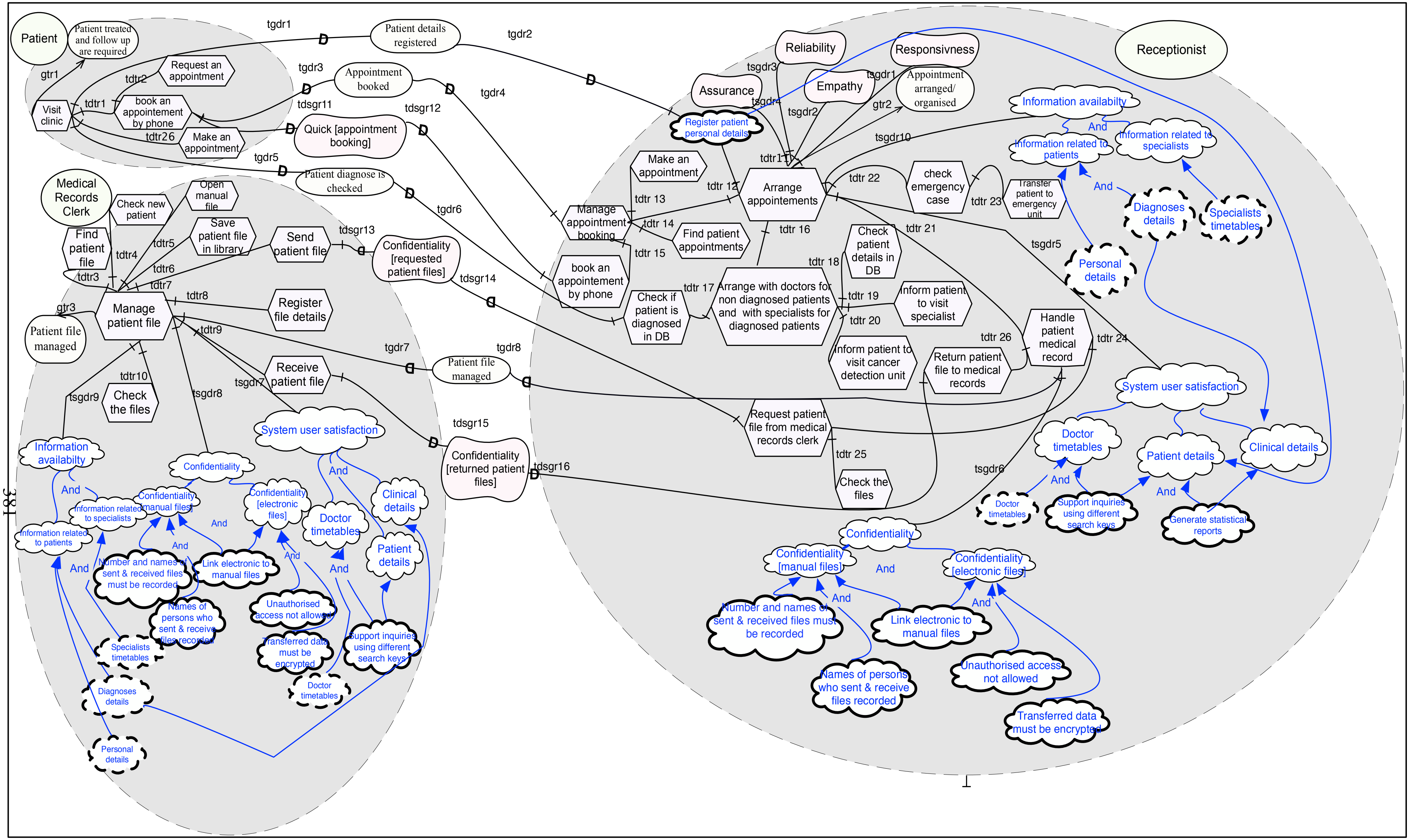


Figure G.16: The linking between the GO models and the Quality-Oriented models using the Patient Reception Example.

Appendix H: The GQ-srBPA Ontology Classes and Properties

This appendix presents OWL-DL classes ad properties that set the GQ-srBPA ontology. In Table H.1, the orange shaded cells are adapted from (Yousef, 2010). The white shaded areas are the new refinements integrated into the original srBPA ontology in order to develop the GQ-srBPAOnt. The names of the OWL-DL classes are refined to some extent using yellow shading however; the associated object properties are mostly remained.

Table H.1: The Classes and Properties of the GQ-srBPAOnt [Source: (Yousef,2010), Adapted with the author's permission].

Concept	Description	Attributes
G_EBE	The Essential Business Entities of an enterprise.	1) isConsideredUOW: Boolean.
EBQ	The Essential business Qualities of an enterprise.	
GQ_UOW_Diagram	The units of work diagram according to the Riva method.	1) hasUOW of type G_UOW, and 2) hasOutsideWorld of type Outside_world.
PA_1st_Cut_Diagram	The 1st cut process architecture diagram according to the Riva method.	1) hasCP of type CP, 2) hasCMP of type CMP, and 3) hasOutsideWorld of type Outside_world.
GQ_PA_2nd_Cut_Diagram	The 2nd cut process architecture diagram according to the Riva method.	1) hasCP of type CP, 2) hasCMP of type CMP, and 3) hasOutsideWorld of type Outside_world.
G_UOW	The units of work in the UOW diagram, according to the Riva method.	1) BelongsToUOWDiagram of type GQ_UOW_Diagram, 2) hasDesiredEssentialQuality of type EBQ, 3) hasDesiredQualityonUoW of type Q_UOW, 4) hasCorrespondingCP of type CP, and 5) hasGenerateRelation of type Generate.

Table H.1 (Cont'd): The Classes and Properties of the GQ-srBPAOnt [Source: (Yousef,2010), Adapted with the author's permission].

Concept	Description	Attributes
Q_UOW	The quality requirement on the UoW.	1) constrainsUoW of type G_UOW, 1) belongsToUoWDiagram of type GQ_UoW_Diagram, 2) isElaborative: Boolean, 3) hasConstrainRelation of type Constrain, and 4) hasQualityReference of type Quality_Model_Reference
Quality_Model_Reference	The reference for a quality model for a particular elaborative Q-UoW.	1) belongsToQUoW of type Q_UOW
CP	The case processes in the 1st cut and 2nd cut PA diagrams, according to the Riva method.	1) BelongsTo1stCutDiagram of type PA_1st_Diagram, 2) BelongsTo2ndCutDiagram of type GQ_PA_2nd_Diagram, 3) hasCorrespondingUOW of type G_UOW, 4) hasRequestRelation of type Rrequest, 5) hasDeliverRelation of type Deliver, 6) hasStartRelation of type Start, 7) hasGoP of type GOP, and 8) hasQoP only QOP. 9) hasQualityReference of type Quality_Model_Reference
CMP	The case management process in the 1st cut and 2nd cut PA diagrams, according to the Riva method.	1) BelongsTo1stCutDiagram of type PA_1st_Diagram, 2) BelongsTo2ndCutDiagram of type GQ_PA_2nd_Diagram, 3) hasManagingCP of type CP, 4) hasStartRelation of type Start, 5) isActive of type Boolean, and 6) hasGoP of type GOP, and 7) hasQoP only QOP. 8) hasQualityReference of type Quality_Model_Reference
GOP	The goal of a process in the GQ 2nd cut architecture, according to the refined Riva method.	1) belongsTo2ndCutDiagram of type GQ_PA_2nd_cut_Diagram , 2) belongsToProcess of type CP and 3) belongsToProcess of type CMP

Table H.1 (Cont'd): The Classes and Properties of the GQ-srBPAOnt [Source: (Yousef,2010), Adapted with the author's permission]

Concept	Description	Attributes
QOP	The quality of the process in the GQ 2nd cut architecture, according to the refined Riva method.	1) belongsTo2ndCutDiagram of type GQ_PA_2nd_cut_Diagram, 2) isElaborative: Boolean, 3) hasQualityReference of type Quality_Model_Reference, and 4) constrainsProcess of type (CP or CMP)
Outside_World	The outside world in the UOW, 1st cut and 2nd cut PA diagrams, according to the Riva method.	1) hasOutsideWorldRelation of type Outside_relation, 2) BelongsToUOWDiagram of type GQ_UOW_Diagram, 3) BelongsTo1stCutDiagram of type PA_1st_Diagram, and 4) BelongsTo2ndCutDiagram of type GQ_PA_2nd_Diagram.
Generate	The generate relationship in the UOW diagram between UOW class members.	1) hasUOWSource of type G_UOW, 2) hasUOWDestinaiton of type G_UOW, and 3) belongsToUOWDiagram of type GQ_UOW_Diagram.
Request	The relationship in the PA diagram between members of the CP and the CMP classes.	1) hasCPSource of type CP, 2) hasCPDestination of type CP, 3) hasCMPDestinaiton of type CMP, 4) isActive of type Boolean, 5) belongsToPA1Diagram of type PA_1st_cut_diagram, and 6) belongsToPA2Diagram of type GQ_PA_2nd_cut_diagram.
Deliver	The deliver relationship in the PA diagrams between the CP class members.	1) hasCPSource of type CP, 2) hasCPDestinaiton of type CP, 3) isActive of type Boolean, 4) belongsToPA1Diagram of type PA_1st_cut_diagram, and 5) belongsToPA2Diagram of type GQ_PA_2nd_cut_diagram.
Start	The start relationship in the PA diagrams between members of the CP and the CMP classes.	1) hasCMPSource of type CMP, 2) hasCPSource of type CP, 3) hasCPDestinaiton of type CP, 4) isActive of type Boolean, 5) belongsToPA1Diagram of type PA_1st_cut_diagram, and 6) belongsToPA2Diagram of type GQ_PA_2nd_cut_diagram.

Table H.1 (Cont'd): The Classes and Properties of the GQ-srBPAOnt [Source: (Yousef,2010), Adapted with the author's permission]

Concept	Description	Attributes
Outside_Relation	The relation from the outside world to a member of the UOW, CP or CMP classes.	1) hasOutsideWorldSource of type outside_world, 2) hasUOWDestination of type G_UOW, 3) hasCPDestination of type CP, 4) hasCMPDestination of type CMP, 5) isActive of type Boolean, 6) belongsToPA1Diagram of type PA_1st_cut_diagram, 7) belongsToPA2Diagram of type GQ_PA_2nd_cut_diagram, and 8) belongsToUOWDiagram of type GQ_UOW_Diagram.
Constrain	The "constrains" relation from a member of the Q-UoW to a member of the G-UoW.	1) belongsToUoWDiagram of type GQ_UoW_Diagram, 2) hasQUoWSource of type Q_UOW, and 3) hasUoWDestination of type G_UOW.

Appendix I: Ontologising the GQ-Riva Method Steps

This appendix aims at presenting the GQ-Riva method steps that are ontologised using the OWL-DL and SWRL rules. Table I.1 is borrowed from (Yousef, 2010). However, the table has been extended with respect to the refined Riva method in Section 5.5. In Table I.1, the text in red represents the refinements carried out on the original work represented in black text.

Table I.1: The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

Riva Step (Refined)	**Steps 1**: Agree the boundary of the organisation. Brainstorm the organisations' subject matter to identify Essential Business Entities (EBEs) and Essential Business Qualities (EBQs).
Related Classes	**G_EBE, EBQ**
Class Properties	-
Class Instances	EBE class instances represent all EBEs of an organisation. In addition, the EBQ class instances represent the entire main desired quality requirements with regard to the agreed boundary of the organisation. In case we are developing the ontology instance of an organisation, EBE and the EBQ class instances that make the main input, and should be entered after identifying them.
OWL restrictions	-
SWRL Rules	-
Riva Step (Refined)	**Step 2 (a)**: Classify G_EBEs that have a lifetime which is handled by, or the responsibility of, members of the organisation as Goal-based Units of Work (G_UOWs)
Related Classes	**G_EBE**, **G_UOW**
Class Properties	**isConsideredUOW**: is set to True for an EBE instance if it can be considered a UOW.
Class Instances	Instances of class UOW represent the Units of work identified from the set of EBEs. No new instances should be created because UOWs are actually reclassification of EBE instances whose isConsideredUOW property value is True. The UOW class instances can be automatically instantiated using the SWRL rule: Rule_UOW_Instances.
OWL restrictions	-

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

SWRL Rules	**Rule_UOW_Instances**: *EBE(?x) ^ isConsideredUOW(?x, True) → UOW(?x)*
Riva Step (Refined)	**Step 2 (b)**: Elaborate the EBQs that constrain each particular identified G_UoW, into Quality of the Unit of Works (Q-UOWs).
Related Classes	**EBQs, Q_UOW, G_UOW,**
Class Properties	**hasDesiredEssentialQuality**: to relate the G_UOW with its essential business quality requirements. **constrainsUoW**: to relate the Q_UOW with its relevant G_UOW.
Class Instances	Q_UOW class instances that are created as desired quality attributes or requirements with a lifetime for their corresponding G_UOW member.
OWL Restrictions	**Q_UOW: ∀** *constrainsUoW only G_UOW*. **G_UOW: ∀** *hasDesiredEssentialQuality only EBQ*.
SWRL Rules	**Rule_Q_UoW_Instances:** EBQ(?ebq) ∧ G_UOW(?gu) ∧ hasDesiredEssentialQuality(?gu, ?ebq) → Q_UOW(?ebq) ∧ constrainsUoW(?ebq, ?gu)
Riva Step (Refined)	**Step 2** ©: Classify the Q-UoWs into elaborative and non-elaborative ones. Identify the Reference of the Quality Model that represents an elaborative Q-UOW where the non-elaborative one has no reference for its quality requirement.
Related Classes	**EBQs, Q_UOW**
Class Properties	**isElaborative:** is set true when the quality requirement on the UoW has associated quality model that represent the requirement. **hasQualityReference**: to set the reference of the quality model for the identified Q_UoW. **belongsToQUoW**: for a particular elaborative Q-UOW, the identified quality model reference instances that belong to it.
Class Instances	Q_UOW instances that are considered as elaborative and having associated quality reference model.
OWL Restrictions	**Q_UOW: ∀** *hasQualityReference only Quality_Model_Reference.*

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

SWRL Rules	**Rule_hasAssociatedQualityModelReference**: *Q_UOW(?qu) ∧isElaborative(?qu, true) ∧ hasQualityReference(?qu, ?qr) → Quality_Model_Reference(?qr) ∧ belongsToQUoW(?qr, ?qu)*
Riva Step (Refined)	**Step 3**: Draw a GQ-UOW diagram that shows just dynamic functional relationships between G-UOWs and the "constrain" quality relationships between the Q_UOWs and the G_UOWs.
Class Properties	**hasUOWSource**, **hasUOWDestination, hasOutsideworldSource and hasOutsidewoldDestinaiton**: to set the sources and destinations for each relation in the GQ UOW-Diagram. **belongsToUOWDiagram**: for a certain GQ UOWDiagram, values of this property can be set automatically using a rule executed using the JessTab in Protégé. **hasGenerateRelation**: Any G-UOW that generates another unit of work, will have this property set to that relation. **hasConstrainRelation**: Regardless to the elaboration possibility, any Q-UOW that constrain associated G-UOW must have their property. **constrainsUoW:** to set what Q-UOW that constrains a G-UOW. **hasQUoWSource** and **hasUOWDestination:** to set the source and the destination for the constrain relation between Q_UOW and G_UOW.
Class Instances	Instances of the classes Generate and Outside_relation together with the class properties that are related to the sources and destinations represent the relationships between members of UOW class and between members of UOW and Outside_world classes. In addition, instances from the class Constrain along with their associated object properties that are related with members of the source and destination.

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

Jess usage (in case an ontology instance for an organisation is to be developed)	Values of the property hasUOWRelation can be set programmatically using a rule executed using the JessTab.
OWL restrictions	G_UOW: ∀ *BelongsToUoWDiagram only GQ_UOW_Diagram* ∀ *hasGenerateRelation only Generate* Q_UOW*:* ∀ *belongsToUoWDiagram only GQ_UoW_Diagram* ∀ *constrainsUoW only G_UOW* ∀ *hasConstrainRelation only Constrain.* Constrain: ∀ *belongsToUoWDiagram only GQ_UoW_Diagram.* ∀ *hasQUoWSource only Q_UOW* ∀ *hasUoWDestination only G_UOW*
SWRL Rules	**Rule_hasGenerateRelation**.: *UOW (?u) ^ hasGenerateRelation (?u, ?g) →Generate(?g) ^ hasUOWSource (?g, ?u)* **Rule_hasConstrainRelation***:* *Q_UOW(?qu) ∧ G_UOW(?u) ∧ constrainsUoW(?qu, ?u) ∧ hasConstrainRelation(?qu, ?cr) ∧ belongsToUoWDiagram(?u, ?uowm) → Constrain(?cr) ∧ hasQUoWSource(?cr, ?qu) ∧ hasUoWDestination(?cr, ?u) ∧ belongsToUoWDiagram(?cr, ?uowm) ∧ belongsToUoWDiagram(?qu, ?uowm)*
Riva Step	**Step 4**: Assume for each UOW that there will be: A case process that handles single instances of the UOW; and A case management process for dealing with the flow of instances.
Related Classes	**UOW, CP, CMP**
Class Properties	**hasCorrespondingCP, hasCorrespondingUOW**: to correspond each UOW to a CP, and vice versa. **hasManagingCP**: to correspond each CMP to a CP.
Class Instances	CP and CMP Instances represent the case processes and case management processes.

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

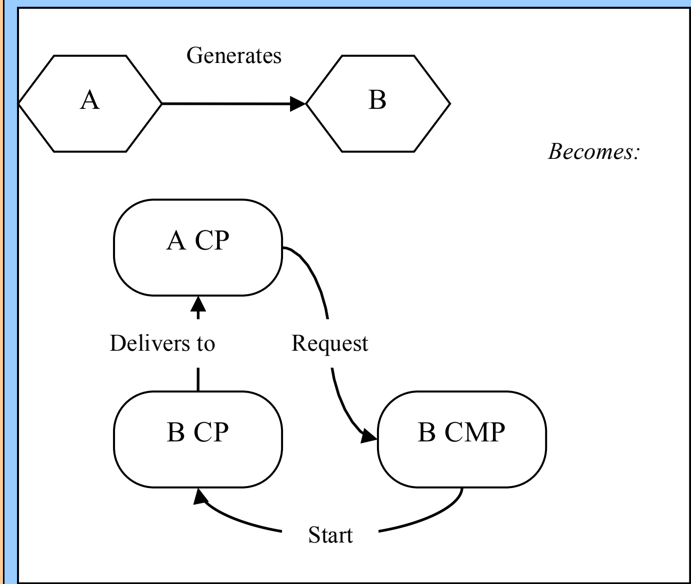
Jess usage (in case an ontology instance for an organisation is to be developed)	CP and CMP instances can be created programmatically, together with the properties hasCorrespondingCP, hasCorrespondingUOW and hasManagingCP using the JessTab.
OWL restrictions	UOW: ∀ *hasCorrespondingCP only CP* CP: ∀ *hasCorrespondingUOW only UOW* CMP: ∀ *hasManagingCP only CP*
SWRL Rules	**Rule_hasCorrespondingElement**: *hasCorrespondingCP(?x,?y) → hasCorrespondingUOW(?y, ?x)*
Riva Step	**Step 5**: Transform the UOW diagram into a first-cut process architecture. This step is entirely mechanical and is accomplished by hypothesising that each 'generates relationship' between two UOWs in the UOW diagram can be translated into relationships between two corresponding processes, for example A and B: 
Related Classes	**UOW, CP, CMP, PA_1st_Diagram, Generate, Deliver, Request, Start**
Class Properties	**belongsTo1stCutDiagram** **hasDeliverRelation. hasRequestRelation, hasStartRelation.**
Class Instances	**PA_1st_Diagram instance represent the specific 1st cut BPA diagram for an organisation.**

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

Jess usage (in case an ontology instance for an organisation is to be developed)	**JessTab can be used to set values of belongsTo1stCutDiagram for both CP and CMP to the PA_1st_Diagram Instance.**
OWL restrictions	**CP: ∀ BelongsTo1stCutDiagram only PA_1st_Diagram** **CMP: ∀ BelongsTo1stCutDiagram only PA_1st_Diagram**
SWRL Rules	**Rule_1st_cut_translated_relations:** **UOW(?a) ^ UOW(?b) ^ Generate(?g) ^ hasUOWSource(?g, ?a) ^ hasUOWDestination(?g, ?b) ^ hasCorrespondingCP(?a, ?acp) ^ hasCorrespondingCP(?b, ?bcp) ^ CP(?acp) ^ CP(?bacp) ^ hasManagingCP(?bcmp, ?bcp) ^ CMP(?bcmp) ^ hasRequestRelation(?acp, ?r) ^ hasStartRelation(?bcmp, ?s) ^ hasDeliverRelation(?bcp, ?d) ^ PA_1st_cut_Diagram(?d1)** **→** **Deliver(?d) ^ hasCPSource(?d, ?bcp) ^ hasCPDestination(?d, ?acp) ^ Request(?r) ^ hasCPSource(?r, ?acp) ^ hasCMPDestination(?r, ?bcmp) ^ Start(?s) ^ hasCMPSource(?s, ?bcmp) ^ hasCPDestination(?s, ?bcp) ^ belongsTo1stCutDiagram(?acp,?d1) ^ belongsTo1stCutDiagram(?bcp,?d1) ^ belongsTo1stCutDiagram(?bcmp,?d1)**
Riva Step	**Step 6: Use the provided heuristics, to generate a GQ second-cut process architecture.**
Related Classes	**PA_1st_Diagram, GQ_PA_2nd_Diagram**
Class Properties	**isActive: default value set to True, and to be modified by the user if the corresponding CMP needs to be deleted. It is also used to inactivate the relations Start and Request of the deleted CMP and the relation Deliver when the heuristics apply.** **belongsTo2ndCutDiagram**
Class Instances	**GQ_PA_2nd_Cut_Diagram instance represents the specific 2nd cut BPA diagram for an organisation.**
Jess usage (in case an ontology instance for an organisation is to be developed)	**Default values for isActive can be set, and then changed by the user.** **A Jess rule can also be used to automatically inactivate the Start and Request relations for all inactive CMPs.**

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

OWL restrictions	CP: ∀ *BelongsTo2ndCutDiagram only PA_2nd_Diagram* CMP: ∀ *BelongsTo2ndCutDiagram some PA_2nd_Diagram*
SWRL Rules	**Rule_inactive_CMP_relevant_Relations**: *CMP(?bcmp) ^ isActive(?bcmp, False) ^ hasStartRelation(?bcmp, start) ^hasRequestRelation(?acp, ?request)^ hasCMPSource(?request, ?bcmp)* *→Request(?request) ^ isActive(?request, False) ^Start(?start) ^ isActive(?start, False)*
Riva Step (Refined)	**Step 7:** For each process in the generated GQ_PA_2nd_Cut_Diagram, identify the Goals of the Process (GoP) and the Quality of the Process (QoP).
Related Classes	**GOP, QOP, CP, CMP, GQ_PA_2nd_Cut_Diagram**
Class Properties	**hasGoP** and **hasQoP:** to set the goals of the process and the quality requirements of the process respectively. **belongsToProcess:** Any identified goals in the 2nd cut diagram must be related to a process, which is CP or CMP, using this property. **constrainsProcess:** the processes that set up the 2nd cut architecture diagram and that are characterised with desired quality requirements must allow the QOP be related to using this property. **isElaborative**: is set to true if the QOP instances are elaborative and has a corresponding reference for their quality models. **hasQualityReference:** to set the needed quality model reference for the CPs and CMPs. **belongsTo2ndCutDiagram**
Class Instances	GOP and QOP instances for each process that constitute the GQ_PA_2nd_cut_diagram.

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

<table>
<tr><td>OWL Restrictions</td><td>CP: ∀ hasGoP only GOP.
∀ hasQoP only QOP
∀ hasQualityReference only Quality_Model_Reference
CMP: ∀ hasGoP only GOP.
∀ hasQoP only QOP
∀ hasQualityReference only Quality_Model_Reference
GOP: ∀ belongsTo2ndCutDiagram only GQ_PA_2nd_cut_Diagram
∀ belongsToProcess only CP
∀ belongsToProcess only CMP
QOP: ∀ belongsTo2ndCutDiagram only GQ_PA_2nd_cut_Diagram
∀ belongsToProcess only CP
∀ belongsToProcess only CMP</td></tr>
<tr><td>SWRL Rules</td><td>Rule_GoP_CP_or_CMP_in_PA2_Diagram:
hasGoP(?p, ?gop) ⋀ belongsTo2ndCutDiagram(?p, GQ_PA_2nd_cut_Diagram) → GOP(?gop) ⋀ belongsToProcess(?gop, ?p) ⋀ belongsTo2ndCutDiagram(?gop, GQ_PA_2nd_cut_Diagram)

Rule_Elaborative_QoP_CP_or_CMP_in_PA2_Diagram
hasQoP(?p, ?qp) ⋀ isElaborative(?qp, true) ⋀ hasQualityReference(?qp, ?qref) ⋀ belongsTo2ndCutDiagram(?p, GQ_PA_2nd_cut_Diagram) → QOP(?qp) ⋀ constrainsProcess(?qp, ?p) ⋀ belongsTo2ndCutDiagram(?qp, GQ_PA_2nd_cut_Diagram)

Rule_NonElaborative_QoP_CP_or_CMP_in_PA2_Diagram
hasQoP(?p, ?qp) ⋀ belongsTo2ndCutDiagram(?p, GQ_PA_2nd_cut_Diagram) ⋀ isElaborative(?qp, false) → QOP(?qp) ⋀ constrainsProcess(?qp, ?p) ⋀ belongsTo2ndCutDiagram(?qp, GQ_PA_2nd_cut_Diagram)</td></tr>
</table>

Table I.1 (Cont'd): The OWL-DL and the SWRL Rules of the GQ-Riva Method Steps [Source: (Yousef,2010), Adapted with the author's permission].

Outside World	The outside world can be used in all Riva diagrams, and can be related to any element.
Related Classes	**Outside_world, Outside_Relation, UOW_Diagram, PA_1st_Diagram, PA_2nd_Diagram**
Class Properties	**hasOutsideworldRelation** to assert the outside relation for each outside world element. **belongsToUOWDiagram** to assert that the diagrams belongs to**: hasOutsideworldSourse** and **hasUOWDestination**.
Class Instances	Instances of the class Outside_world and Outside_relation are entered by the user. Theses are only graphical elements which does not affect Riva derivation.
Jess usage (in case an ontology instance for an organisation is to be developed)	hasOutsideworldRelation and hasOutsideworldSource values can be entered using the JessTab.
OWL restrictions	*Outside_world:* ∀ *hasOutsideWorldRelation only Outside_relation* *Outside_world:* ∀ *belongsToUOWDiagram only UOW* *Outside_world:* ∀ *belongsTo1stCutDiagram only PA_1st_Diagram* *Outside_world:* ∀ *belongsTo2ndCutDiagram only PA_2nd_Diagram* *Outside_relation:* ∀ *hasOutsideWorldSource only outside_world* *Outside_relaion:* ∀ *hasUOWDestination only UOW* *Outside_relaion:* ∀ *hasCPDestination only CP* *Outside_relaion:* ∀ *hasCMPDestination only CMP*
SWRL Rules	-

Appendix J: Alignment Rules for the GQ-srBPA Ontology

This appendix presents the alignment classes, properties and SWRL that are not shown in Appendices H and I. This separation is recommend in order to highlight the alignment related OWL-DL specification apart from the GQ-srBPA ontology (i.e., attached in Appendices H and I). This allows users to employ the GQ-srBPA ontology regardless the need for an alignment with ta BSV. In short, this separation is recommended in order to address users needs. Table J.1 presents the alignment related OWL-DL classes and properties. Table J.2 shows alignment related SWRL rules.

Table J.1: The Semantic-based Alignment Rules for the First Alignment Approach.

Class	Description	Properties
G_EBE	The BSV driven EBEs	1) isNewEBE: Boolean, 2) isConsideredUoW: Boolean
New_EBE	New EBEs are classified from the G_EBE because they are new.	
Detected_EBE	Detected EBEs are classified from the G_EBE because they are still needed in the Riva BPA.	
EBQ		1)isNewEBQ: Boolean,
New_EBQ		
Detected_EBQ		
G_UOW	The UoWs filtered out of G_EBEs.	1) isNewUOW: Boolean, 2) hasCorrespondingCP of type CP.
New_UOW	New UoWs are classified from the G_UOW because they are new ones and needed.	
Detected_UOW	Detected UoWs are classified from the G_UOW because they are still needed UoWs in the Riva BPA.	
Quality_Model_Reference	The reference for elaborative Q_UoW.	1)isNewQualityRerfernce: Boolean
New_Quality_Reference	New quality reference added to the Q-UoW	
Detected_Quality_Reference	Current quality reference is still needed with the Q-UoW	
CP	A case process in the 2nd cut architecture.	1) isNewProcess: Boolean, 2) hasManagedByCMP of type CMP,
CMP	A case management process in the 2nd cut architecture.	1) isNewProcess: Boolean,

Table J.2: The Semantic-based Alignment Rules for the Second Alignment Approach

Alignment Related SWRL Rule	Description
Rule-Classifiying_GEBEs_into_New_EBEs: G_EBE(?x) ∧ isNewEBE(?x, true) → New_EBE(?x)	This rule classifies the G_EBE instances into new EBEs.
Rule-Classifying_GEBEs_into_Detected_EBEs G_EBE(?x) ∧ isNewEBE(?x, false) → Detected_EBE(?x)	This rule classifies the G_EBE instances into detected EBEs.
Rule-Classifying_EBQ_into_NewEBQs EBQ(?ebq) ∧ isNewEBQ(?ebq, true) → New_EBQ(?ebq) **Rule-Classifying_EBQs_into_Detected_EBQs** EBQ(?ebq) ∧ isNewEBQ(?ebq, false) → Detected_EBQ(?ebq)	The EBQ are classified into new ones in the first SWRL rule, where the second rule classifies the EBQs into detected ones that are currently exist and needed.
Rule-Classifying_GUoW_into_New_UoWs G_EBE(?ebe) ∧ isNewEBE(?ebe, true) ∧ isConsideredUoW(?ebe, true) → New_UOW(?ebe) ∧ isNewUOW(?ebe, true) **Rule-Classifying_GUoW_into_Detected_UoWs:** G_EBE(?ebe) ∧ isConsideredUoW(?ebe, false) ∧ isNewUOW(?ebe, true) → Detected_UOW(?ebe) ∧ isNewUOW(?ebe, false)	This rule classifies the G_UoWs into new and detected UoWs based on their EBEs.
Rule-Classifying_New_Quality_Reference Quality_Model_Reference(?qr) ∧ isNewQualityReference(?qr, true) → New_Quality_Reference(?qr) **Rule-Classifying_Detected_Quality_Reference** Quality_Model_Reference(?qr) ∧ isNewQualityReference(?qr, false) → Detected_Quality_Reference(?qr)	This rule classifies the quality reference into new and detected UoWs based on their EBEs.
Rule-Classifying_New_CPs: G_UOW(?u) ∧ isNewUOW(?u, true) ∧ hasCorrespondingCP(?u, ?cp) → isNewProcess(?cp, true) **Rule-Classifying_New_CMPs:** G_UOW(?u) ∧ isNewUOW(?u, true) ∧ hasCorrespondingCP(?u, ?cp) ∧ hasManagedByCMP(?cp, ?cmp) → isNewProcess(?cmp, true) ∧ isNewProcess(?cp, true)	This rules classifies the processes in the 2nd cut architecture into new ones.
Rule-Classifying_Detected_CPs: G_UOW(?u) ∧ isNewUOW(?u, false) ∧ hasCorrespondingCP(?u, ?cp) → isNewProcess(?cp, false) **Rule-Classifying_Detected_CMPs:** G_UOW(?u) ∧ isNewUOW(?u, false) ∧ hasCorrespondingCP(?u, ?cp) ∧ hasManagedByCMP(?cp, ?cmp) → isNewProcess(?cmp, false) ∧ isNewProcess(?cp, false)	This rules classifies the processes in the 2nd cut architecture into detected ones.

Appendix K: Interview Questions

Interview Questions

Q1-Please tick the most relevant main objective(s) of CEMS Faculty Administration process.

1. Growth and continuity of the organization
2. Meeting financial objectives
3. Meeting personal objectives
4. Meeting responsibility to employees
5. Meeting responsibility to society
6. Meeting responsibility to country
7. Meeting responsibility to shareholders
8. Managing market position
9. Improving business processes
10. Managing quality and reputation of products

Q2- Name the main actors and associated main goals (responsibilities) in CEMS faculty administration process. Also, list any sub actors if exist.

Actor	Goals	Sub actors (if exist)

Q3- According to the above **goals**, does each actor depend on another to achieve his/her intentions? If yes, define the **Dependers** who depend on **Dependees** to achieve their goals. (You can Draw a table or diagram as shown in the attached slides)

Q4- What are the associated **quality requirements** (constraints or quality attributes) for each goal if exist. List main quality requirements in this process. (In relation to Activity, Human, Non-Human (systems), Data and Information and any others e.g: exception).

Q5- Discuss how it is possible to achieve each goal and quality. (To easily answer this question, please use the Breaking Rule as shown in slide number 10 and 11).

Appendix L: Aligning the BSV and the Riva BPA: A Case Study Using the CEMS Faculty Administration

This appendix presents results of an investigation carried out in order to discover the alignment relation between the BSV and the corresponding as-is Riva-BPA using the CEMS Faculty of Administration. The investigation involved choosing the appropriate goal-oriented approach. The *i** framework goal-oriented approach was chosen and employed in this investigation due to its richness of goal-oriented concepts (e.g., goal, task, actor, resource) for the relevant Riva BPA concepts (e.g., process and EBE). The CEMS Faculty Administration process example is used in order to produce the pilot results for this research before conducting a comprehensive evaluation of the alignment using the CCR case study, which is a larger example.

Time and effort are allocated for this investigation, as no attempts were found regarding bridging the gap the *i** framework GO models and Riva-based BPAs. The alignment is carried out after conducting three face-to-face interviews in order to elicit the hard goals and soft goals of the CEMS Faculty Administration taking into account the existence of its as-is Riva BPA. The number of conducted interviews is few due to the limited time allowed for the interviewee to participate. The current Riva BPA for the CEMS Faculty Administration already exists in (Ould and Green, 2004) and as attached in Appendix A. Due to the limited interview sessions, a number of partial SD models are obtained and mapped them to the related corresponding part of the CEMS Faculty Administration Riva BPA.

The reasons beyond selecting the CEMS Faculty Administration process instead of the CCR for this investigation are described next. **First**, the CCR is allocated for the comprehensive evaluation following the work in (Yousef, 2010). **Second,** the CCR process runs in the health care sector for the overall evaluation. Therefore, employing another example that runs in a different domain is desired in order to inform the alignment approach with domain independent examples. This enhances the author's earlier assertion regarding the domain independency characteristic of the proposed framework, which was discussed in Chapter 3. Although the CEMS Faculty Administration example is employed in investigating the alignment, still it paths the way for evaluating the overall framework in Chapter 7. **Third,** the BPMs for this pilot study are absent and hence, this influences a less dependency on the resources required for the carrying out the alignment and paying more attention to the BPA instead of the BPMs.

This appendix is structured as follows. Section L.1 presents the BSV models for the CEMS Faculty Administration resultant from the three interviews. Section L.2 presents current Riva BPA model for the obtained BSV. It is necessary to inform the reader that the presented Riva BPA in this section in not aligned with the BSV yet. However, it is only extracted from Appendix A. Finally, Section L.3 presents the GQ-Riva BPA resultant from the carried out alignment for this pilot case study.

L.1 Designing Partial BSV for the CEMS Faculty of Administration

Although the BP models of the CEMS Faculty Administration are still absent; yet they exist in the participants mind. The associated documents of the CEMS Faculty Administration PA (i.e., EBE list, UoW diagram and 1st cut PA) exist in (Ould and Green, 2004). However, its goals and associated quality requirements are not documented. Therefore, the absent BPs, goals and quality requirements alerted to conduct interviews in order to generate the goal models. Due to limited time allocated for the interviews, a partial set of goal models for the corresponding Riva BPA are generated. The limited time of interviews did not allow the interviewer to proceed with the SR models in addition to the soft goals elicitation that did not address their NFR framework models representation. The interview questions are attached in Appendix K.

Figure L.1 illustrates a partial network of the elicited organisational goals in CEMS Faculty Administration example. The goals' elicitation started by determining the main goal of CEMS Faculty Administration, which is to improve the administration business process. The main goal appears as the highest goal in abstraction in Figure I.1. The goal was selected within the canonical list as discussed in Chapter 2. However, it is apparent that the main goal does not provide enough information regarding the way towards its fulfilment. In this case of ambiguity, an elaboration or refinement is required in order to derive more specific goals. Figure I.1 illustrates the decomposition of the main goal into six sub goals in order to reduce the ambiguity. The decomposition is in turn performed on these six goals and so on. Figure I.1 illustrates the resulted goal network, where the leafs contain the most specific or refined goals in the network.

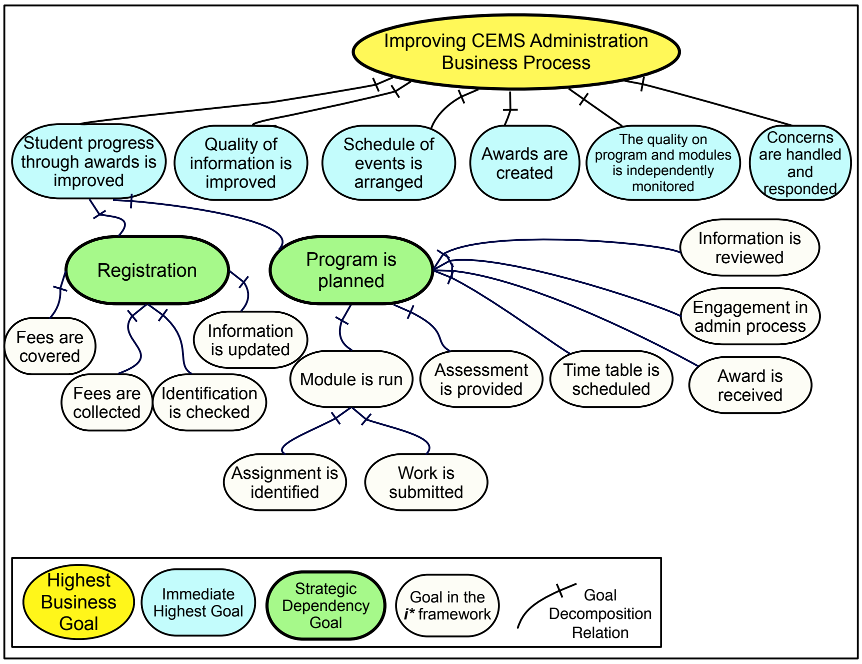


Figure L.1: The Elicited Goal Network for the CEMS Faculty of Administration

For each BP in the interviewee's mind, there was a corresponding SD model that designs the network of goal and soft goal dependencies in the BP. The participant was easily able to generate the SD models for the registration goal and the program-planned goal. The registration SD model is shown in Figure L.2. The SD model of the program-planned BP is shown in Figures L.3 and L.4. Each SD model in the three figures designs a partial SD model for its corresponding BP due to the limited session conducted for this investigation. However, these sessions were useful to path the way for the upcoming comprehensive evaluation.

In Figure L.2, a student is defined as an actor who depends on the CEMS admin team for the goal registered. The CEMS admin in turn depends on the student for the identification-

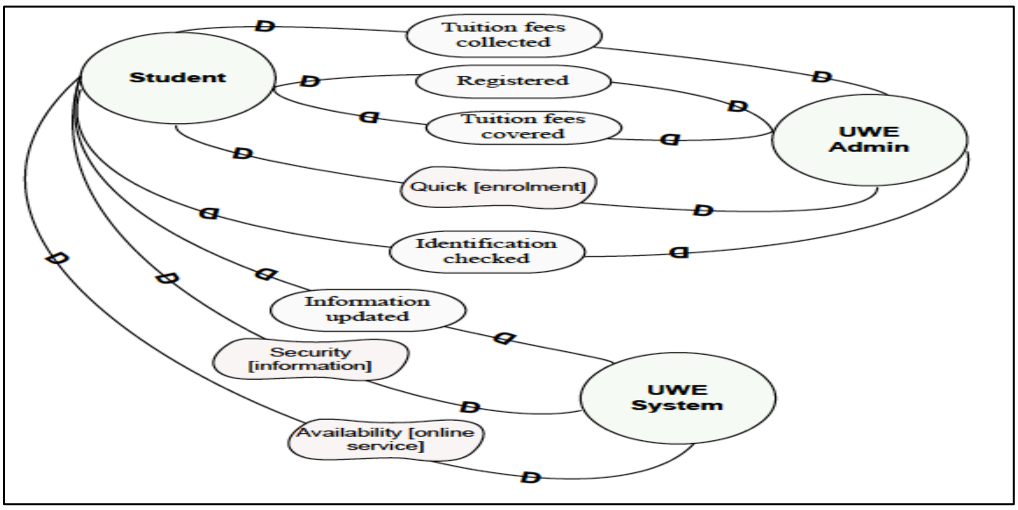


Figure L.2: Partial SD Model of the Registration BP in CEMS Faculty Administration

checked goal in order to fulfil the registration. These two dependencies are goal dependency relations. For a soft goal dependency example, a student depends on the UWE system for the availability of related information for registration. Moreover, a student depends on the UWE system to secure her/his information and by not making it visible to public.

The program planned BP SD models are illustrated into two figures in order to increase the clarity of the dependency relations. However, they both belong into one SD model that is program planned. In Figure L.3, a student depends on the CEMS admin for the program planned; module run and awards received goals. A timely module run initiates a soft goal dependency relation from a student to the CEMS admin in the same figure. In Figure L.4, a student depends on the CEMS admin for assessment provided goal where in turn the CEMS admin depends on the student for his/her engagement in the administration process. These two dependencies are designed as goal dependency relations. In Figure L.4, the availability of the CEMS admin produces a soft goal dependency from a student to the CEMS admin.

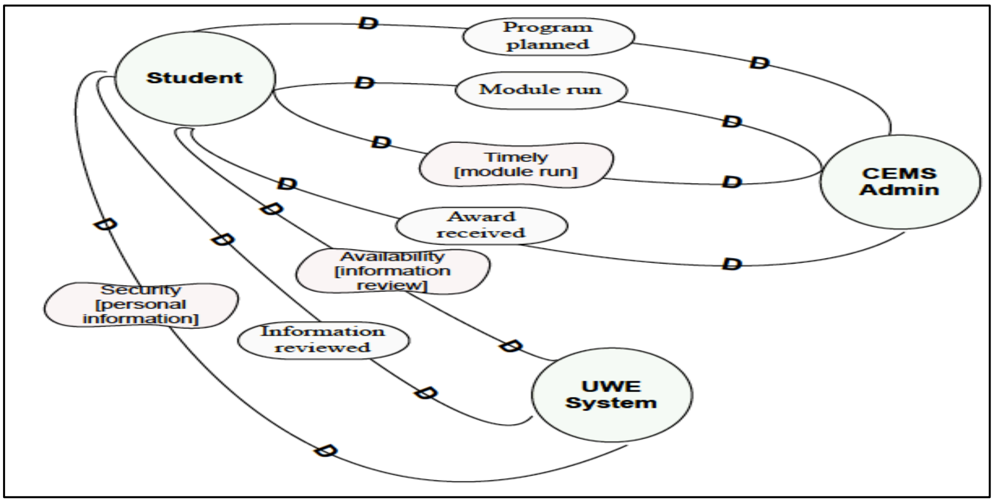


Figure L.3: The First Part of Partial SD Model of the Program Plan BP in CEMS Faculty Administration

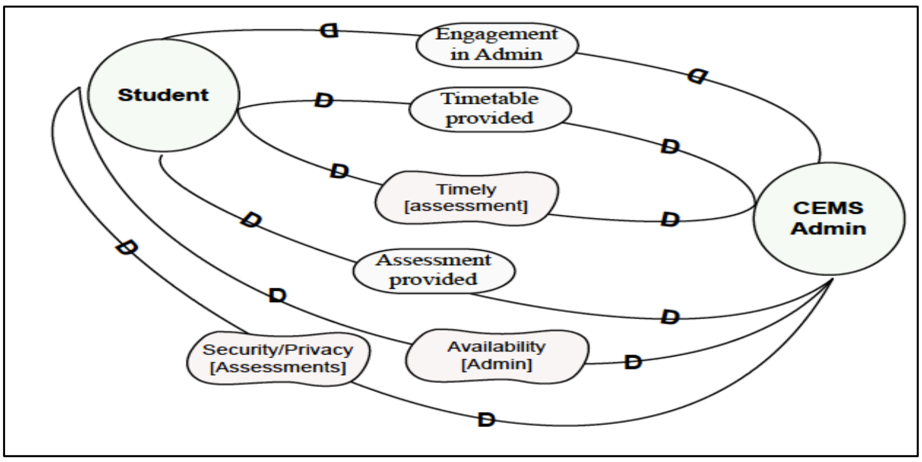


Figure L.4: The Second Part of Partial SD Model of the Program Plan BP in CEMS Faculty Administration.

L.2: The Current Riva BPA for the CEMS Faculty of Administration

This section resents the current Riva-based BPA for the CEMS Faculty of Administration extracted from Appendix A in order to align it with the BSV presented in Section L.1.

The current list of the identified EBEs and UoWs (i.e., as-is EBEs and as-is UoWs) for the CEMS Faculty Administration is shown in Table L.1, where the bold are UoWs. This table shows **9 EBEs** where **7 of them are UoWs**. This list was generated from the workshop conducted by Ould and Green in (Ould and Green, 2004). The overall as-is Riva models are not necessary in this stage, as the extraction is only employed for the investigation and initial evaluation purposes. The related extractions of the as-is Riva models for CEMS Faculty Administration are shown in Figures L.5 and L.6.

Table L.3: The as-is EBEs Before the Alignment with the BSV for the CEMS Faculty of Administration

As-is EBEs and as-is UoWs in the CEMS GO Models
Student
Program plan
Module run
Award definition/creation
Student tuition fees
Submission
Assignment identification
Award
UWE admin

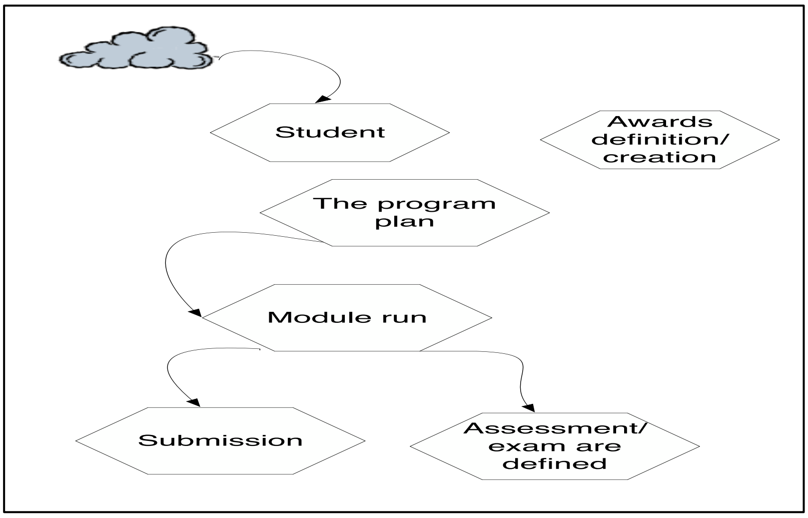


Figure L5: The extracted relevant as-is UoW diagram from the CEMS Faculty Administration Example. (Source: (Ould and Green, 2004), Adapted with the author's permission).

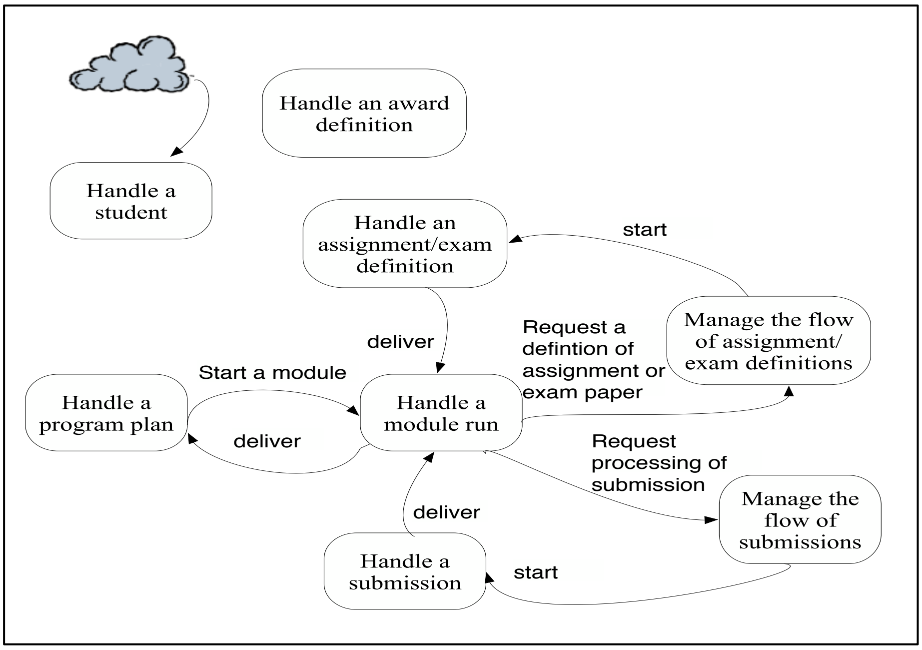


Figure L.6: Partial as-is Riva BPA of CEMS Faculty Administration [Source: (Ould and Green, 2004), Adapted with the author's permission].

L.3: Aligning the Current Riva BPA with the BSV for the CEMS Faculty of Administration

According to the available inputs for the alignment, Algorithm 5.1 "Aligning the as-is Riva-BPA, designed using the original approach, to the BSV of a Particular Business Organisation" appears the appropriate algorithm in order to conduct the desired alignment.

Concepts mapping appears as an appropriate method for the desired alignment in order to initiate bridging that fills the gap between the GO models and Riva-based PAs. In particular, EBEs that are in the GO models (i.e., represent part of a BSV for the organisation) are detected, as they will be reused for the to-be BPA. However, some of the EBEs that live in the GO models are new for the as-is EBE list thus, they are derived and not detected. The detected and derived EBEs are shown Table L.2. As Algorithm 5.1 is employed for this alignment, then EBQs are derived as shown in Table L.3.

Table L.2: The Detected and the Derived EBEs and UoWs in the CEMS Faculty Administration pilot study.

Detected: As-is EBEs and as-is UoWs in the CEMS GO Models	**Derived: New EBEs and UoWs from the CEMS GO models**
Student	**Registration**
Program plan	Tuition fees collection
Module run	Identification check
Award definition/creation	**Timetable schedule**
Student tuition fees	Program monitoring
Submission	Module monitoring
Assignment identification	Events schedule
Award	Tuition fees cover
UWE admin	Student information update
	Student engagement
	Assessment
	Student enrolment

Table L.3: EBQs for the CEMS Faculty Administration Process

EBQs for the CEMS Faculty of Administration Process
Quick [student enrolment]
Security [student information]
Timely [module run]
Privacy [student information]
Security [submission]
Privacy [submission details]
Accuracy [timetable schedule]
Timely [registration]
Accuracy [Assignment and exams]

The EBEs and the UoWs in the Alignment

Although the investigation rustled few goal models, still they are useful and rich to proceed with discovering the desired alignment relation and its implication.

First, the resulted BSV for CEMS Faculty Administration are shown from Figures L.1 to L.4. Second, the as-is Riva elements and models are shown from Figures L.5 and L.6. Since the alignment embodies the concept of reuse as presented in Section 5.2, the as-is EBEs list is reused in order to extract the EBEs that are in the resulted BSV. And this emphasises the concept of reuse. Therefore, the as-is EBEs in the goal models are called **detected EBEs.** This is because they have been reused. However, the systematic elaboration of goals stimulated the identification of new EBEs, namely **the derived EBEs**. The detected and the derived EBEs in GO models in the form of actor or goal and this is the way of how the concepts are mapped between the Riva-oriented concepts and the GO concepts. The examples of the detected and derived EBEs are shown in Table L.2, where the UoWs are the EBEs in bold. The number of EBEs after the alignment is **21** where the number of UoWs is **9**. For example, the registration goal plays an important part in the BPA and it is a new added knowledge in the GQ-UoW diagram as shown in Figure L.7 and in the ultimate model that is the to-be GQ-Riva-BPA as shown in Figure L.8. A similar example is the timetable schedule goal. Table L.4 presents information about processes shown in Figure L.8. Table L.5 summaries the number of EBEs and UoWs before and after the alignment. Also, it shows the number of processes that design a Riva-BPA before and after the alignment for the CEMS Faculty of Administration pilot study.

The Quality Integration

The generated SD models represent the quality perspective through the soft goal concept. The current Riva method lacks the integration of quality requirements (Ould, 2006). Therefore, the current CEMS Faculty Administration Riva BPA does not represent the quality requirements perspective, yet it only represents the functional perspective in the form of the EBEs, UoWs, CPs and CMPs. These functional-oriented concepts require an association of the non-functional-oriented concepts that represent the desired attributes on the functional ones. The concept mapping is referred again to perform the quality requirements integration through using the mapping of the soft goal concept in the GO models to a new quality concept in the Riva method, namely the Essential Business Quality (EBQ). The EBQ refers to the main NFR for a particular EBE or UoW. The non-functional concepts existence assists in bridging the gap between soft goal models and BPAs.

It is claimed that each of the EBQs in the UoW diagram has a corresponding NFR framework in Figure L.7. The goal-based and quality-linked Riva BPA of this pilot study is shown in Figure L.8. A star is associated with the quality-linked process where the colored process denotes for the new BP because of the goals alignment. Table L.4 shows the quality of process linked within each process in the GQ-Riva BPA in Figure L.8. Table L.5 shows the number of the quality-linked processes before and after the alignment.

Using Table L.5, it is apparent that number of EBEs, UoWs and process in increased after conducting an alignment with the BSV. New EBEs, UoWs and process means new knowledge integrated into the current Riva BPA. This signals that the current Riva BPA (i.e., before the alignment) missed addressing and representing required information. Hence, this may make the current Riva BPA as a not useful resource for development. Moreover, having goals of a process as shown in Table L.4 informs about addressing business goals in the Riva-

BPA. In each process in Table L.4, the first goal is the goal of this process, where the reset inform about contribution to other goals. Having BPs linked with quality requirements paves the way to increase its acceptability in the domain and to improve the BP in the future.

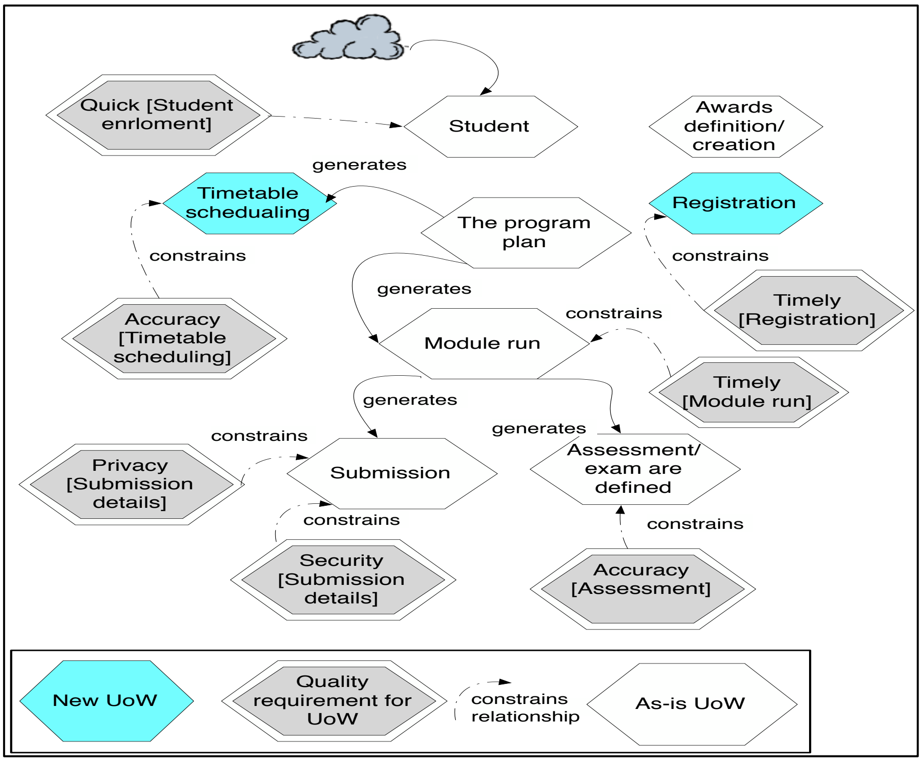


Figure L.7: Partial Snapshot of the Generated to-be GQ UoW Diagram From an Alignment With the Quality-Linked Goal Models

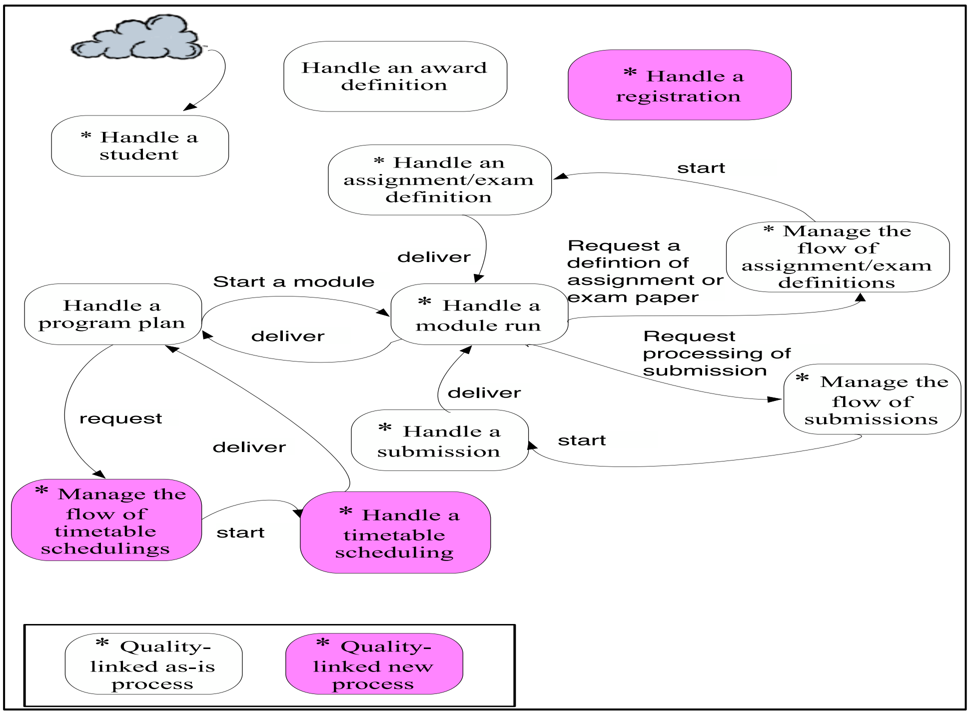


Figure L.8: Partial Snapshot of the Generated to-be Riva-based BPA of CEMS Faculty Administration.

Table L4: Goal-based and Quality-linked BPs in the CEMS Faculty of Administration Riva BPA

BP no	BP Name	New BP/ As-is BP	Goals of the Process	Quality of the Process
1	Handle a module run	As-is BP	1- **Module run**, 2- Program plan. 3-Student progress through awards.	1- Timely.
2	Handle a submission	As-is BP	1-**Submission is handled**, 2- Student progress through awards	1- Security. 2- Privacy.
3	Manage the flow of submissions	As-is BP	1-**Sumbissiona are managed**, 2- Program is planned, 3- Student progress through awards.	1- Security. 2- Privacy.
4	Handle a program plan	As-is BP	1- **Program is planned**, 2-Student progress through awards.	NFR is not determined.
5	Handle a timetable scheduling	New BP	1- **Timetable schedule is handled**, 2-Student progress through awards.	1- Accuracy.
6	Manage the flow of the timetable scheduling	New BP	1- **Timetable schedule is managed**, 2- Student progress through awards.	1- Accuracy
7	Handle a student	As-is BP	1**- Student is handled**, 2-Student progress through awards.	1-Quick [student enrolment]
8	Handle an award definition	As-is BP	1-**Award is received or handled**, 2- Program is planned, 3-Student progress through awards.	NFR is not determined.
9	Handle a registration	New BP	1- **Registration is handled**, 2-Student progress through awards.	1- Timely.
10	Handle an assignment/exam definition	As-is BP	1- **Assignment and exam definition is handled**, 2- Module is run, 3- Program is planned, Student progress through awards	1- Accuracy.
11	Manage the flow of assignment/exam definitions	As-is BP	1- **Assignment and exam definition is managed**, 2- Module is run, 3- Program is planned, Student progress through awards.	2- Accuracy.

Table L.5: Quantative Summary for Riva BPA Elements Before and After the Alignment for the CEMS Faculty of Administration.

	Before the Alignment	After the Alignment
Number of EBEs	9	21
Number of UoWs	6	8
Number of BPs in the Riva BPA	8	11
Number of the quality-linked BPs in the Riva BPA	Zero	6

Appendix M: The CCR Process Business Strategy View

This appendix presents the CCR process BSV resultant from the instantiation of the first layer of the GQ-BPAOntoSOA framework. Since only three processes (i.e., patient reception, cancer detection and cancer treatment processes) out of six are considered representative enough of the CCR process, as shown in Chapter 7 (Section 7.3.1), they are agreed as an organization that requires the presence of a BSV. The original representation of the three processes is attached in Appendix F. The BSV for these processes is represented using a BS model (in Figure M.2), a HSD model (in Figure M.3), three SD models (in Figure M.4-M.6) and three corresponding SR models (in Figures M.7-M.9) aided with linking with the associated quality models (i.e., NFR framework models of the three processes in Appendix F). Also, the goal network of the CCR is shown in Figure M.10. A snapshot from the Protégé window is captured for each component in the GQOnt ontology instantiation layer. Figure M.11, M.12 and M.13 shows the instantiation of the CCR_siGoal ontology, CCR_sQuality ontology and CCR_GQOnt, respectively.

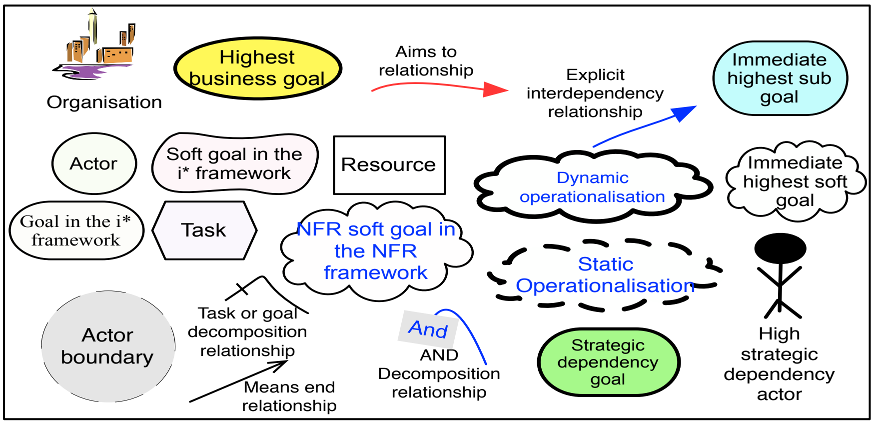


Figure M.15: A Notation Key for the BSV Models

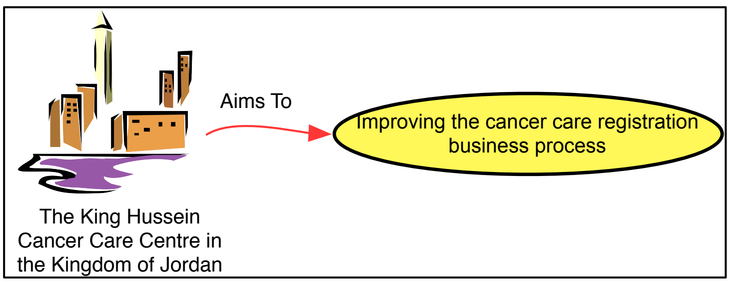


Figure M.2: The BS Model for the CCR Case Study

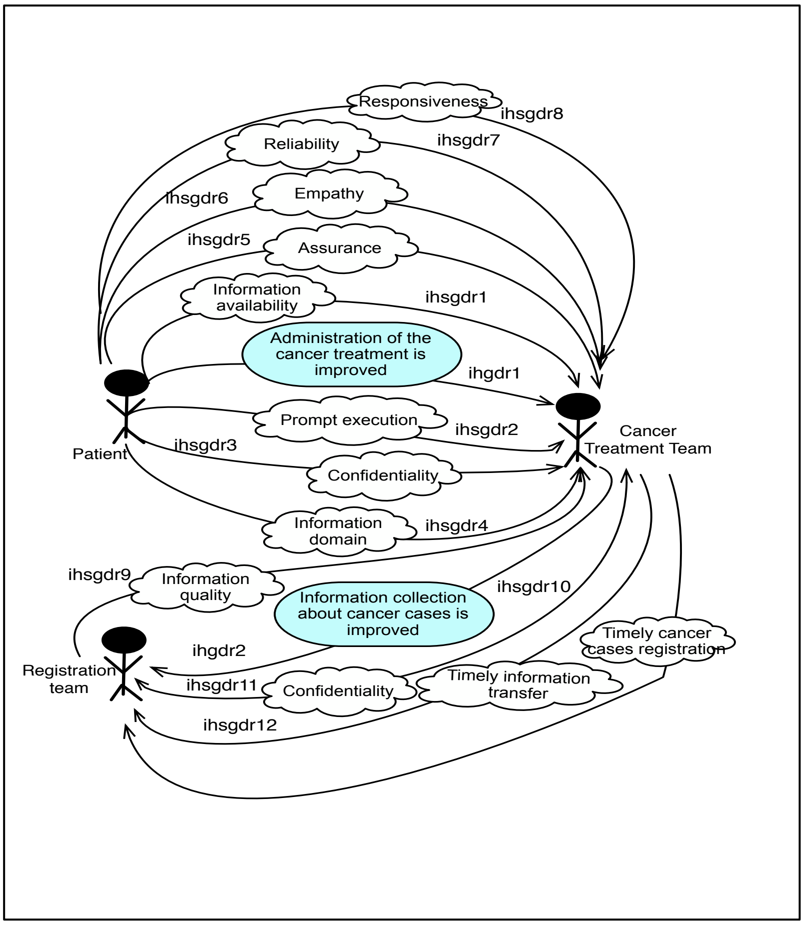


Figure M.3: The HSD Model for the CCR Case Study

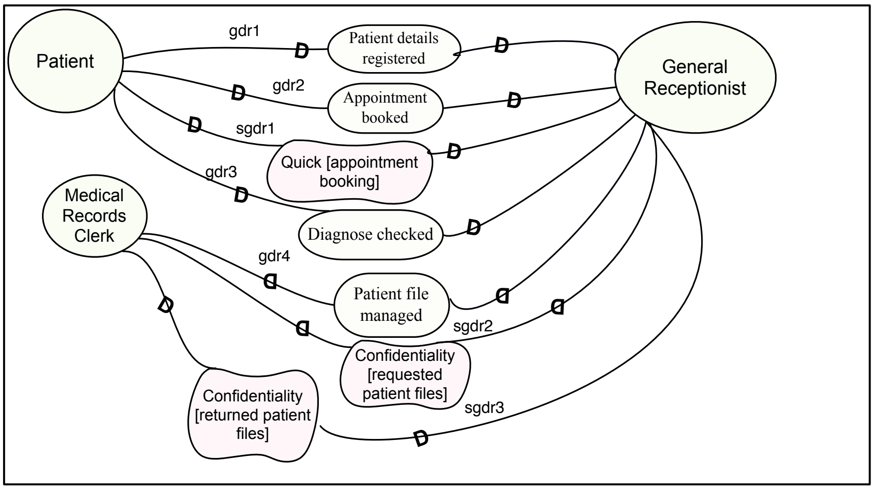


Figure M.4: The SD Model for the Patient Reception Process

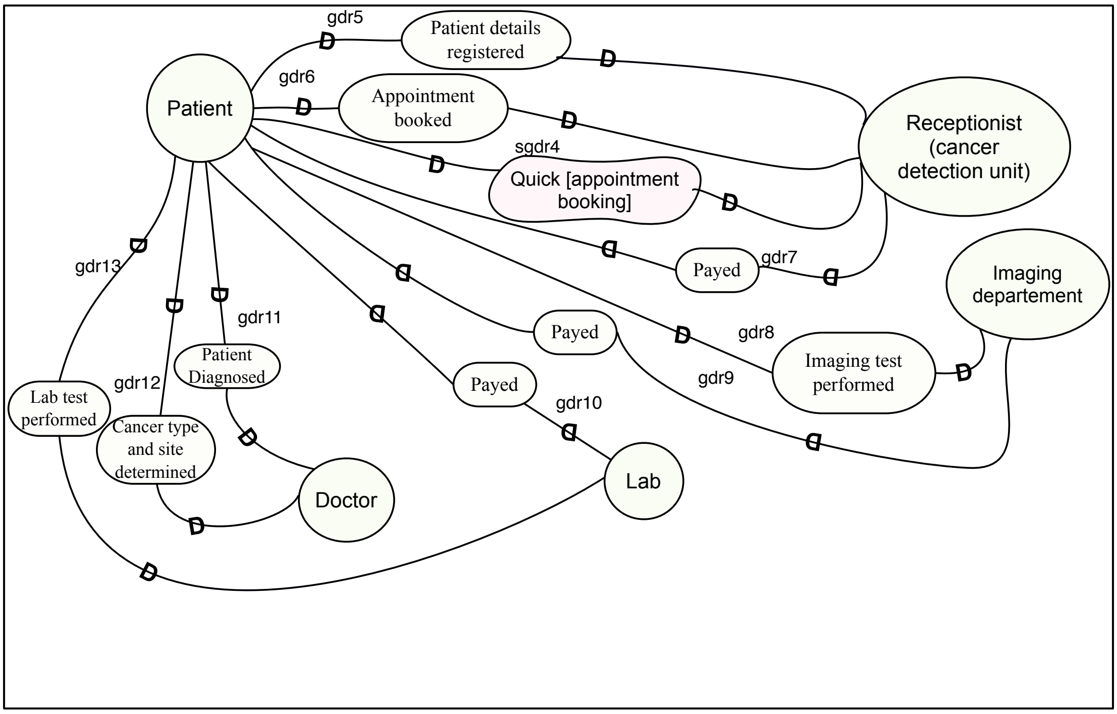


Figure M.5: The SD Model for the Patient Detection Process

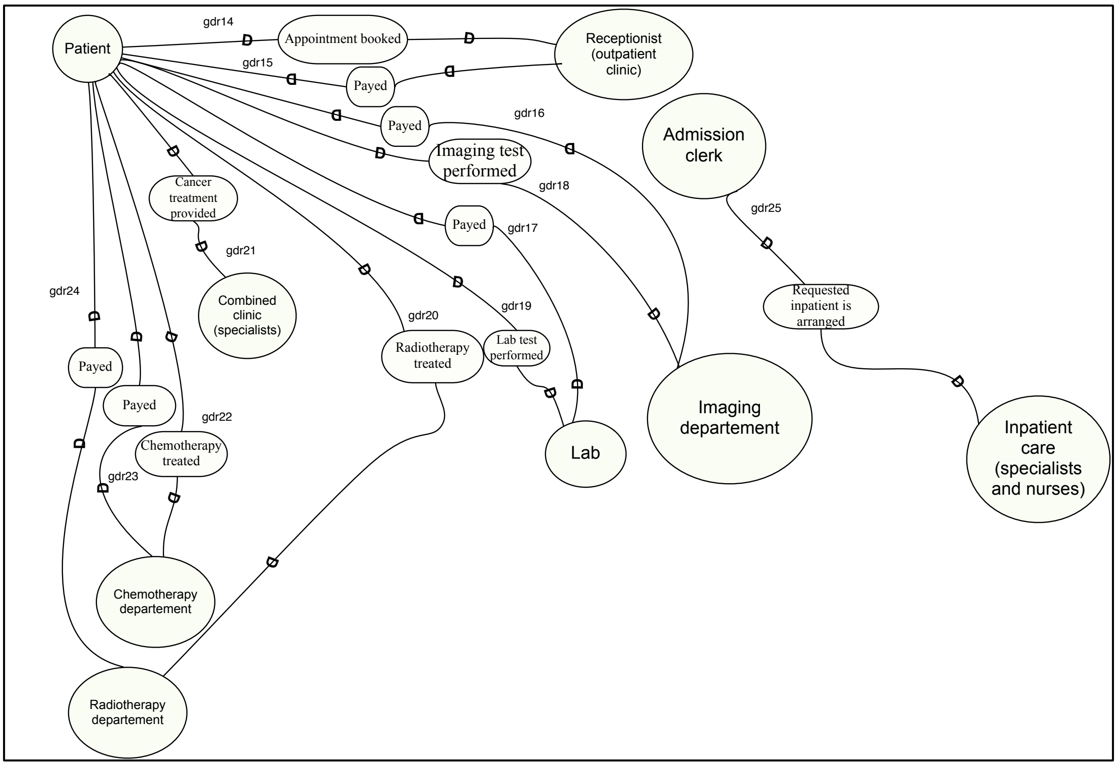


Figure M.6: The SD Model for the Patient Treatment Process

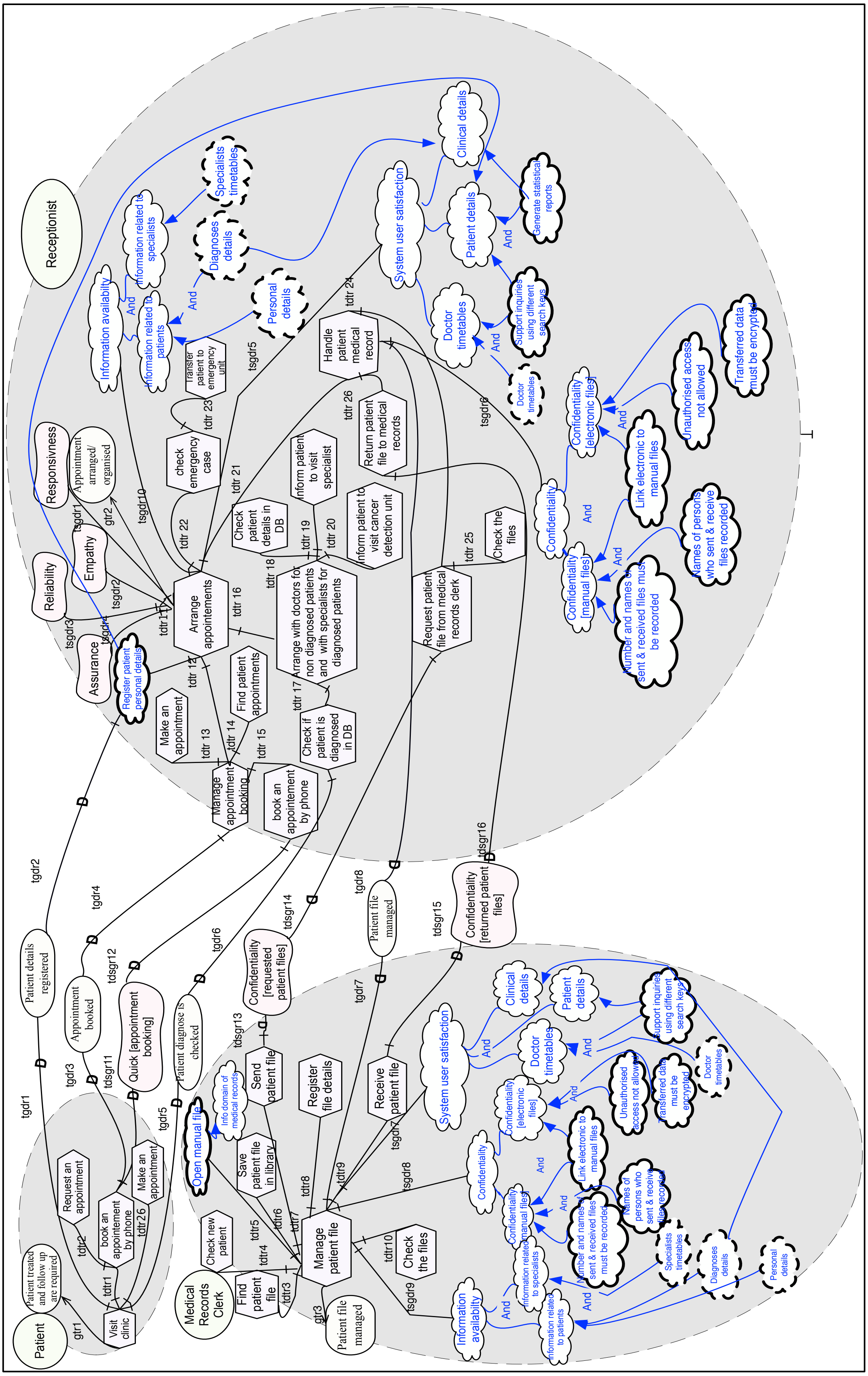


Figure M.7: The SR Model for the Patient Reception Process

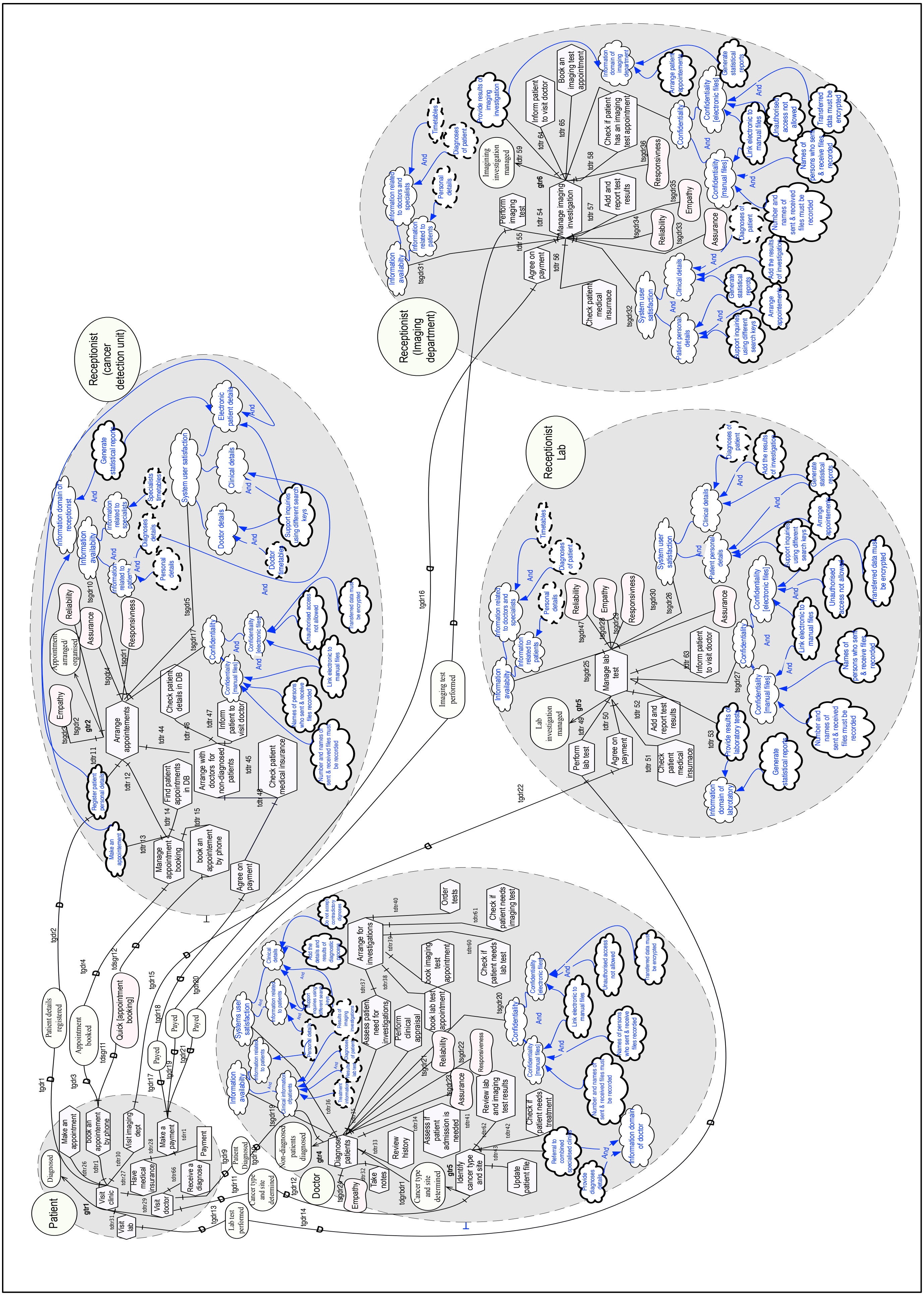


Figure M.8: The SR Model for the Cancer Detection Process

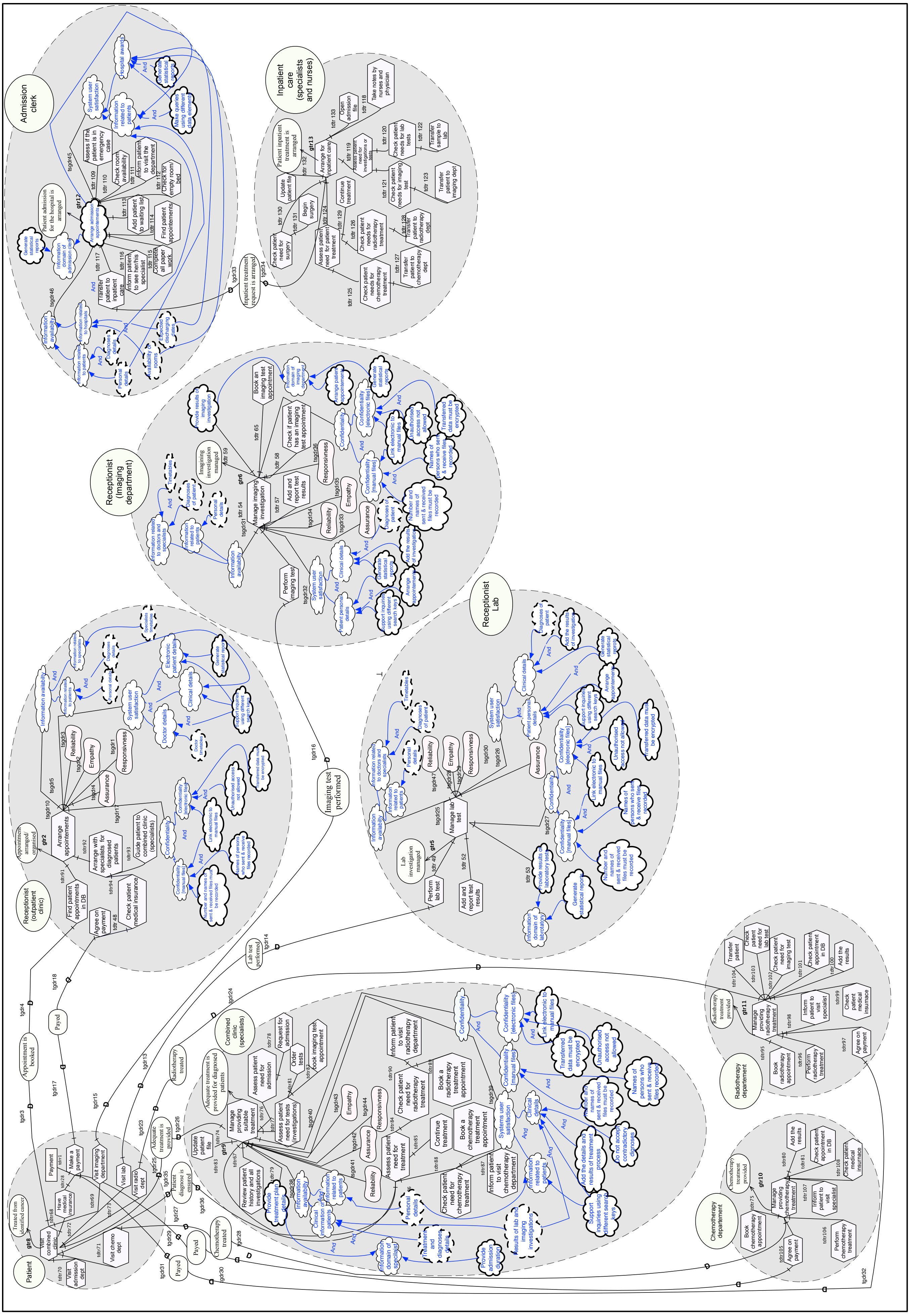


Figure M.9: The SR Model for the Cancer Treatment Process

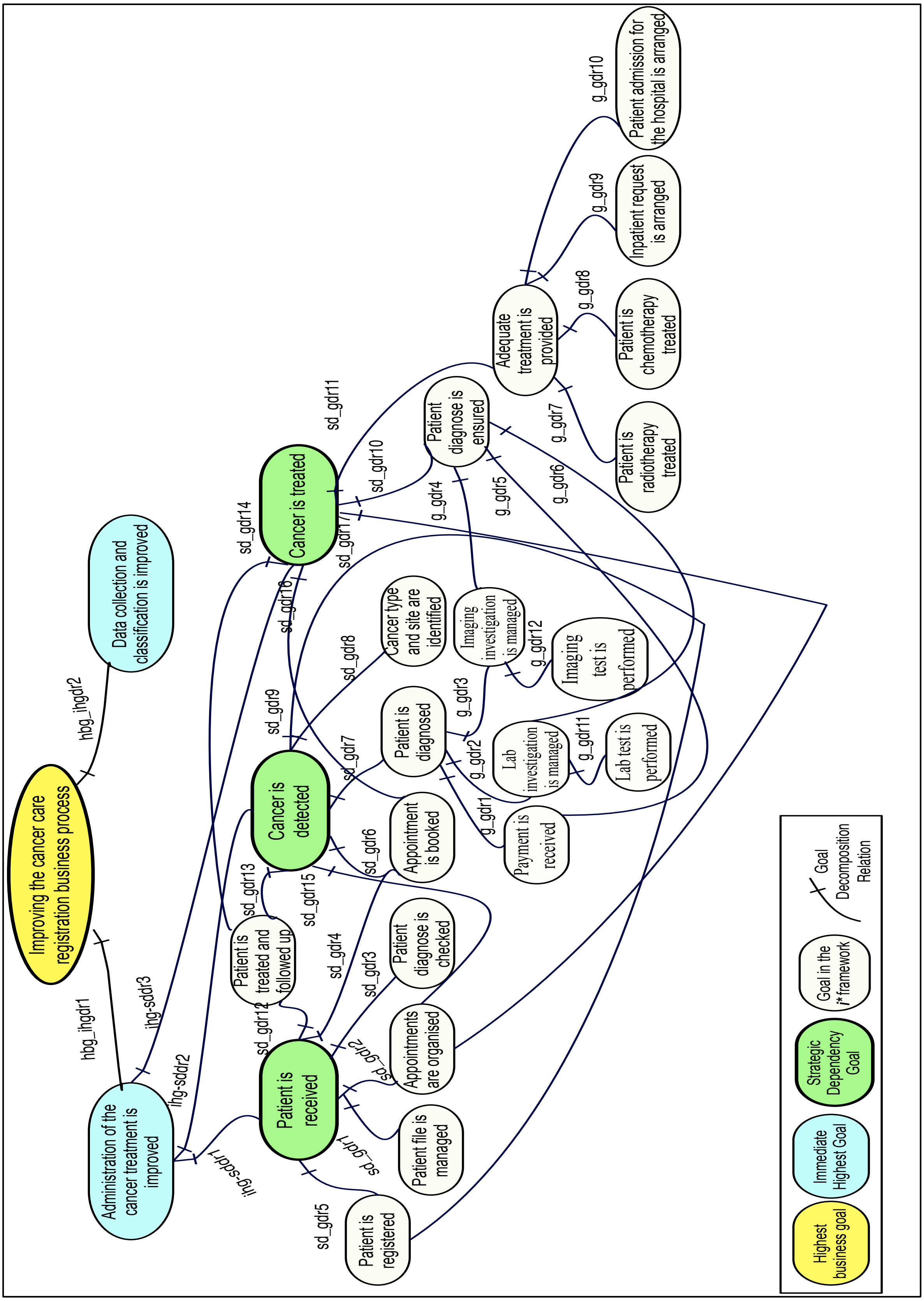


Figure M.160: Goal Network for the CCR Process

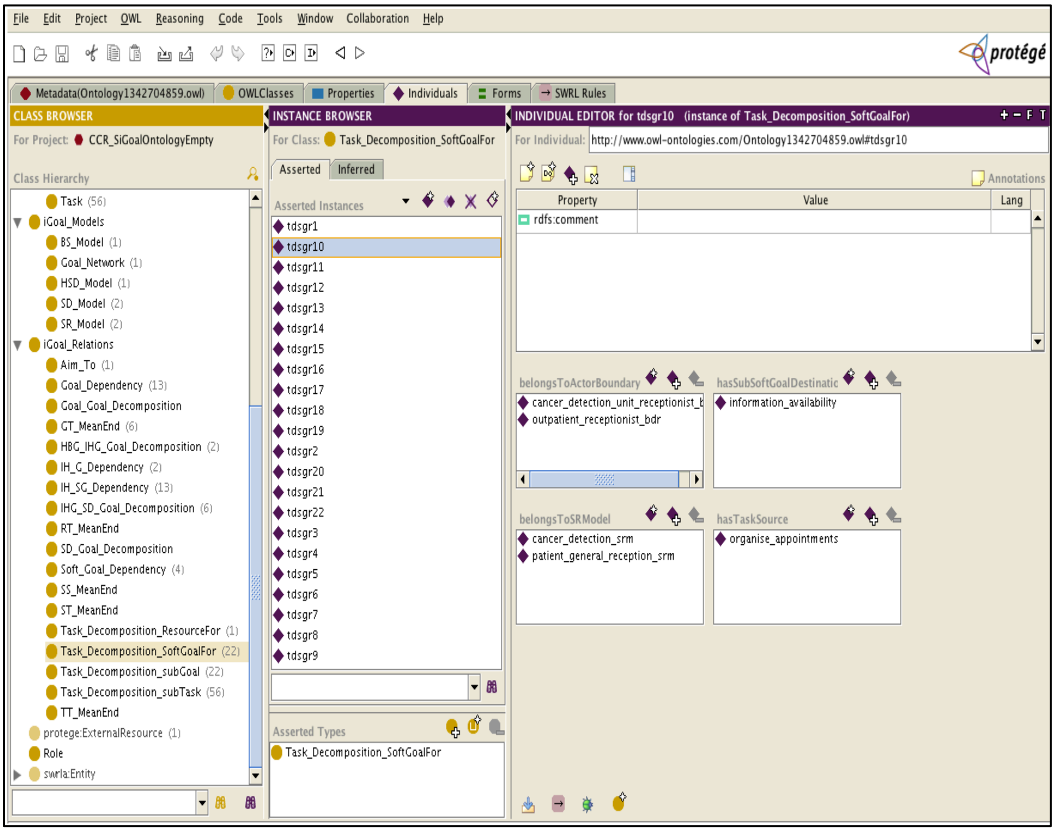


Figure M.11: A Snapshot Captured from Protégé Editor Window Showing An Instantiation of siGoal_CCR Ontology.

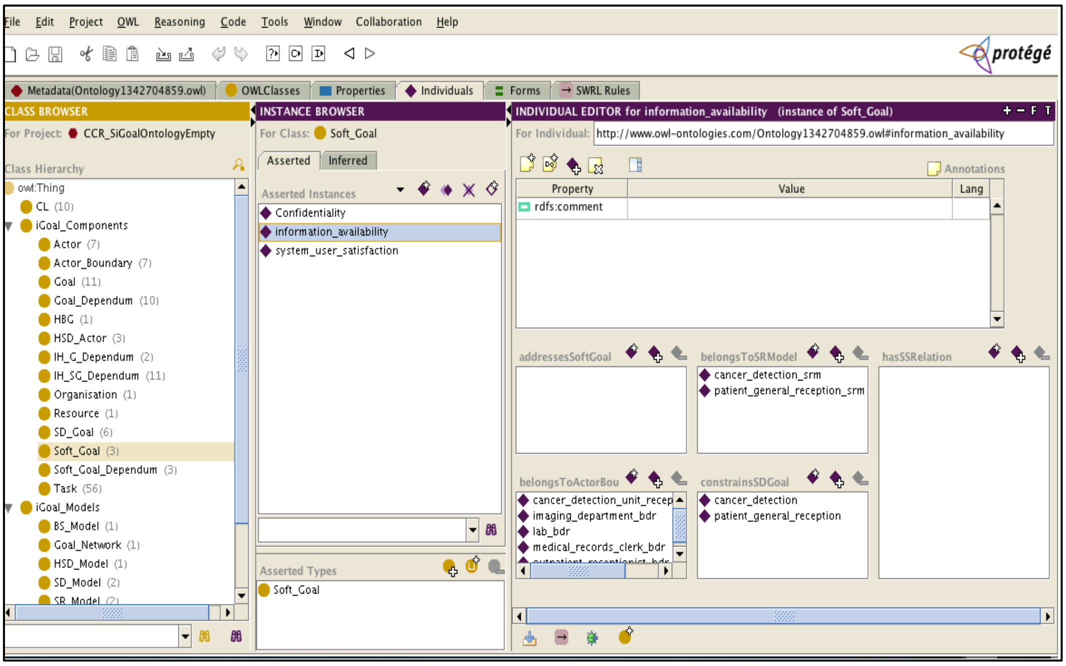


Figure M.12: An Example of Instantiating the sQuality Ontology to sQuality_CCR Ontology.

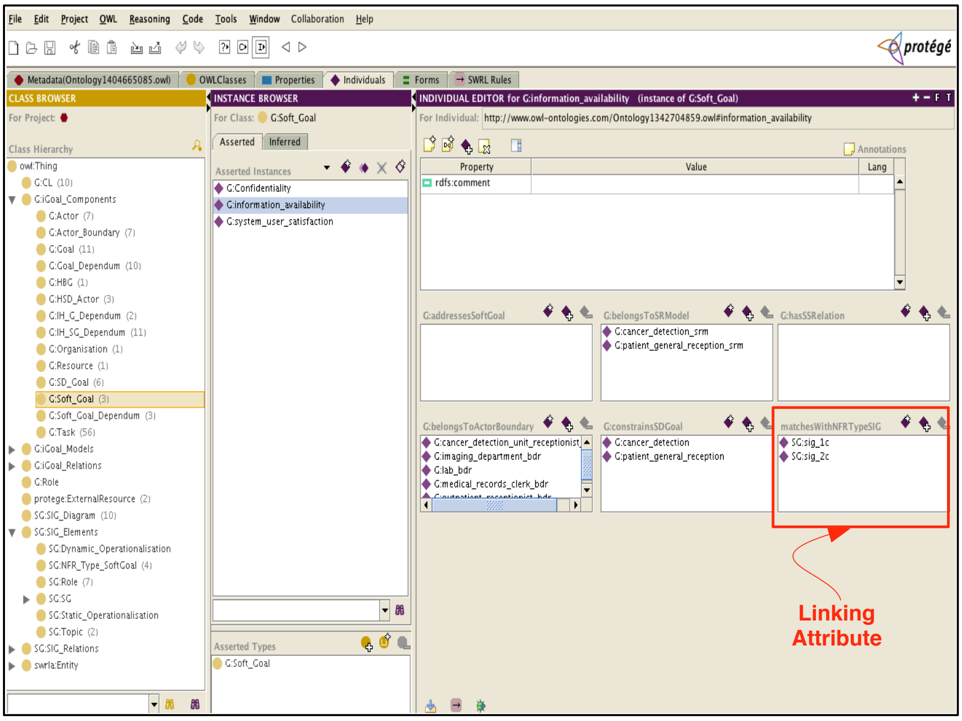


Figure M.13: A Protégé Editor Window that Shows an Example of Linking the siGoal_CCR Ontology and the sQuality_CCR Ontology

Appendix N: Aligning the CCR's BSV with the As-is CCR Riva-based BPA

This appendix presents the work of aligning the CCR's BSV with its As-is Riva BPA. According to the available input (i.e., CCR BSV and the as-is Riva BPA designed using the original approach), Algorithm 5.1 "Aligning the as-is Riva BPA, designed using the original, to the BSV of a particular Business Organisation" in (Section 5.8), appears appropriate in order to carry out this alignment. The CCR's BSV is attached in Appendix M, where the as-is Riva BPA is shown in Section N.1. The GQ-Riva BPA resultant from this alignment is shown in Section N.2.

N.1 Recalling the As-is Riva BPA for the CCR Representative Processes

Since this research has considered the first three processes (i.e., patient reception process, cancer detection process and the cancer treatment process) in the CCR are representative enough for the evaluation, then it is necessary to show the current Riva-based BPA designed using Ould's original method. Therefore, this section presents the current EBEs and UoWs list (shown in Table N.1), current UoW diagram (shown in Figure N.1) and current Riva-based BPA (shown in Figure N.2). Table N.1 shows about **52 EBEs** where **11 are UoWs** shown in bold text. The 11 UoWs are related through the dynamic generate relationships in order to design the UoW diagram as shown in Figure N.1.

Table N.1: The CCR's As-is EBE List in the Riva BPA [Source: (Yousef, 2010), Used with the author's permission].

Patient General reception	**Imaging test**
Receptionist (general)	Imaging department
Patient	Imaging test results
Medical records	Combined clinic
Appointment	**Patient treatment**
Patient file	Receptionist (outpatient clinic)
Emergency unit	**Outpatient clinic reception**
Cancer detection unit	Admission clerk
Database	Room availability
Patient details	Emergency case
Specialist	Waiting list
Cancer detection	Paper work
Receptionist (cancer detection unit)	Radiotherapy department
Doctor (diagnostician)	**Radiotherapy treatment**
Clinic	Chemotherapy department
Medical insurance	**Chemotherapy treatment**
Payment	Surgery
Clinical appraisal	**Inpatient care**
Notes	Nurses
History	Bed
Patient admission	Receptionist (inpatient care)
Investigations	Receptionist (admission department)
Lab test	Receptionist (Imaging department)
Lab	Receptionist (cancer detection)
Lab test results	Receptionist (laboratory)
Medical records clerk	Receptionist (chemo)
Receptionist (radio)	

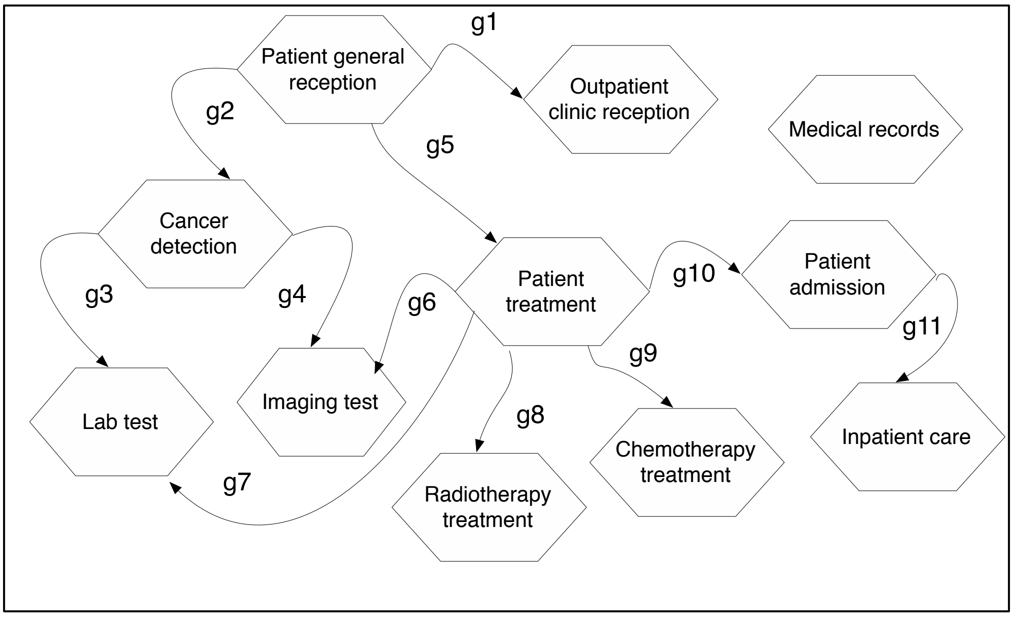


Figure N.1: The CCR UoW Diagram Designed Using the Original Riva BPA Method [Source: (Yousef, 2010), Used with the author's permission].

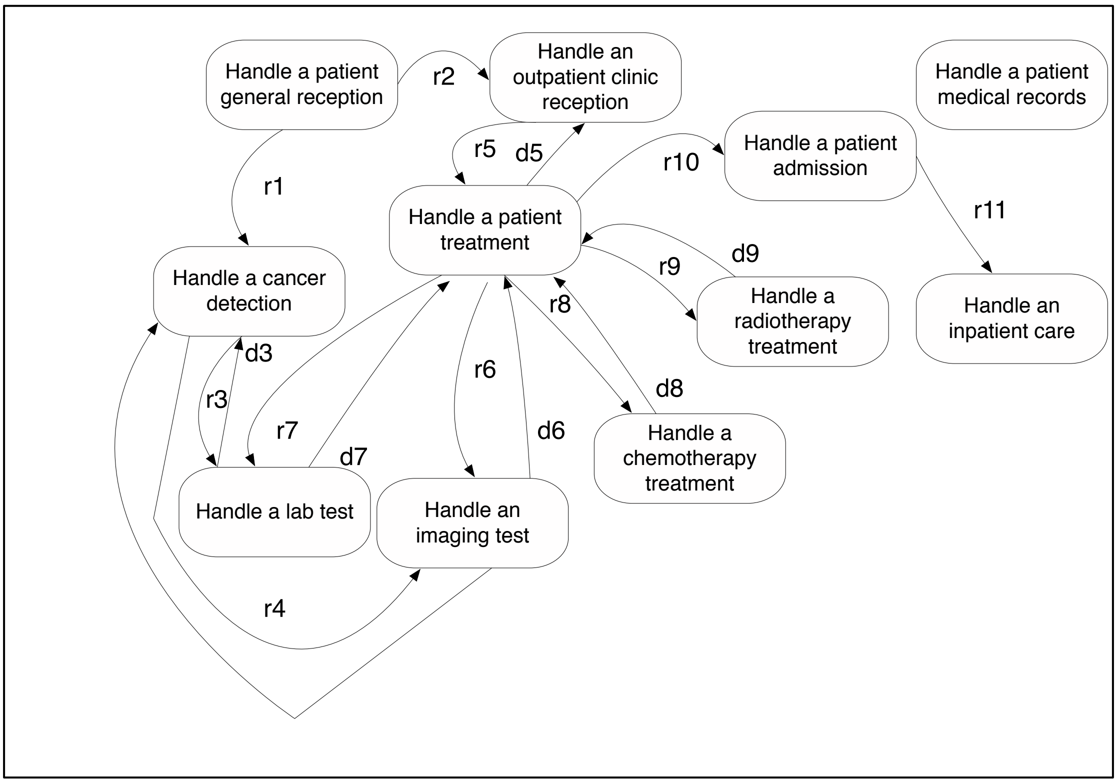


Figure N.2: The Current CCR Riva BPA Designed Using the Original Riva Method [Source: (Yousef, 2010), Used with the author's permission].

N.2 Deriving the CCR's GQ-Riva BPA from an Alignment of the BSV with the As-is Riva BPA

According to what has been presented in Chapter 5, the CCR's BSV (attached in Appendix M) is reused in order to carry out an alignment with the current or as-is CCR's Riva BPA. Therefore, the pre-alignment elements and models of the CCR's Riva BPA (i.e., as-is EBEs, as-is UoWs, as-is UoW diagram, etc.) should be present, as shown in the previous section of this appendix. Since the current CCR's Riva BPA was designed using the original Riva method in (Yousef, 2010), then the given case study meets the input of algorithm "Aligning the as-is Riva BPA, which is generated using the original approach, to the up-to-date BSV for a particular organisation" (Algorithm 5.1 in Chapter 5 Section 5.8) and should be applied in order to generate the CCR's to-be GQ Riva BPA. Recalling that in this algorithm, the alignment of an as-is Riva BPA with a BSV is substantially based on detecting the as-is EBEs and deriving the to-be ones owing to that the EBEs are the fundamental blocks of the Riva method. In this section, the GQ-Riva BPA resultant from the alignment algorithm 5.1 is presented.

The as-is EBEs list of the CCR BPA appears in Table N.1, where the bolded EBEs are the as-is UoWs (Yousef, 2010). The pre-alignment EBEs' list comprises of **52 EBEs** where **11 of them were determined as the as-is UoWs** (Yousef, 2010). The list in Table N.1 was derived from the CCR BP models without taking into account their hard goals and soft goals (Yousef, 2010). After applying algorithm 5.1, the resulted to-be EBEs list of CCR is comprised of the detected as-is EBEs (52 EBEs), in black text, and of the newly identified EBEs (25 EBEs), in red text, stemming from the CCR's BSV as shown in Table N.2. The **new 25 EBEs** have been filtered into **7 UoWs,** which are written in bolded red text in Table N.2. The 11 as-is UoWs remain in the post-alignment list in Table N.2. As overall, the entire to-be EBEs list contains about **77 EBEs** where **18** of them are UoWs, in bold, as shown in Table N.2. It is apparent that the as-is EBEs list is **subset** from the to-be EBEs list of the CCR study. This relation is justified because of the unchanged business in the CCR considering that the CCR's as-is EBE list only suffered from the absence of its complement set of EBEs derived using CCR's BSV. Therefore, it is claimed that there is no gap of analysis found in the elicitation of the as-is EBEs for the to-be EBEs. The as-is EBEs were found goal-based (i.e., based on the subset relationship from the to-be EBEs), yet they missed their complement that are the new 25 EBEs. The EBQs that constrain the post-alignment EBEs are derived from the BSV using the same algorithm 5.1 and are shown in Table N.3.

The identified EBQs of the CCR's representative processes are either designed in its quality models (e.g., NFR catalogues) that are involved in the BSV or not designed in catalogues such as non-clinical quality requirements (e.g., empathy) (Aburub, 2006), yet they still exist in the BSV of the CCR. And this is a credit to the proposed BSV that includes all types of NFRs. The NFR catalogues are numbered in order to reference them in the relevant G-UoW and within resulted process in the GQ-Riva BPA. This is because a one NFR type may be differently interpreted and designed for each UoW and process. Therefore, it is preferred to identify the NFR type with a pointer or reference to the catalogue (i.e., SIG diagram) in order to facilitate traceability and linking. For example, the information availability NFR catalogue in the processes that stem from the cancer detection process is different from the ones that stem from the cancer treatment process as attached in Appendix F (Aburub, 2006). The EBQs are elaborated into Q-UoWs. The entire GQ-UoW diagram is shown in Figure N.3, where the associated individual representation of each G-UoW is shown in Figures N.4 - N.21. For example, lab test is classified as an as-is UoW and detected UoW in the to-be UoWs. It is constrained with 8 EBQs that are elaborated into 8 Q-UoWs where 4 of them are represented using the NFR framework. A representation of the lab test GQ UoW is shown in Figure N.11. The difference between the as-is UoW diagram (Figure N.1) and the to-be UoW (Figure N.3) of the selected case study is apparently manifested in the integration of goals (i.e., new UoWs) and quality requirements (i.e., in the Q-UoW). Table N.4 summaries the information of the BPs that constitute the GQ Riva BPA shown in Figure N.22. In Table N.4, the entries are colored with the reference to the colors in the GQ 2nd cut architecture in Figure N.22. The presence of the detailed representation of BPs allowed classifying the BPs into: novel (in pink color), as-is process with required redesign (in yellow color) and as-is process with no required redesign (in white color) as shown in Figure N.22 and Table N.4.

Table N.2: The To-be EBE for the CCR Process

Patient General reception	**Imaging test**
Receptionist (general)	Imaging department
Patient	Imaging test results
Medical records	Combined clinic
Appointment	**Patient treatment**
Patient file	Receptionist (outpatient clinic)
Emergency unit	**Outpatient clinic reception**
Cancer detection unit	Admission clerk
Database	Room availability
Patient details	Emergency case
Specialist	Waiting list
Cancer detection	Paper work
Receptionist (cancer detection unit)	Radiotherapy department
Doctor (diagnostician)	**Radiotherapy treatment**
Clinic	Chemotherapy department
Medical insurance	**Chemotherapy treatment**
Payment	Surgery
Clinical appraisal	**Inpatient care**
Notes	Nurses
History	Bed
Patient admission	Receptionist (inpatient care)
Investigations	Receptionist (admission department)
Lab test	Receptionist (Imaging department)
Lab	Receptionist (cancer detection)
Lab test results	Receptionist (laboratory)
Medical records clerk	Receptionist (chemo)
Receptionist (radio)	Statistical reports
Patient registration	**Cancer type and site identification**
Appointment booking	Contradictory diagnoses
Reception patient diagnose check	Treatment information
Library	Imaging test appointment
Diagnose details	Lab test appointment
Specialist timetables	Admission
Patient personal details	Admission duration
Doctor timetables	Expected discharging dates
Search keys	Inpatient treatment
Admission file	**Patient diagnose check**
Patient diagnose	**Adequate treatment plan**
Treatment team	

Table N.3: The EBQs for the CCR Process

EBQs for the CCR Case Study
1- Empathy
2- Assurance
3- Responsiveness
4- Reliability
5- Information availability
6- Confidentiality
7-System user satisfaction
8- Information domain
9-Quick/Prompt processing

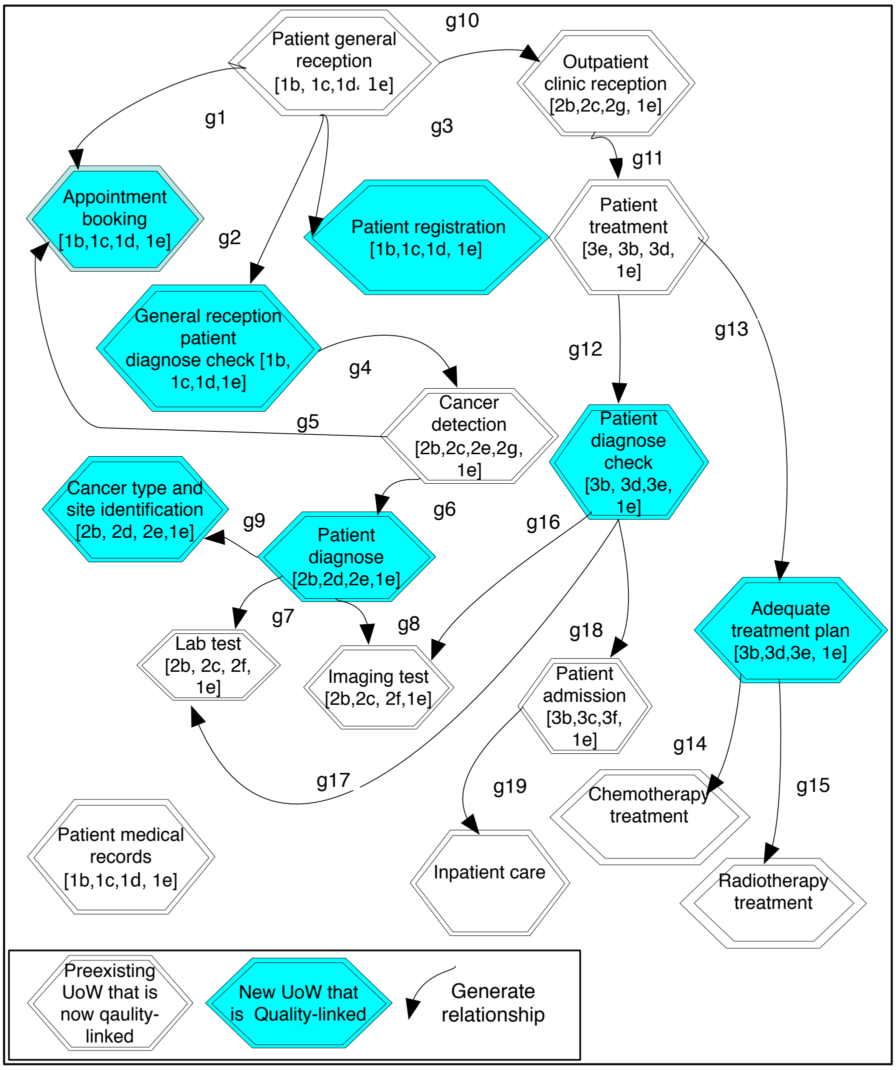


Figure N.3: The GQ-UoW Diagram for the CCR Case Study.

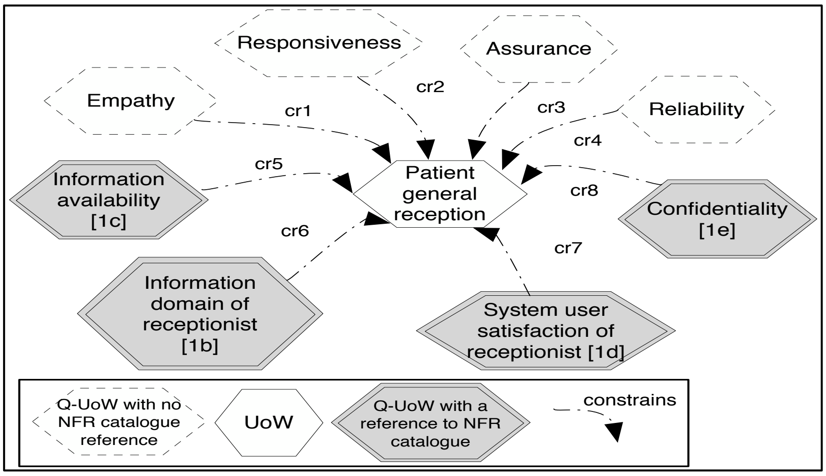


Figure N.4: The Q-UoWs for the Patient General Reception G-UoW.

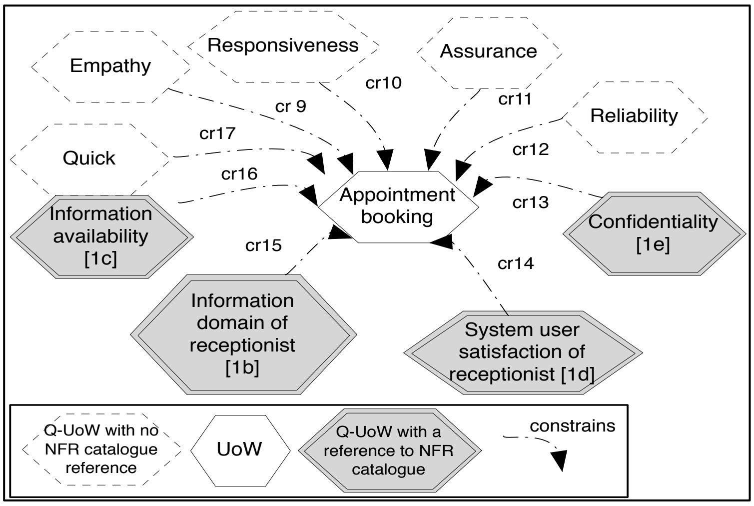


Figure N.5: The Q-UoWs for the Appointment Booking G-UoW.

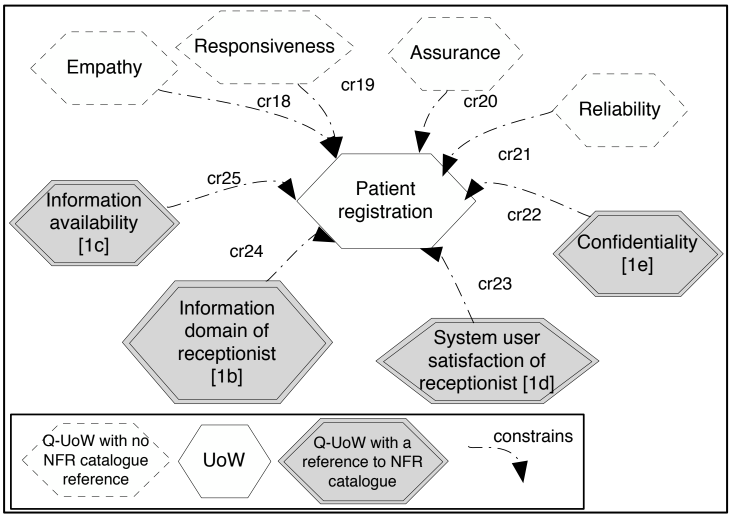


Figure N.6: The Q-UoWs for the Patient Registration G-UoW.

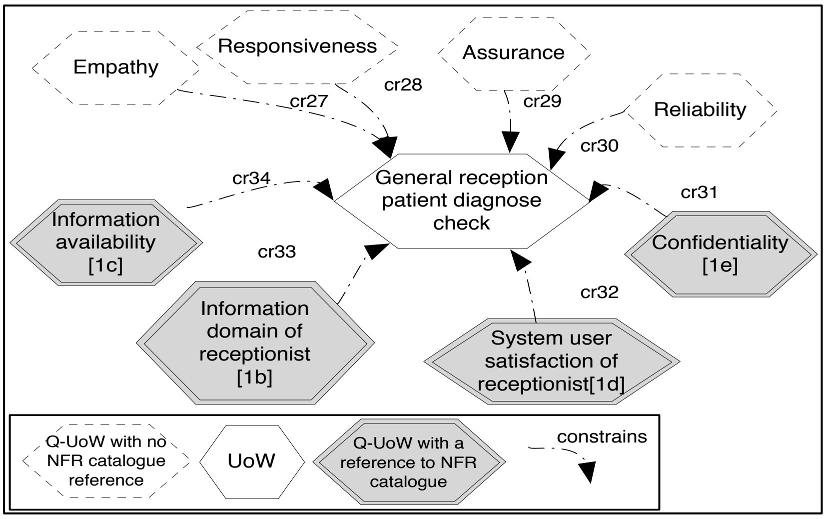


Figure N.7: The Q-UoWs for the General Reception Patient Diagnose Check G-UoW.

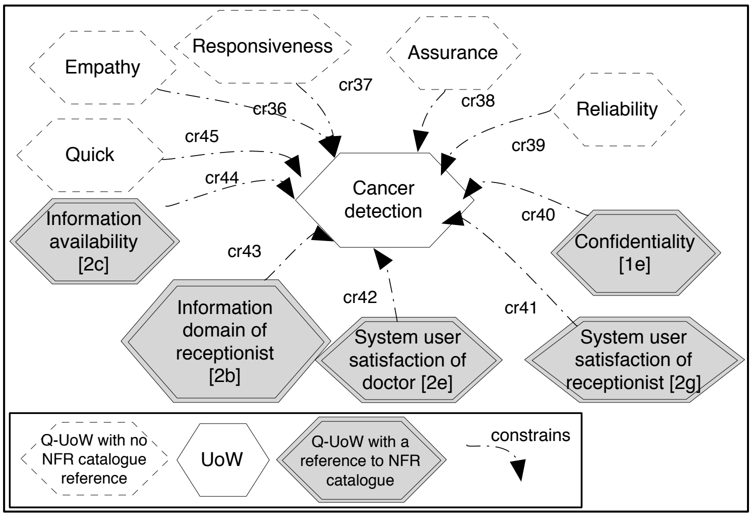


Figure N.8: The Q-UoWs for the Cancer Detection G-UoW.

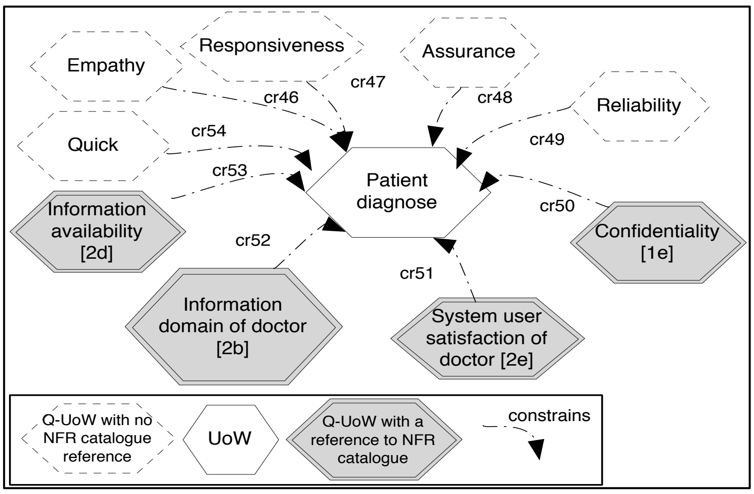


Figure N.9: The Q-UoWs for the Patient Diagnose G-UoW.

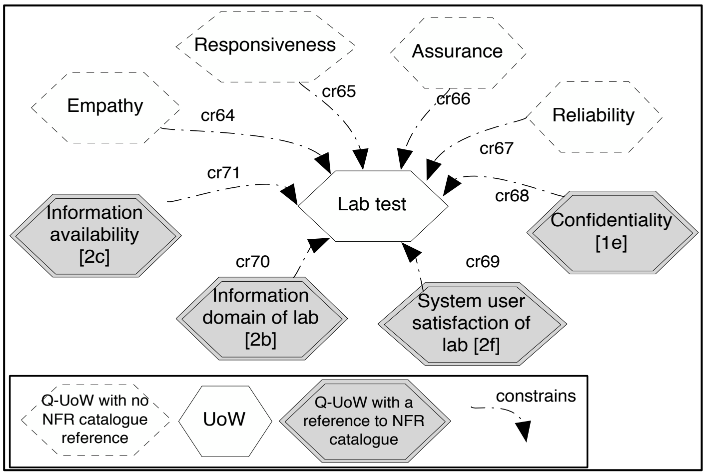


Figure N.10: The Q-UoWs for the Cancer Type and Site Identification G-UoW.

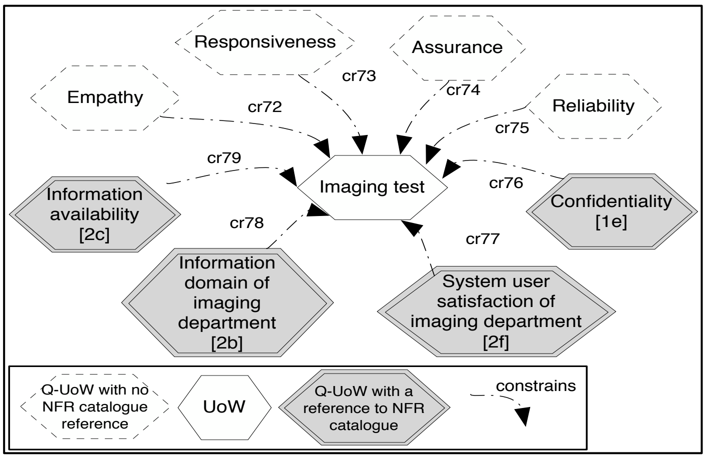


Figure N.11: The Q-UoWs for the Lab Test G-UoW.

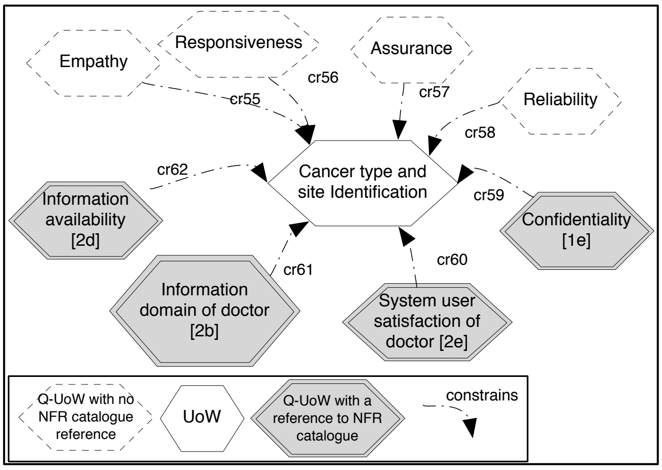


Figure N.12: The Q-UoWs for the Imaging Test G-UoW.

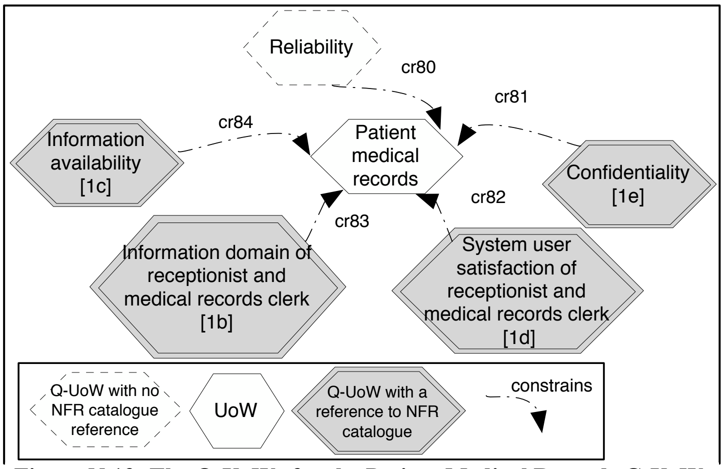


Figure N.13: The Q-UoWs for the Patient Medical Records G-UoW.

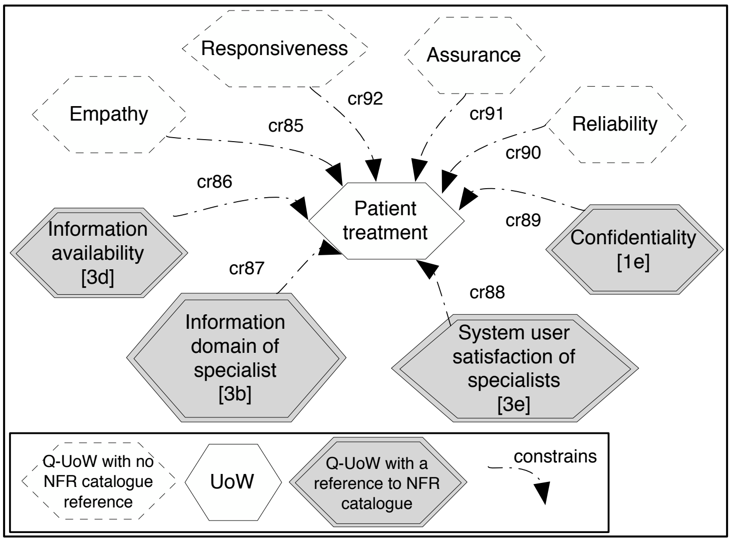


Figure N.14: The Q-UoWs for the Cancer Treatment G-UoW.

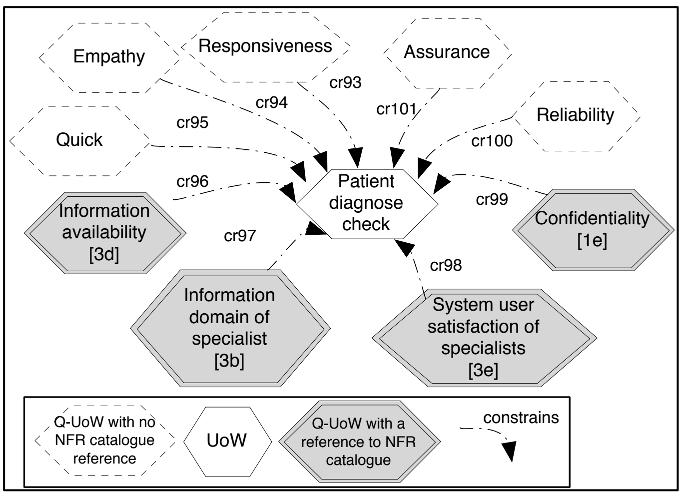


Figure N.15: The Q-UoWs for Patient Diagnose Check G-UoW.

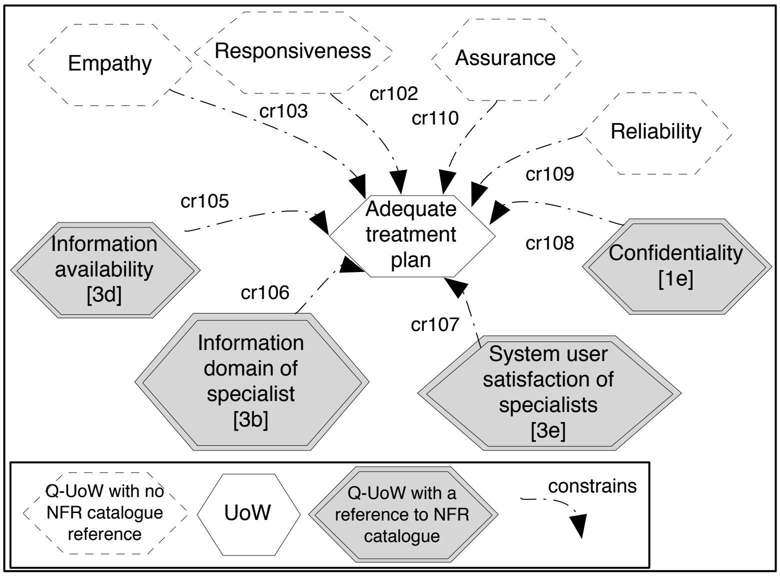


Figure N.16: The Q-UoWs for the Adequate Treatment Plan G-UoW.

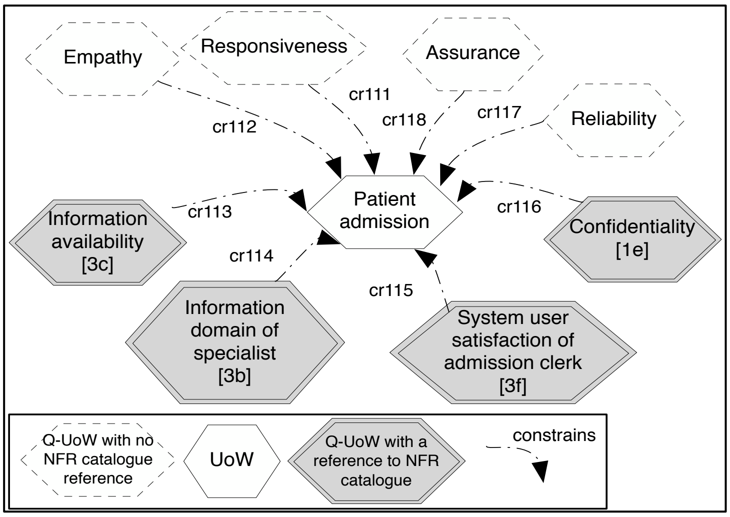


Figure N.17: The Q-UoWs for the Patient Admission G-UoW

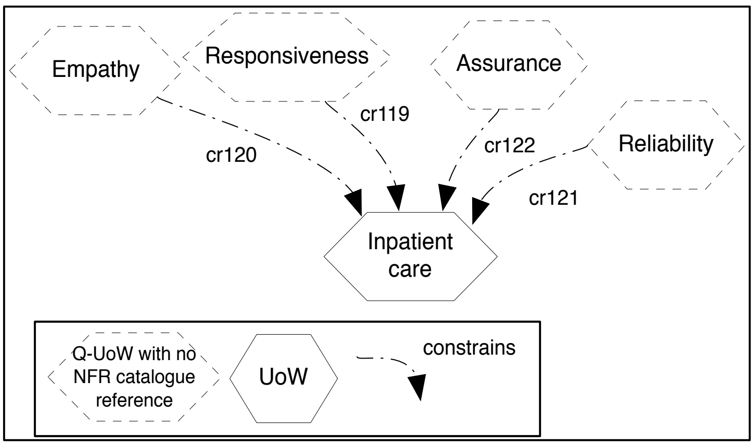


Figure N.18: The Q-UoWs for the Inpatient Care G-UoW

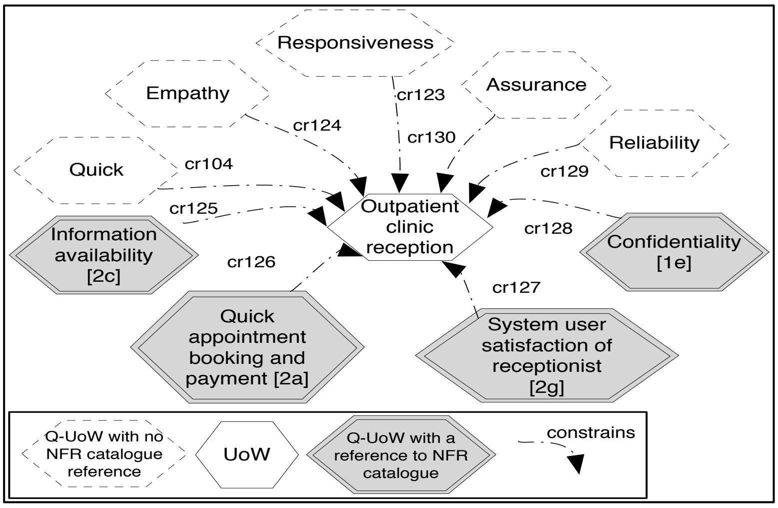


Figure N.19: The Q-UoWs for the Outpatient Clinic Reception G-UoW

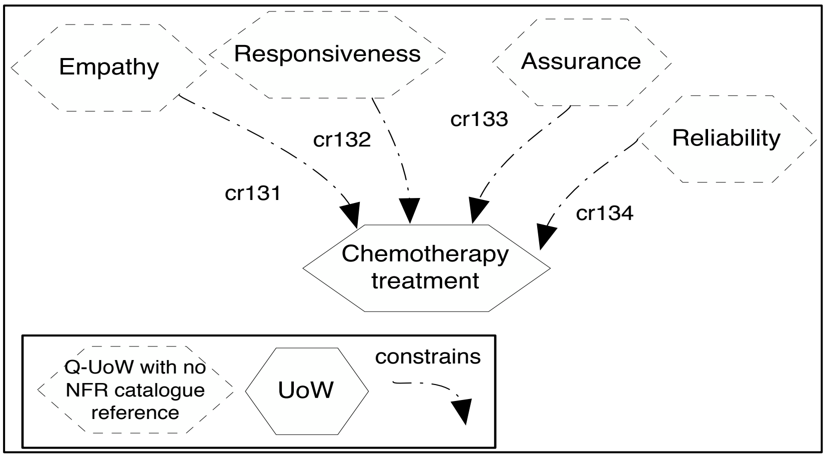


Figure N.20: The Q-UoWs for the Chemotherapy Treatment G-UoW

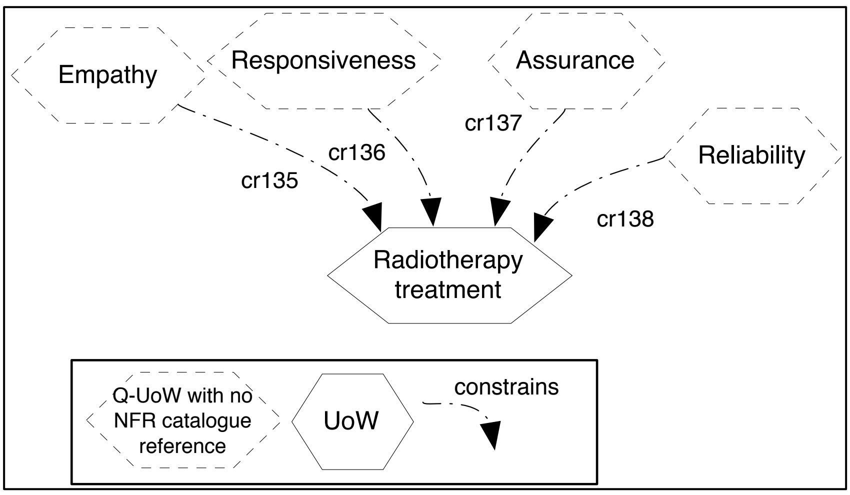


Figure N.21: The Q-UoWs for the Radiotherapy Treatment G-UoW

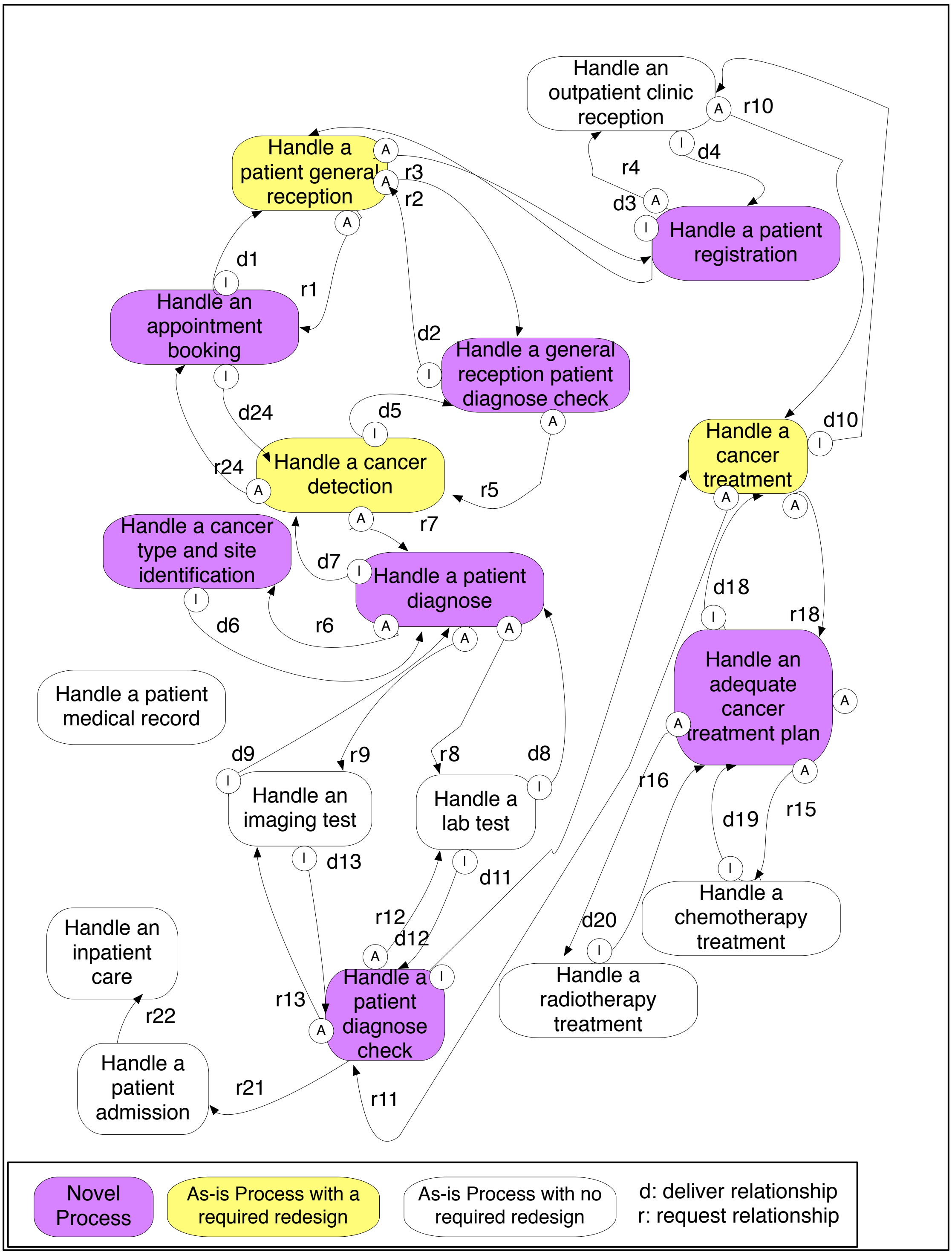


Figure N.22: The GQ-Riva BPA for the CCR Process.

Table N.4: The GQ Process in the CCR GQ Riva-based BPA

Process no	Process Name	Alignment Implication	Goals of the Process	Quality of the Process
1	Handle a patient general reception	As-is redesigned BP	GoP: Patient reception Contribute to fulfill: 1- Improving administration of cancer treatment.	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Information availability [1c]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.
2	Handle an appointment booking	Novel BP	GoP: Appointment booking Contribute to fulfill: 1-Improving administration of cancer treatment 2- Patient general reception	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Fast/reliable booking. 4- Information availability. [1c] 5-Confidentiality [1e]. 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability.
3	Handle a patient registration	Novel BP	GoP: Diagnosed patient is registered. Contribute to fulfill: 1-Improving administration of cancer treatment 2- Patient general reception	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Fast/reliable booking. 4- Information availability. [1c] 5-Confidentiality [1e]. 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability.
4	Handle a general reception patient diagnose check	Novel BP	GoP: General reception patient diagnose is checked. Contribute to fulfill: 1-Improving administration of cancer treatment 2- Patient general reception	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Information availability. [1c] 4-Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.

Table N.4 (Cont'd): The GQ Process in the CCR GQ Riva-based BPA

Process no	Process Name	Alignment Implication	Goals of the Process	Quality of the Process
5	Handle a cancer detection	As-is redesigned BP	GoP: Cancer detection Reception. Contribute to fulfill: 1-Improving administration of cancer treatment	1- System user satisfaction of doctor [2e]. 2- System user satisfaction of receptionist [2g]. 3- Information availability [2c]. 4- Reliable/fast appointment request and payment method. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability. 9-Confidentiality [1e]. 10-Information domain of receptionist [2b].
6	Handle a patient diagnose	Novel BP	GoP: Patient is diagnosed for cancer detection. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer detection.	1- Fast appointment request. 2- Information domain of doctor [2b]. 3- System user satisfaction of doctor [2e]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability. 9- Information availability [2d].
7	Handle the identification of cancer type and site	Novel BP	GoP: Cancer type and site are determined. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer detection.	1- Information domain of doctor [2b]. 2- System user satisfaction of doctor [2e]. 3- Confidentiality [1e]. 4- Information availability [2d]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.
8	Handle a patient treatment	As-is redesigned BP	GoP: Patient reception for cancer treatment. Contribute to fulfill: 1-Improving administration of cancer treatment.	1- System user satisfaction of specialists [3e]. 2- Information domain of receptionist [3b]. 3- Empathy. 4- Responsiveness. 5- Assurance. 6- Reliability. 7- Information availability [3d]. 8- Confidentiality [1e].

Table N.4 (Cont'd): The GQ Process in the CCR GQ Riva-based BPA

Process no	Process Name	Alignment Implication	Goals of the Process	Quality of the Process
9	Handle a patient diagnose check	Novel BP	GoP: Patient diagnose is checked. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Patient treatment.	1- Reliable/fast appointment requesting method. 2- Information domain. [3b] 3- Information availability [3d]. 4- System user satisfaction. [3e]. 5- Confidentiality. [1e] 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability.
10	Handle an adequate treatment plan	Novel BP	GoP: Adequate treatment is received. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Patient treatment.	1- Reliability. 2- Information domain. [3b] 3- Information availability [3d]. 4- System user satisfaction. [3e]. 5- Confidentiality. [1e] 6- Empathy. 7- Responsiveness. 8- Assurance.
11	Handle an outpatient-clinic reception	As-is BP with no required redesign	GoP: outpatient clinic reception. Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Cancer detection.	1- System user satisfaction of receptionist [2g]. 2- Information availability [2c]. 3-Reliable/fast-appointment request and payment method. 4- Empathy. 5- Responsiveness. 6- Assurance. 7- Reliability. 8- Confidentiality [1e].
12	Handle a lab test	As-is BP with no required redesign	GoP: a lab test is performed Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient diagnose.	1- Information domain of lab [2b]. 2- Information availability [2c]. 3-System user satisfaction of lab [2f]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.

Table N.4 (Cont'd): The GQ Process in the CCR GQ Riva-based BPA

Process no	**Process Name**	**Alignment Implication**	**Goals of the Process**	**Quality of the Process**
13	Handle an imaging test	As-is BP with no required redesign	GoP: an imaging test is performed Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient diagnose.	1- Information domain of imaging department [2b]. 2- Information availability [2c]. 3-System user satisfaction of imaging department [2f]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.
14	Handle a chemotherapy treatment	As-is BP with no required redesign	GoP: Patient chemotherapy treated. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Patient treatment. 3-Adequate treatment plan is received.	1- Empathy. 2- Responsiveness. 3- Assurance. 4- Reliability.
15	Handle a radiotherapy treatment	As-is BP with no required redesign	GoP: Patient radiotherapy treated. Contribute to fulfill: 1-Improving administration of cancer treatment. 2- Cancer treatment. 3- Adequate treatment plan is received. .	1- Empathy. 2- Responsiveness. 3- Assurance. 4- Reliability.
16	Handle an inpatient care	As-is BP with no required redesign	GoP: Inpatient care. Contribute to fulfill: 1-Improving administration of cancer treatment. 2-Patient treatment.	1- Empathy. 2- Responsiveness. 3- Assurance. 4- Reliability.

Table N.4 (Cont'd): The GQ Process in the CCR GQ Riva-based BPA

Process no	Process Name	Alignment Implication	Goals of the Process	Quality of the Process
17	Handle a patient medical records	As-is BP with no required redesign	GoP: Medical record of patient is handled. Contribute to fulfill: 1-Improving administration about cancer treatment.	1- ID of receptionist and medical records. [1b] 2- System user satisfaction of medical records. [1d] 3- Information availability. [1c] 4- Confidentiality [1e]. 5- Reliability.
18	Handle a patient admission	As-is BP with no required redesign	PG: patient is admitted. Contribute to fulfill: 1-Improving collection about cancer cases.	1- Information domain of admission clerk [3b]. 2- Information availability [3c]. 3-System user satisfaction of admission clerk [3f]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability

Appendix O: Business Process Models Representing the Novel and the As-is Redesigned CPs for the CCR Case Study

In this appendix, the novel processes (i.e., presented in pink colour in CCR's GQ-Riva BPA in Appendix N) and the as-is processes with required redesign (i.e., presented in yellow colour in CCR's GQ-Riva BPA in Appendix N) are presented. The processes are modelled using the BPMN language using Omni Graffle Professional Tool. Section O.1 presents the novel processes, where Section O.2 presents the as-is redesigned processes.

O.1 The Novel Processes in the CCR GQ-Riva BPA

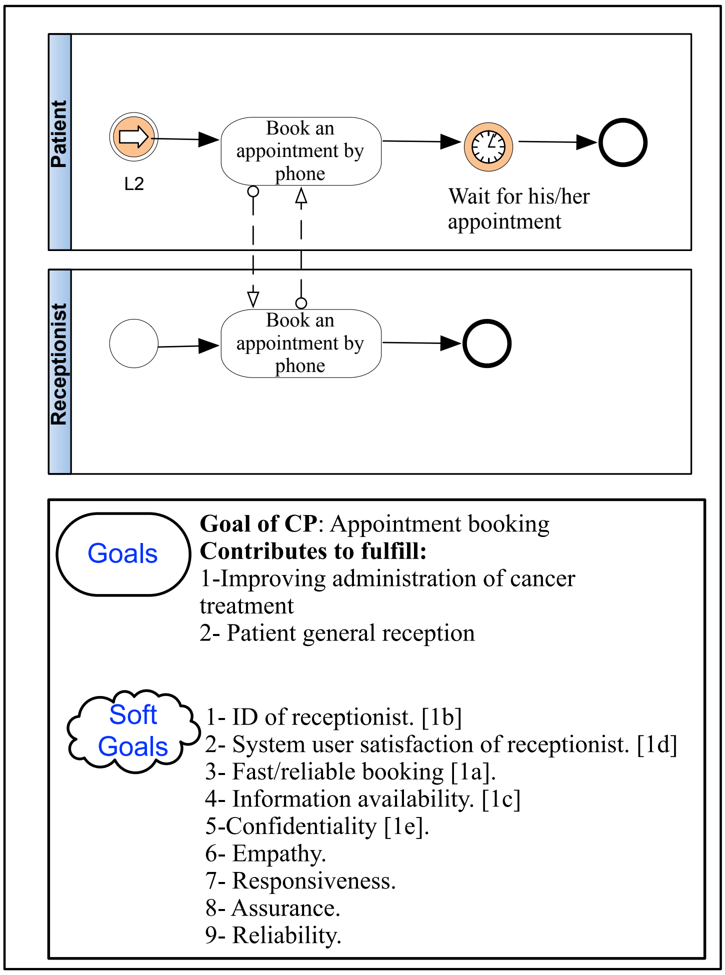


Figure O.1: CP2: Handle An Appointment Booking

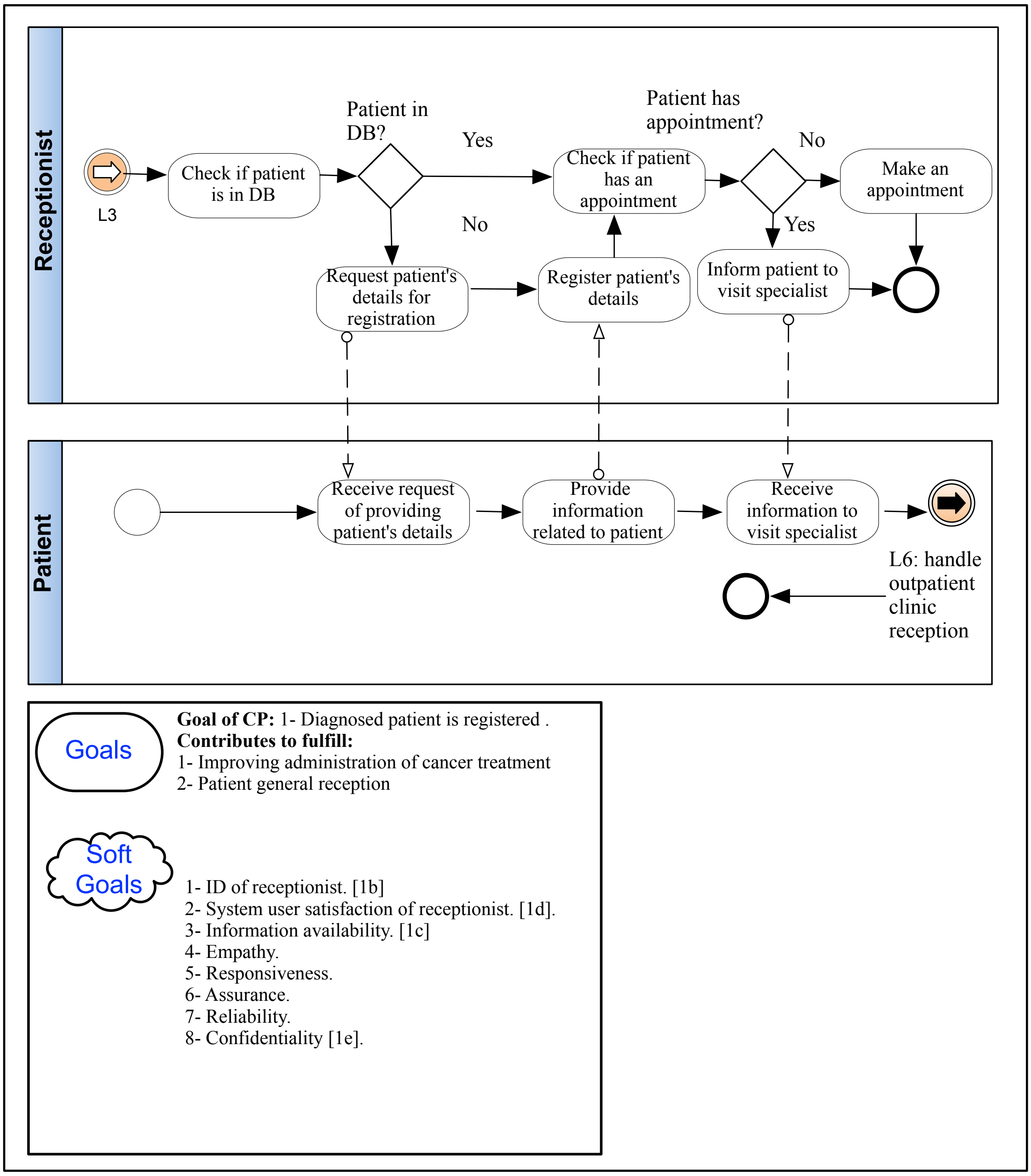


Figure O.2: CP3: Handle A Patient Registration

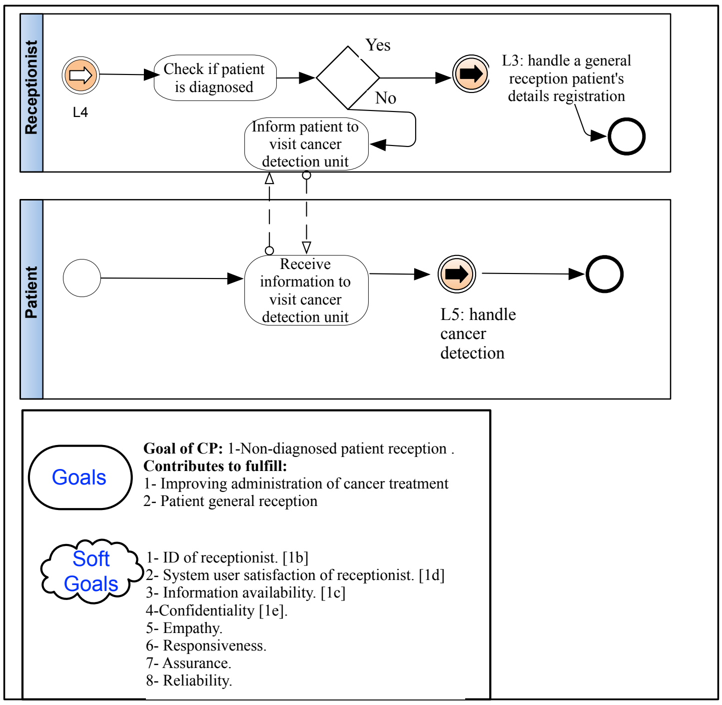


Figure O.3: CP4: Handle A General Reception Patient Diagnose Check

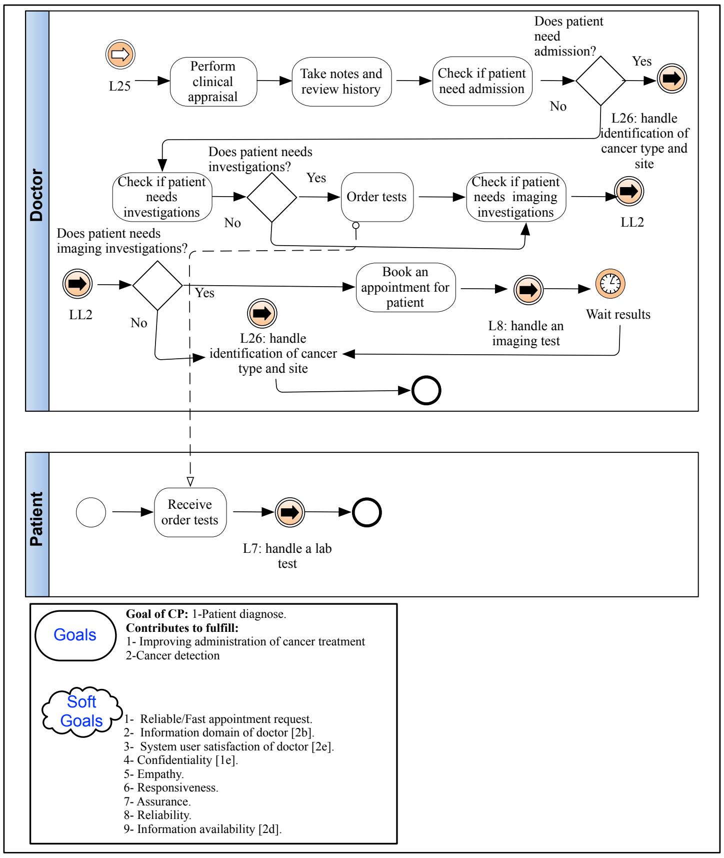


Figure O.4: C6: Handle a Patient Diagnose

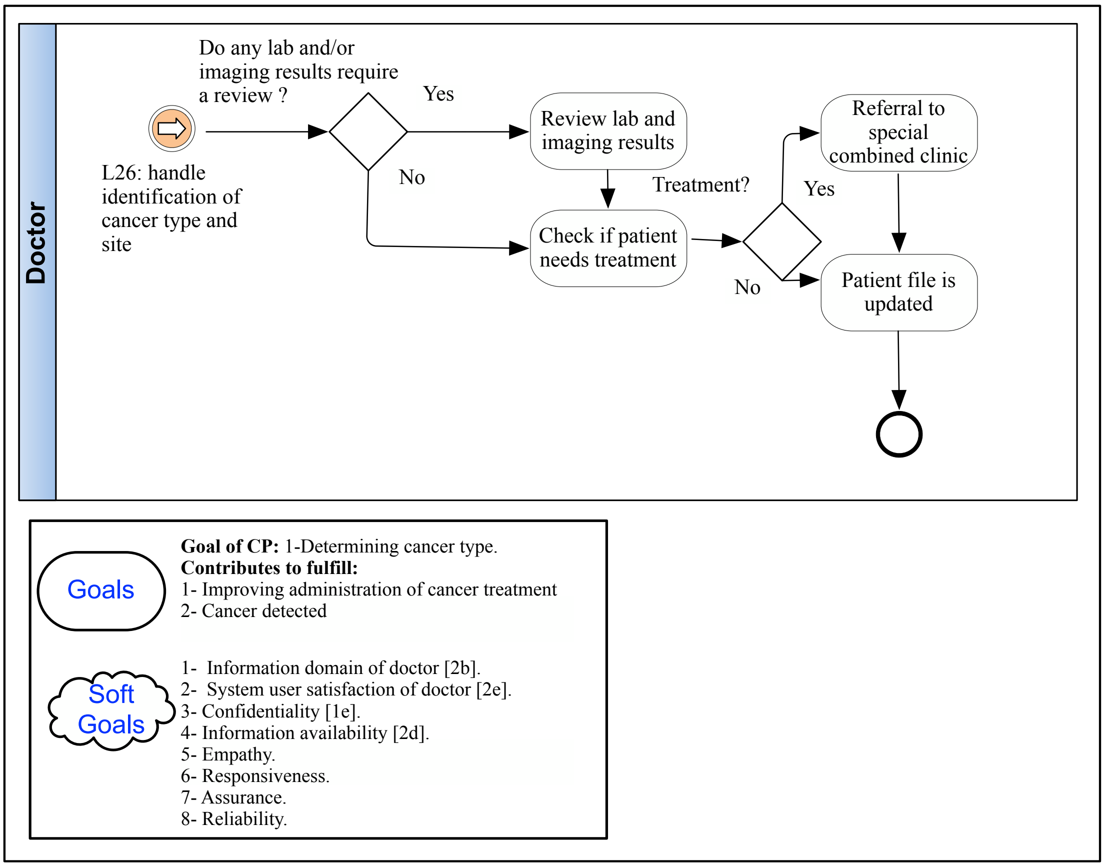


Figure O.5: CP7: Handle A Cancer Type and Site Identification.

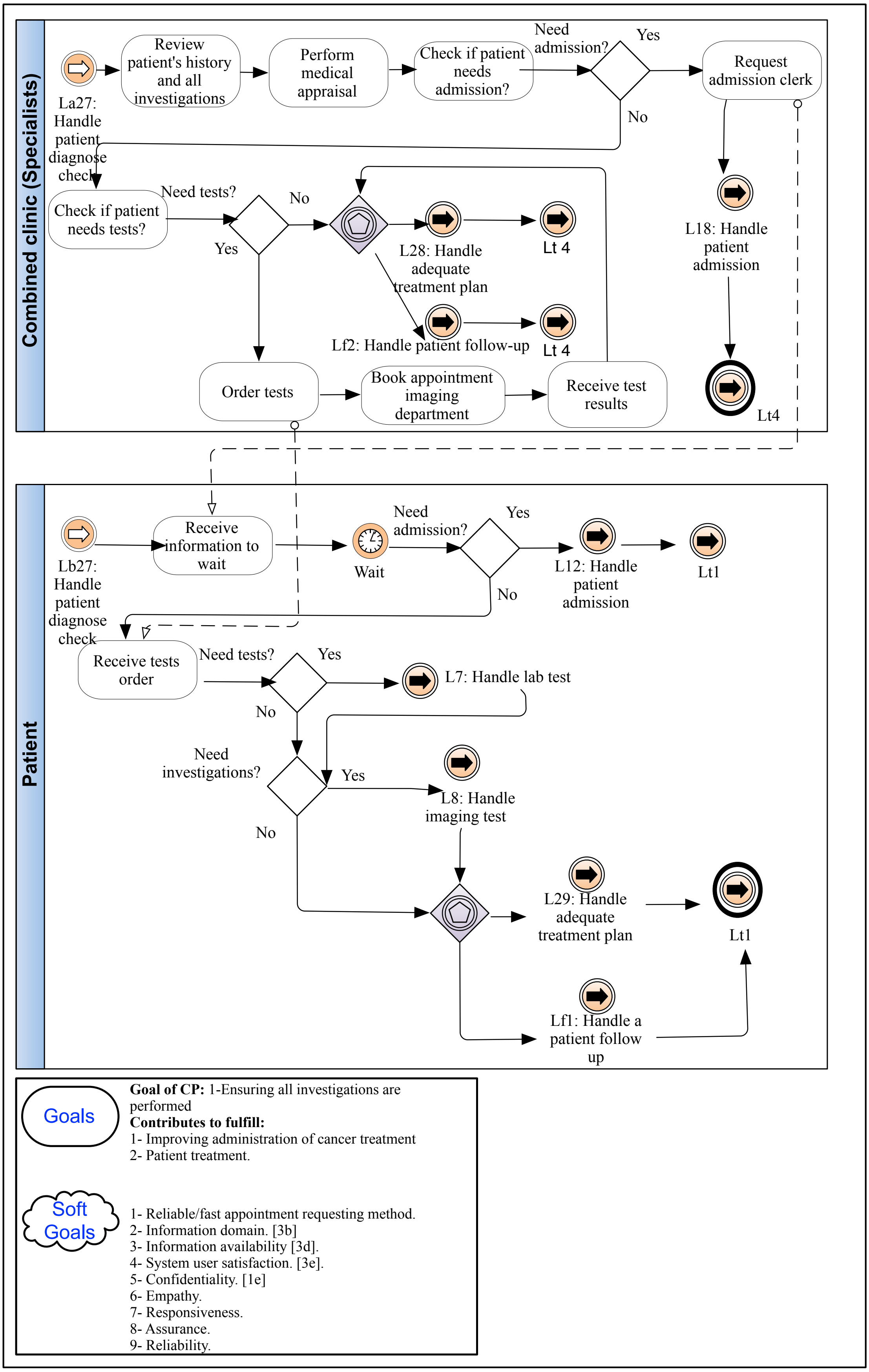


Figure O.6: CP9: Handle a Patient Diagnose Check.

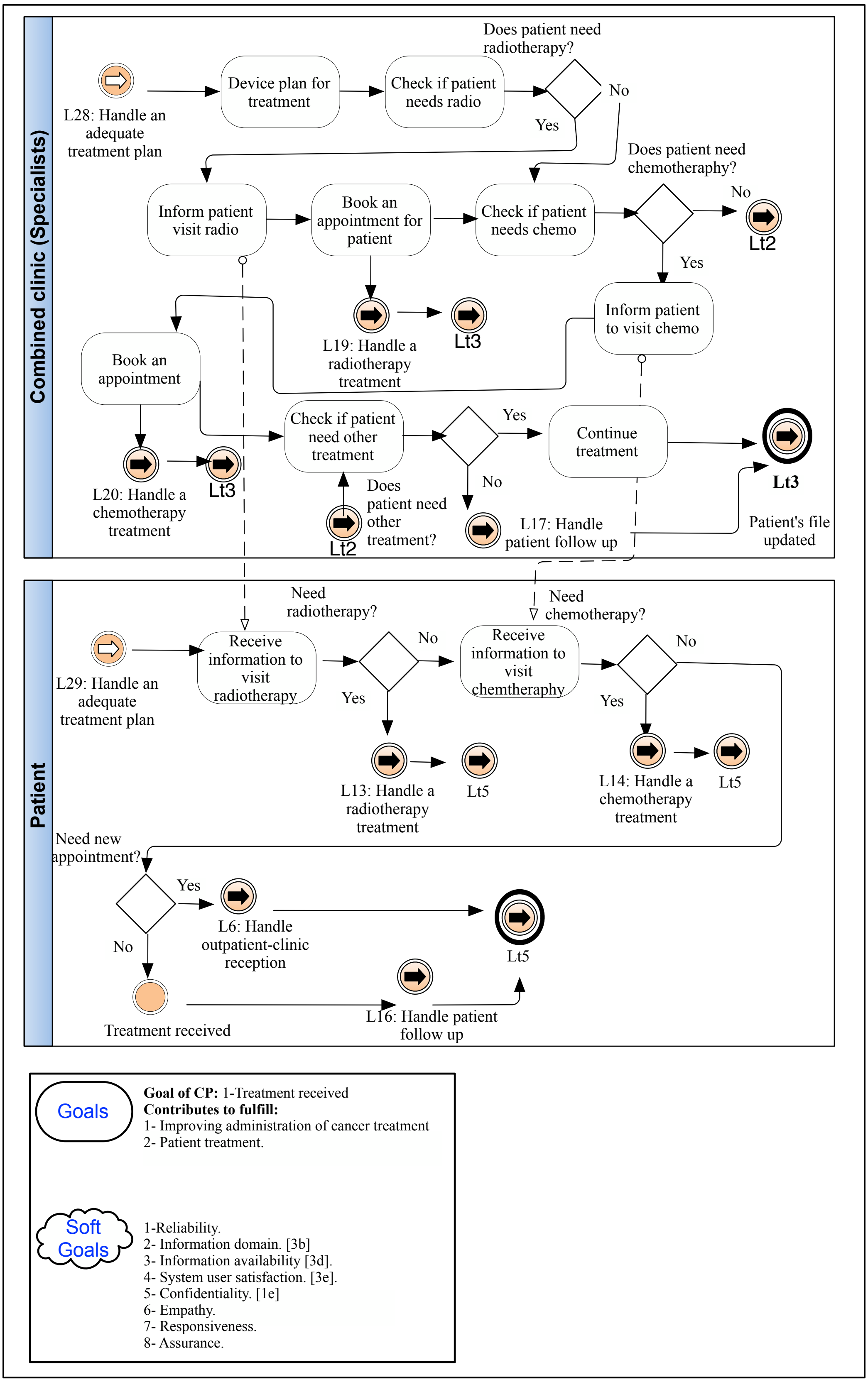


Figure O.7: CP 10; Handle An Adequate Treatment Plan.

O.2 The As-is Processes With Required Redesign in the CCR GQ-Riva BPA

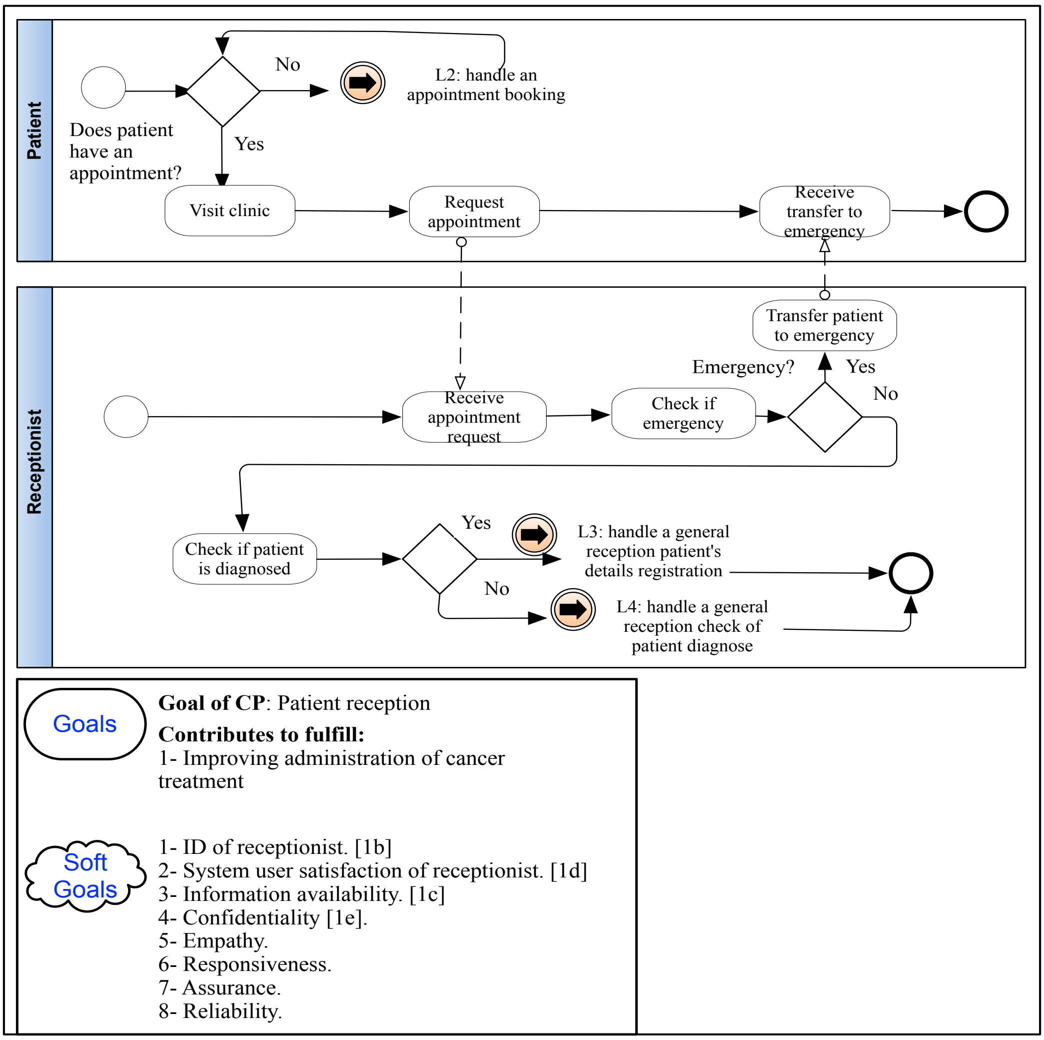


Figure O.8: CP1: Handle A Patient General Reception

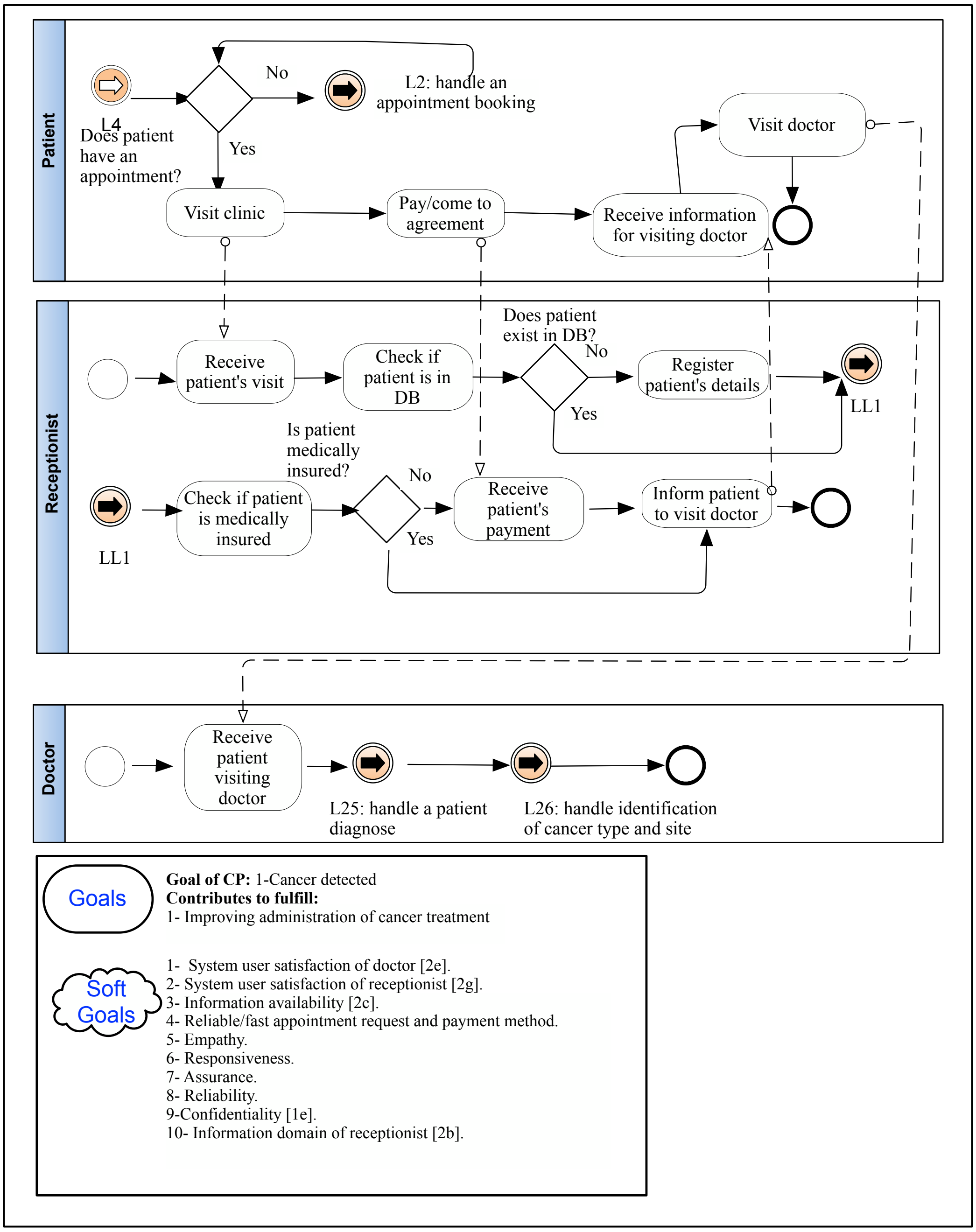


Figure O.9: CP5: Handle a Cancer Detection.

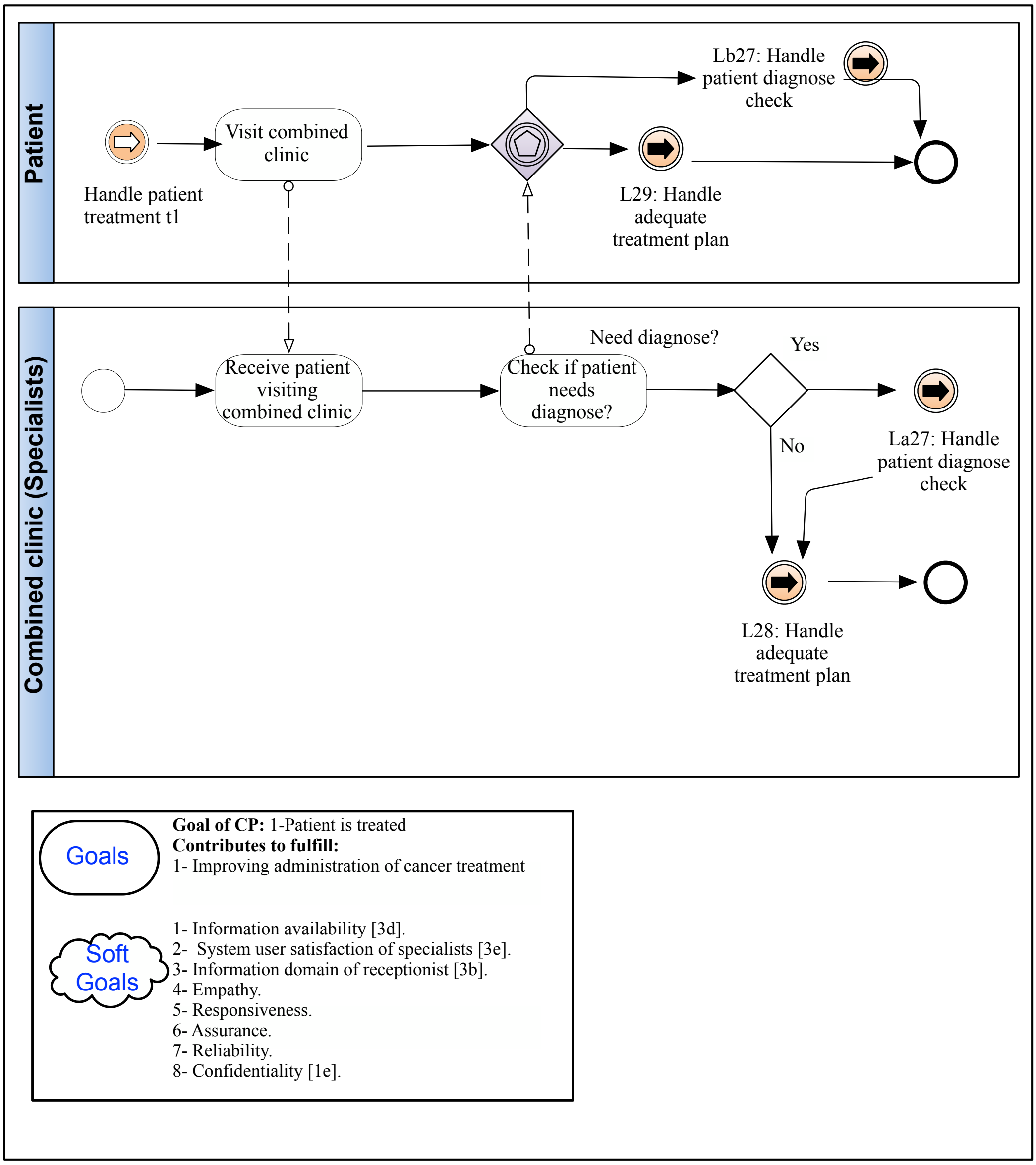


Figure O.10; CP 8: Handle A Patient Treatment.

Appendix P: Business Process Models Representing the As-is With No Required Redesign CPs for the CCR Case Study

In this appendix, the as-is processes with no required redesign (i.e., presented in white colour in CCR's GQ-Riva BPA in Appendix N) are presented. The processes are modelled using the BPMN language using Omni Graffle Professional Tool. This appendix presents 8 as-is processes (CPs) that involve light refinements regarding the integrating goals and quality requirements.

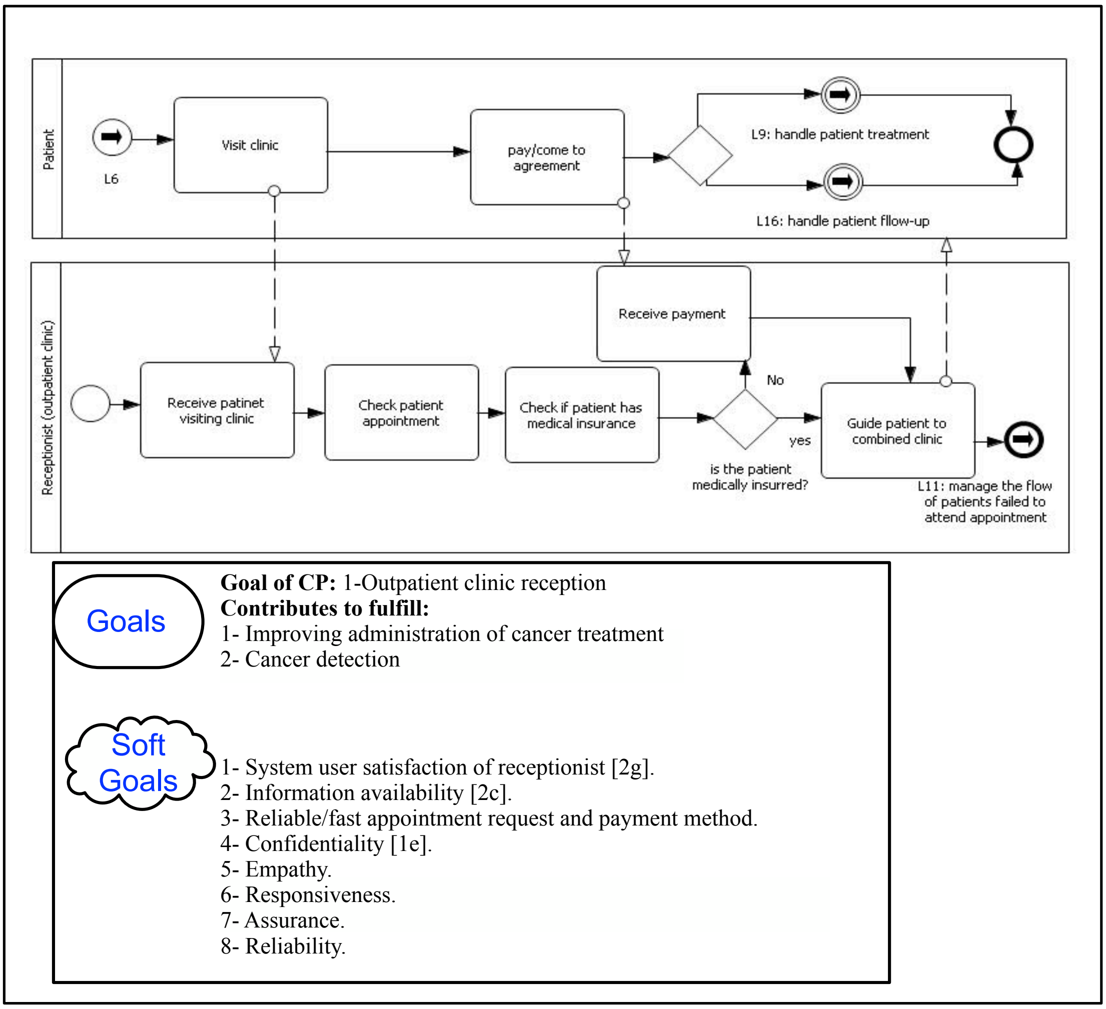


Figure P.1: CP11: Handle an Outpatient-Clinic Reception [Source: (Yousef, 2010), Adapted with the author's permission].

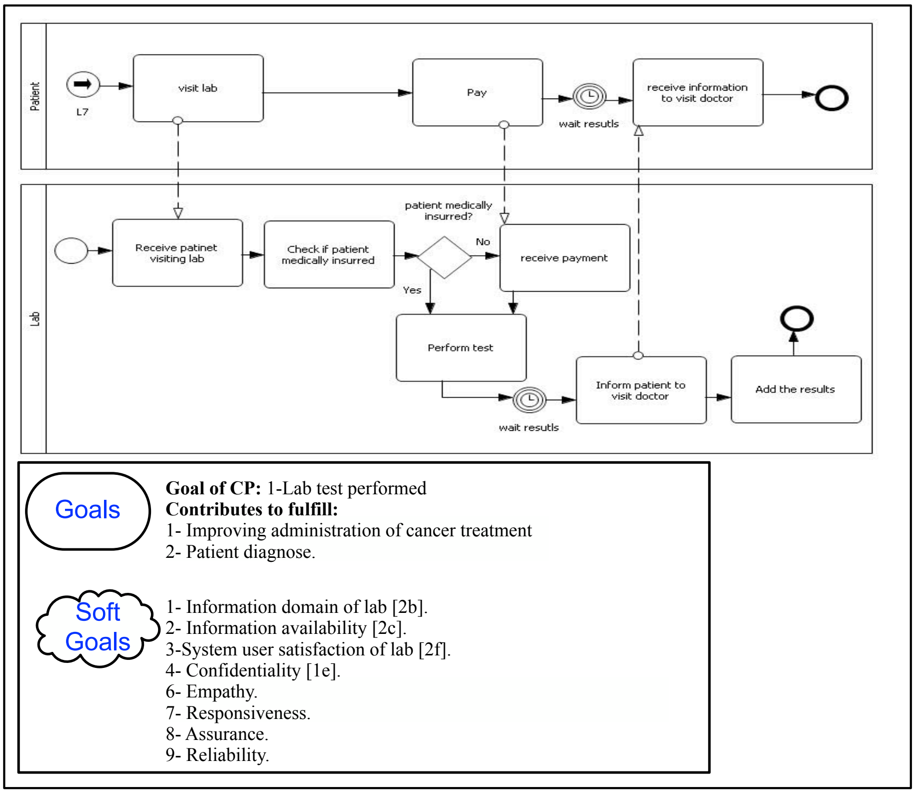


Figure P.2: CP12: Handle a Lab Test [Source: (Yousef, 2010), Adapted with the author's permission].

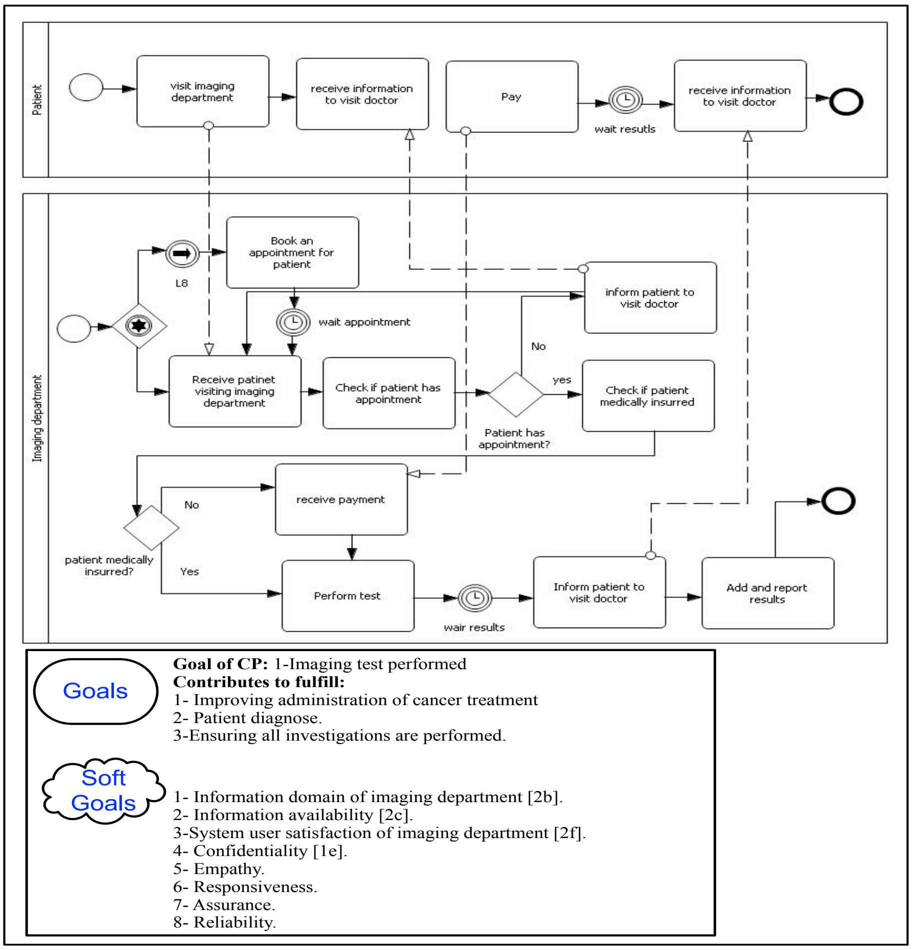


Figure P.3: CP13: Handle An Imaging Test [Source: (Yousef, 2010), Adapted with the author's permission].

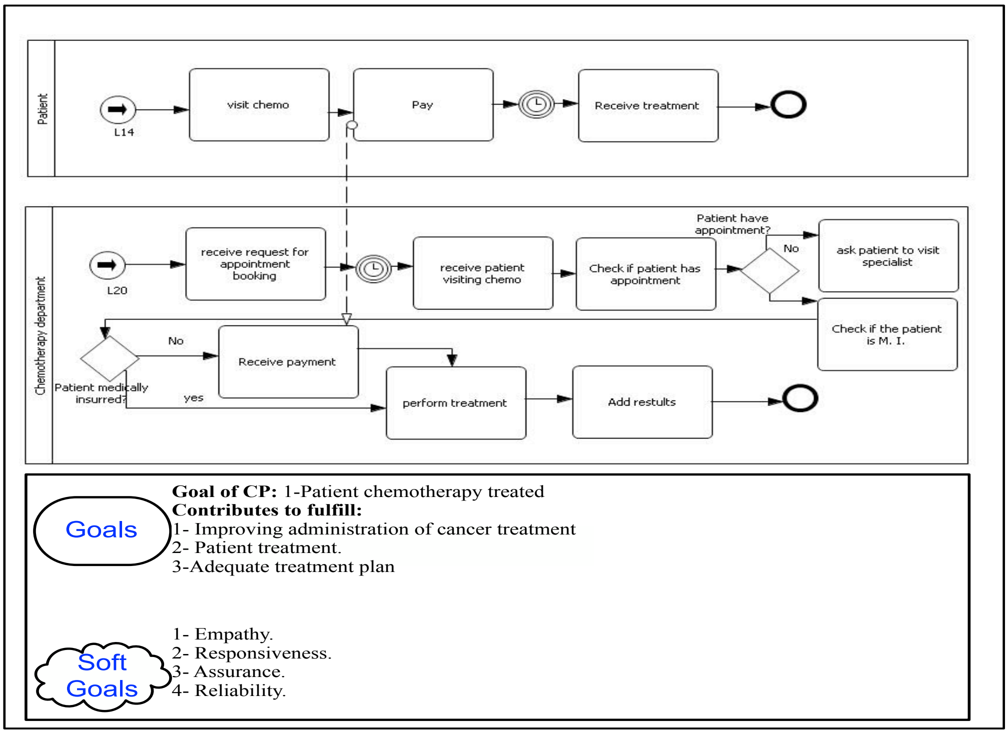


Figure P.4: CP14: Handle a Chemotherapy Treatment [Source: (Yousef, 2010), Adapted with the author's permission].

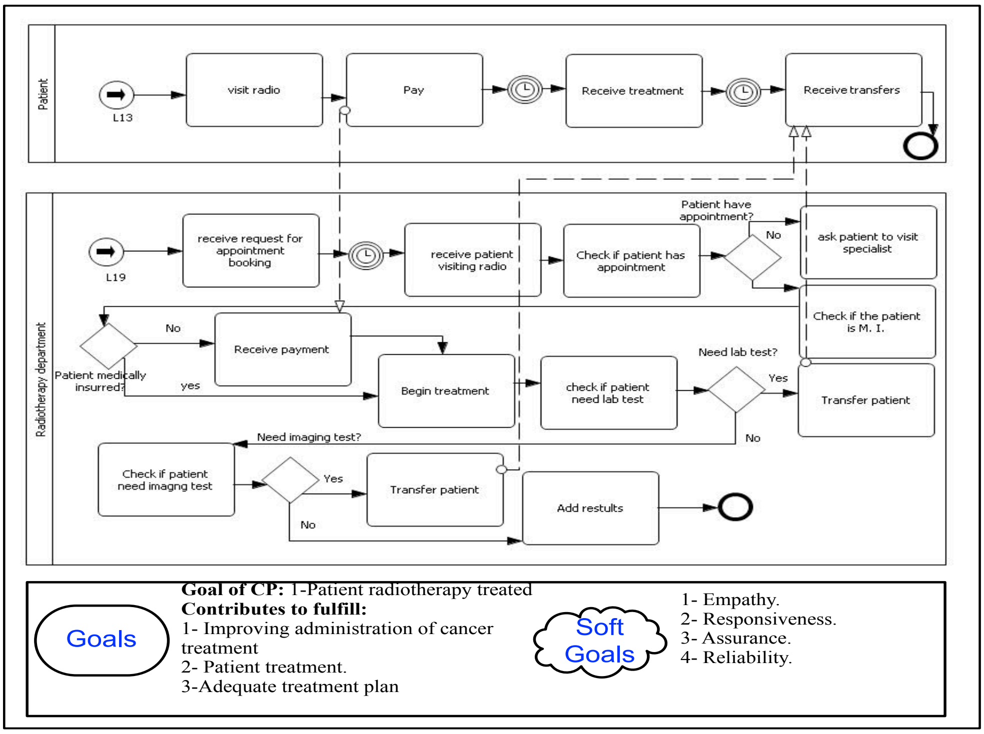
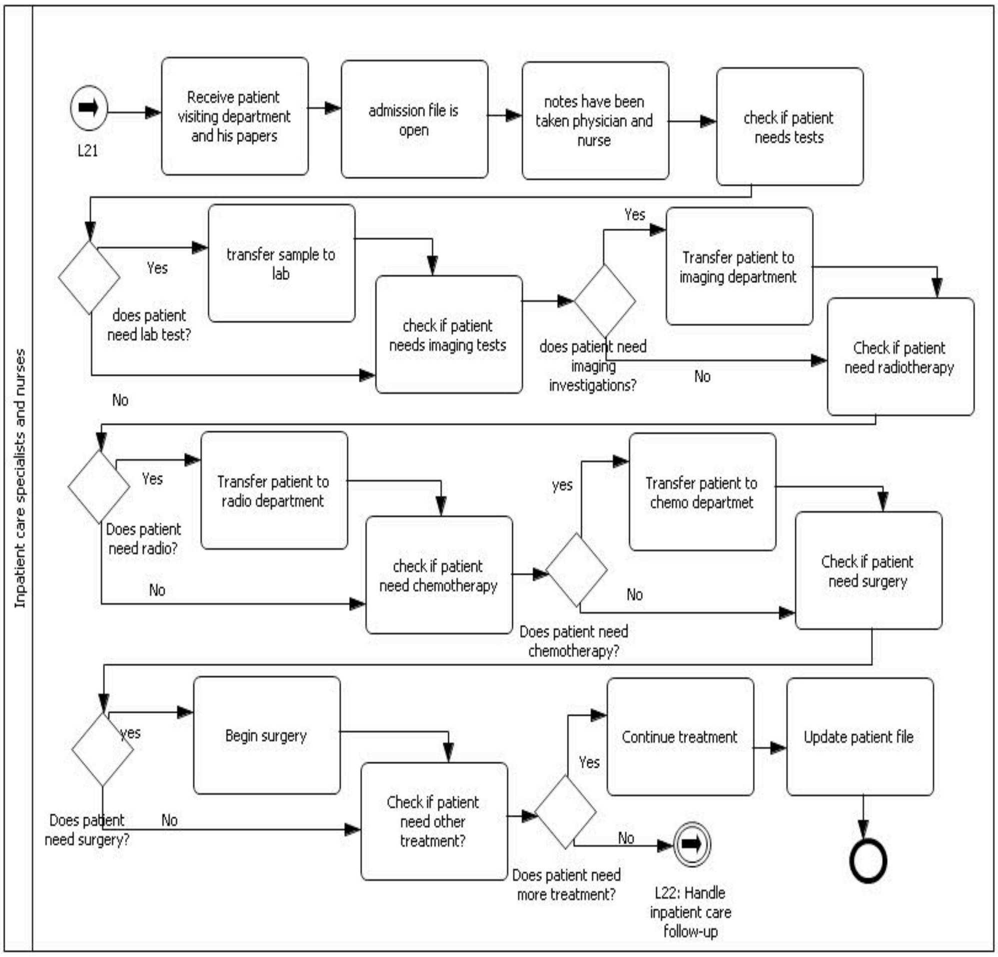


Figure P.5: CP15: Handle a Radiotherapy Treatment [Source: (Yousef, 2010), Adapted with the author's permission].



Goals

Goal of CP: 1-Inpatient care is received
Contributes to fulfill:
1- Improving administration of cancer treatment
2-Patient diagnose check.

Soft Goals

1- Empathy.
2- Responsiveness.
3- Assurance.
4- Reliability.

Figure P.6: CP16: Handle an Inpatient Care [Source: (Yousef, 2010), Adapted with the author's permission].

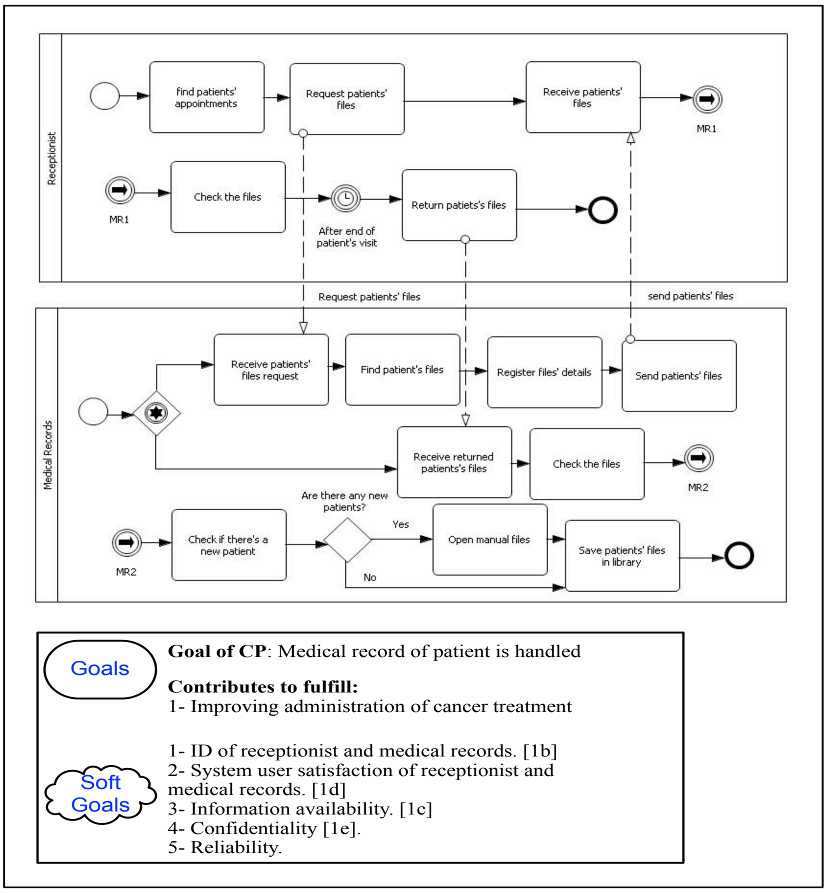


Figure P.7: CP17: Handle a Patient Medical Records [Source: (Yousef, 2010), Adapted with the author's permission].

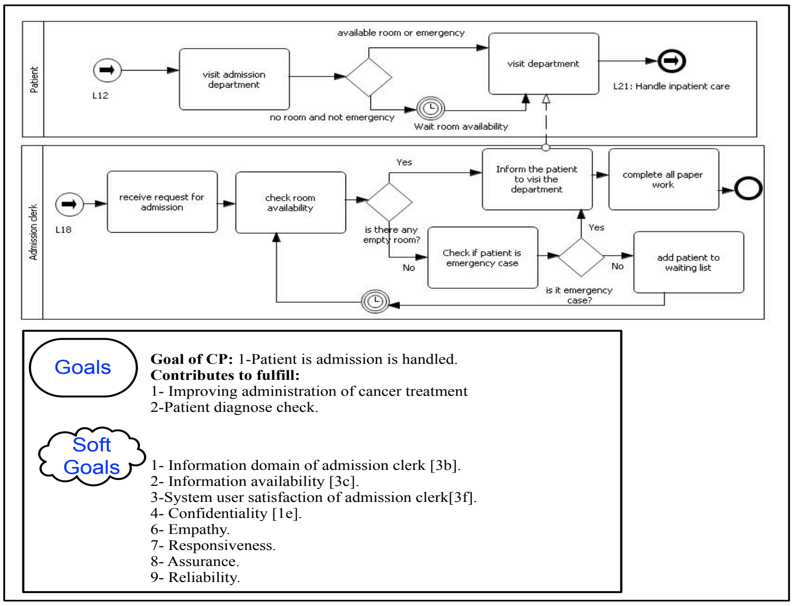


Figure P.8: CP18: Handle a Patient Admission [Source: (Yousef, 2010), Adapted with the author's permission].

Appendix Q: Pools, Goals, Soft Goals and Tasks for the GQ-BPMs that Represent GQ-CPs in the CCR's GQ-Riva BPA Resulted from an Alignment with CCR's BSV

The appendix presents the set of GQ-BPMs (i.e., 18 case processes) resultant from the alignment of the Riva BPA with the BSV for the representative processes in the CCR case study. Table Q.1 shows the goals, soft goals, pools and tasks for each GQ-CP. This table contains the required information for instantiating GQ-sBPMN_CCR. The classes (Process, Goal_Contribution, Goal_Of_Business_Process, Soft Goal, Tasks (UserTask, SendTask, ReceiveTask, ManualTask). It is important to highlight that the order of the instantiations of the classes does not necessarily that they appears in the same order in the GQ-BPM). The orange colored entries mean that they are omitted from the original as-is process in order to address the required redesign (i.e., alignment implication). The Green cells in Table Q.1 refer to the novel processes with new entries, where the white entries refer to the as-is processes with no required redesign.

Table Q.1: The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_Business_Process	Soft goal name	Q Ref		Task name	Task type
Handle Patient general reception	Administration of cancer treatment is improved	Patient reception	Empathy		Patient	Visit clinic	Manual
			Responsiveness			Request appointment	Send
			Assurance			Book appointment by phone	Send
			Reliability			Receive transfer to emergency	Receive
			Information domain	[1b]		Receive info to visit specialist	Receive
			System user satisfaction	[1d]		Receive info to visit cancer detection unit	Receive
			Information availability	[1c]	Receptionist	Receive appointment request	Receive
			Confidentiality	[1e]		Receive appointment booking by phone	Receive
						Check if emergency	Manual
						Transfer patient to emergency	Send
						Inform patient to visit cancer detection unit	Send
						Check if patient diagnosed	User

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals: Goal Contribution	Goals: Goal_Of_Business_Process	Soft goals: Soft goal name	Soft goals: Q Ref	Pools	Tasks: Task name	Tasks: Task type
Handle Patient general reception					Receptionist	Check if patient in DB	User
						Check if patient has appointment	User
						Inform patient to visit specialist	Send
						Register patient details	User
						Make appointment	User
Handle a appointment booking	Administrati-on of cancer treatment is improved	Appointment is booked	Empathy		Receptionist	Book an appointment by phone	Receive
	Patient general reception		Responsiveness		Patient	Book an appointment by phone	Send
			Assurance				
			Reliability				
			Information domain	[1b]			
			System user satisfaction	[1d]			
			Information availability	[1c]			
			Confidentiality	[1e]			
			Quick				
Handle a patient registration	Administration of cancer treatment is improved	Patient is registered	Empathy		Receptionist	Check if patient in DB	User
			Responsiveness			Check if patient has an appointment	User
			Assurance			Make appointment	User
	Patient reception		Reliability			Register patient details	User
			Information domain	[1b]		Request patient details for registration	Send
						Inform patient to visit specialist	Send
			System user satisfaction	[1d]	Patient	Receive request of patient details	Receive
			Information availability	[1c]		Provide information related to patients	Send
			Confidentiality	[1e]		Receive info to visit specialist	Receive
			Quick				
Handle a general reception patient diagnose check	Administration of cancer treatment is improved	General reception patient diagnose is checked	Empathy		Receptionist	Check if patient is diagnosed	User
	Patient general reception		Responsiveness			Inform patient to visit cancer detection unit	Send
			Assurance		Patient	Receive info to visit cancer detection unit	Receive
			Reliability				
			Information domain	[1b]			
			System user satisfaction	[1d]			

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_Business_Process	Soft goal name	Q Ref		Task name	Task type
Handle a general reception patient diagnose check			Information availability	[1c]			
			Confidentiality	[1e]			
Handle a cancer detection	Administration of cancer treatment is improved	Cancer is detected	Empathy		Patient	Book appointment by phone	User
			Responsiveness			Visit clinic	Manual
			Assurance			Pay/Come to an agreement	Manual
			Reliability			Receive information for visiting doctor	Receive
			Information domain	[2b]		Visit doctor	Manual
			System user satisfaction	[2e], [2g]		Receive test orders	Receive
			Information availability	[2c]	Receptionist (cancer detection unit)	Book appointment	User
			Confidentiality	[1e]		Receive patient visit clinic	Manual
			Quick			Check if patient is medically insured	User
						Check if patient in DB	User
						Receive patients payment	Manual
						Inform patient to visit doctor	Manual
						Register patient details	User
					Doctor (Diagnostician)	Receive patient visiting doctor	Manual
						Perform clinical appraisal	Manual
						Take notes and review history	Manual
						Check if patient needs admission	Manual
						Check if patient need investigations	Manual
						Order test	send
						Check if patient need imaging investigations	Manual
						Update patients file	User
						Book appointment for patient	Send
						Review lab and imaging results	Manual
						Refer patient to special combined clinic	Send

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	**Goals**		**Soft goals**		**Pools**	**Tasks**	
	Goal Contribution	**Goal_Of_Business_Process**	**Soft goal name**	**Q Ref**		**Task name**	**Task type**
Handle a patient diagnose	Administration of cancer treatment is improved	Patient is diagnosed	Empathy		Doctor	Perform clinical appraisal	Manual
	Cancer is detected		Responsiveness			Take notes and review history	Manual
			Assurance			Check if patient needs admission	Manual
			Reliability			Check if patient need investigations	Manual
			Information domain	[2b]		Order test	Send
			System user satisfaction	[2e]		Check if patient need imaging investigations	Manual
			Information availability	[2d]		Book appointment for patient	Send
			Confidentiality	[1e]	Patient	Receive test orders	Receive
			Quick	[2a]			
Handle identification of cancer type and site	Administration of cancer treatment is improved	Cancer type and site are identified	Empathy		Doctor	Check if lab and imaging test require review	User
	Cancer is detected		Responsiveness			Review lab and imaging results	Manual
			Assurance			Check if patient needs treatment	Manual
			Reliability			Update patients file	User
			Information domain	[2b]		Referral to specialized combined clinic	User
			System user satisfaction	[2e]			
			Information availability	[2d]			
			Confidentiality	[1e]			
Handle a patient treatment	Improving administration of cancer treatment	Patient reception for cancer treatment	System user satisfaction of specialists	[3e]	Patient	Visit combined clinic	Manual
			Information domain of receptionist	[3b]		Receive information to wait	User
			Information availability	[3d]		Receive test order	Receive
			Confidentiality	[1e]		Receive information to visit radio	Receive
			Empathy			Receive information to visit chemo	Receive
			Responsiveness		Combine clinic (specialists)	Receive patient visiting to combined clinic	User
			Assurance			Check if patient needs diagnose	User
			Reliability			Review patient's history and investigations	User

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_Business_Process	Soft goal name	Q Ref		Task name	Task type
						Check if patient need admission	Manual
						Request admission	Send
						Order tests	Send
						Book imaging appointment	User
						Receive test results	Receive
						Device plan for treatment	User
						Check if patient needs radiotherapy	Manual
						Inform patient to visit radiotherapy department	Send
						Book appointment for radiotherapy treatment	Send
						Check if patient needs chemotherapy	Manual
						Inform patient to visit chemotherapy department	User
						Book appointment for chemotherapy treatment	Send
						Check if patient need other treatment	Manual
						Continue treatment	Manual
Handle a patient diagnose check	1-Improving administration of cancer treatment.	Patient diagnose is checked.	Reliable/fast appointment		Combined clinic (specialists)	Review patient's history and investigations	User
			Information domain.	[3b]		Perform medical appraisal	User
	2- Patient treatment		Information availability	[3d]		Check if patient need admission	Manual
			System user satisfaction.	[3e]		Request admission	Send
			Confidentiality	[1e]		Check if patient needs tests	Manual
			Responsiveness			Order tests	Send
			Assurance			Book imaging appointment	User
			Reliability			Receive test results	Receive
			Empathy		Patient	Receive information to wait	User
						Receive test order	Receive
Handle Adequate Treatment Plan	Improving administration of cancer treatment.	Adequate treatment is received	Reliability.		Combined clinic (specialists)	Device treatment plan	User
			Information domain	[3b]		Check if patient needs radiotherapy	Manual
			Information availability	[3d]		Inform patient to visit radiotherapy department	Send
			System user satisfaction	[3e]		Book appointment for radiotherapy treatment	Send
			Confidentiality	[1e]		Inform patient to visit chemotherapy department	User
	Patient treatment		Empathy			Book appointment for chemotherapy treatment	Send

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_Business_Process	Soft goal name	Q Ref		Task name	Task type
			Responsiveness			Book appointment for chemotherapy treatment	Send
						Check if patient need other treatment	Manual
						Continue treatment	Manual
			Assurance		Patient	Receive information to visit radio	Receive
						Receive information to visit chemo	Receive
Handle outpatient clinic reception	Improving administration of cancer treatment.	Outpatient clinic reception	System user satisfaction of receptionist	[2g]	Patient	Visit clinic	Manual
			Information availability	[2c]		Pay/come to an agreement	Manual
	Cancer detection		Reliable/fast-appointment request and payment method		Receptionist (Outpatient)	Receive patient visiting clinic	Manual
			Empathy			Check patient appointment	User
			Responsiveness			Check if patient have medical insurance	User
			Assurance.			Receive payment	Manual
			Reliability			Guide patient to combined clinic	Manual
			Confidentiality	[1e]			
Handle a lab test	Administration of cancer treatment is improved	Lab test is performed	Empathy		Patient	Visit lab	Manual
			Responsiveness			Handle payment	Manual
			Assurance			Receive information to visit doctor	Send
			Reliability		Lab	Receive patient visiting lab	Manual
	Patient diagnose		Information domain	[2b]		Check if patient medically insured	User
			System user satisfaction	[2f]		Receive payment	Receive
			Information availability	[2c]		Perform test	Manual
			Confidentiality	[1e]		Inform patient to visit doctor	Send
						Add test results	User
Handle an imaging test	Administration of cancer treatment is improved	Imaging test is performed	Empathy		Patient	Visit imaging department	Manual
			Responsiveness			pay	Manual
			Assurance			Receive information to visit doctor	Send
	Patient diagnose		Reliability		Imaging department	Book appointment for patient	User
			Information domain	[2b]		Receive patient visiting imaging department	Manual
			System user satisfaction	[2f]		Check patient's appointment	User

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_ Business_ Process	Soft goal name	Q Ref		Task name	Task type
			Information availability	[2c]		Check if patient is medically insured	User
						Receive payment	Receive
			Confidentiality	[1e]		Perform test	Manual
						Add and report results	User
						Inform patient to visit doctor	Send
Handle a chemotherapy treatment	Improving administration of cancer treatment.	Patient chemotherapy treated	Empathy		Patient	Visit chemo	Manual
						Pay	Manual
			Responsiveness.			Receive treatment	Manual
	Patient treatment.				Chemotherapy department	Receive request for appointment booking	Receive
			Assurance			Receive patient visiting chemo	Manual
						Check if patient has appointment	User
			Reliability.			Ask patient to visit specialist	Manual
						Check if patient is medically insured	User
						Receive payment	Manual
	Adequate treatment plan is received					Perform treatment	Manual
						Add results	User
						Visit chemo	Manual
						Pay	Manual
						Receive treatment	Manual
						Receive request for appointment booking	Receive
Handle a patient radiotherapy treatment	Improving administration of cancer treatment.	Patient radiotherapy treated.	Empathy		Patient	Visit radio	Manual
						Pay	Manual
	Cancer treatment		Responsiveness.			Receive treatment	Manual
						Receive transfers	Receive
			Assurance		Radiotherapy department	Receive request for appointment booking	Receive
						Receive patient visiting radio	Manual
			Reliability.			Check if patient has appointment	User
	Adequate treatment plan is received					Ask patient to visit specialist	Manual
						Check if patient is medically insured	User
						Receive payment	Manual
						Begin treatment	Manual
						Check if patient need lab tests	Manual

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_ Business_ Process	Soft goal name	Q Ref		Task name	Task type
						Check if patient need imaging tests	Manual
						Transfer patient	User
						Add results	User
Handle a patient admission	Improving collection about cancer cases	Patient is admitted	Information domain of admission clerk	[3b]	Patient	Visit admission department	Manual
			Information availability	[3c]	Admission clerk	Receive request for patient admission	Receive
			System user satisfaction of admission clerk	[3f]		Check room availability	User
			Confidentiality	[1e]		Inform patient to visit department	Manual
			Empathy			Check if emergency case	Manual
			Responsiveness			Add patient to waiting list	User
			Assurance			Complete paper work	User
			Reliability				
Handle inpatient care	Improving administration of cancer treatment	Inpatient care.	Empathy		Inpatient care specialists and nurse	Receive patient visiting department and his papers	Manual
						Open admission file	User
						Add notes to file	User
	Patient treatment		Reliability			Check if patient needs tests	Manual
						Transfer patient to lab	Manual
						Check if patient needs imaging test	Manual
			Assurance			Transfer patient to imaging department	Manual
						Check if patient needs radiotherapy	Manual
						Transfer patient to radiotherapy department	Manual
			Responsiveness			Check if patient needs chemotherapy	Manual
						Transfer patient to chemotherapy department	Manual
						Check if patient needs surgery	Manual
						Begin surgery	Manual
						Check if patient need other treatment	Manual
						Continue treatment	Manual
						Update patient file	User

Table Q.1 (Cont'd): The 18 Case Processes Resultant from an Alignment of CCR Riva BPA with a BSV

Process (CP and/or CMP)	Goals		Soft goals		Pools	Tasks	
	Goal Contribution	Goal_Of_ Business_ Process	Soft goal name	Q Ref		Task name	Task type
Handle a patient medical records	Administration of cancer treatment is improved	Patient medical records are handled	Reliability		Receptionist	Find patient's appointment	Manual
						Request patient's file	Manual
			Information domain	[2b]		Receive patient file	Manual
						Check the file	Manual
			System user satisfaction	[2f]		Return patient file	Manual
					Medical records	Receive patient's file request	Manual
			Information availability	[2c]		Find patient's file	User
			Confidentiality	[1e]		Register file's details	User
						Send patient's file	Send
						Receive returned patient's file	Manual
						Check the file	Manual
						Check if there's a new patient	Manual
						Open file	User
						Save patient's file in library	User

Appendix R: Instantiating the GQ_BPAOnt for the CCR Representative Processes Using Protégé Editor

This appendix presents the instantiation of the CCR GQ-Riva BPA, CCR GQ-sBPMN and their merging using the Protégé ontology editor.

The GQ-BPAOnt instantiation begins with instantiating the GQ-srBPA ontology to the GQ-srBPA_CCR ontology. The instantiation of GQ-srBPA_CCR onotlogy depends on the presence of the GQOnt_CCR and on executing the alignment rules that requires intelligent human-based decisions, as was shown in Section 5.9. In this section, the business analyst is at charge of carrying out a manual extraction of the fundamental blocks EBEs and EBQs living within the GQOnt_CCR ontology instantiation. The business analyst should revisit all the instantiated individuals (i.e., goal-oriented components and not the relations) and set the alignment decisions. In addition, the business analyst needs to refine the extracted/driven EBEs from their linguistic perspective in order to address Ould's definition of EBEs. The Protégé editor is not intelligent enough to carry out this task, as this requires human intelligence.

The instantiation of the GQ-srBPA_CCR ontology begins by entering the refined EBEs and EBQs resulted from the alignment with GQOnt_CCR ontology as shown in Figure M.13. The business analyst has the option to classify the resulted EBEs and EBQs into new or detected ones using the alignment classification SWRL rules presented in Section 5.9.

The resulted 18 CPs in the generated portion of the GQ Riva BPA are designed and attached in the Appendices O and P. The 18 CPs are semantically presented in the GQ-sBPMN_CCR ontology that is an instantiation of the GQ-sBPMN ontology. In particular, the pools and the tasks in addition to goals and soft goals of the 18 CPs are created and attached in Appendix Q. The instantiation is carried out using the classes (Process, Pool, Tasks, (UserTask, ReceiveTask and ManualTask)). In addition, instantiating the Goals of the Process and the Soft Goals of the Process are needed. Figure R.1 shows a snapshot of instantiating an example of GQ-CP "Appointment booking". An instantiation example of GQ-sBPMN ontology is shown in Figure R.2.

It is apparent that the alignment is not fully automated, yet requires intelligent human-based decisions. However, the derivation of the CCR's GQ-srBPA ontology is automated after deciding the G-UoWs and their associated Q-UoWs using the original work implemented in the BPAOntoSOA framework, where the automation feature is facilitated using the SWRL rules presented in Appendix I.

Since the GQ-srBPA_CCR and GQ-sBPMN_CCR ontologies are present, then the original and new merging rules are applied to relate goals and quality requirements from former ontology to the latter. This is carried out after merging the two ontologies into one repository, namely GQ_BPAOnt_CCR ontology. Figure R.3 depicts a window of this repository.

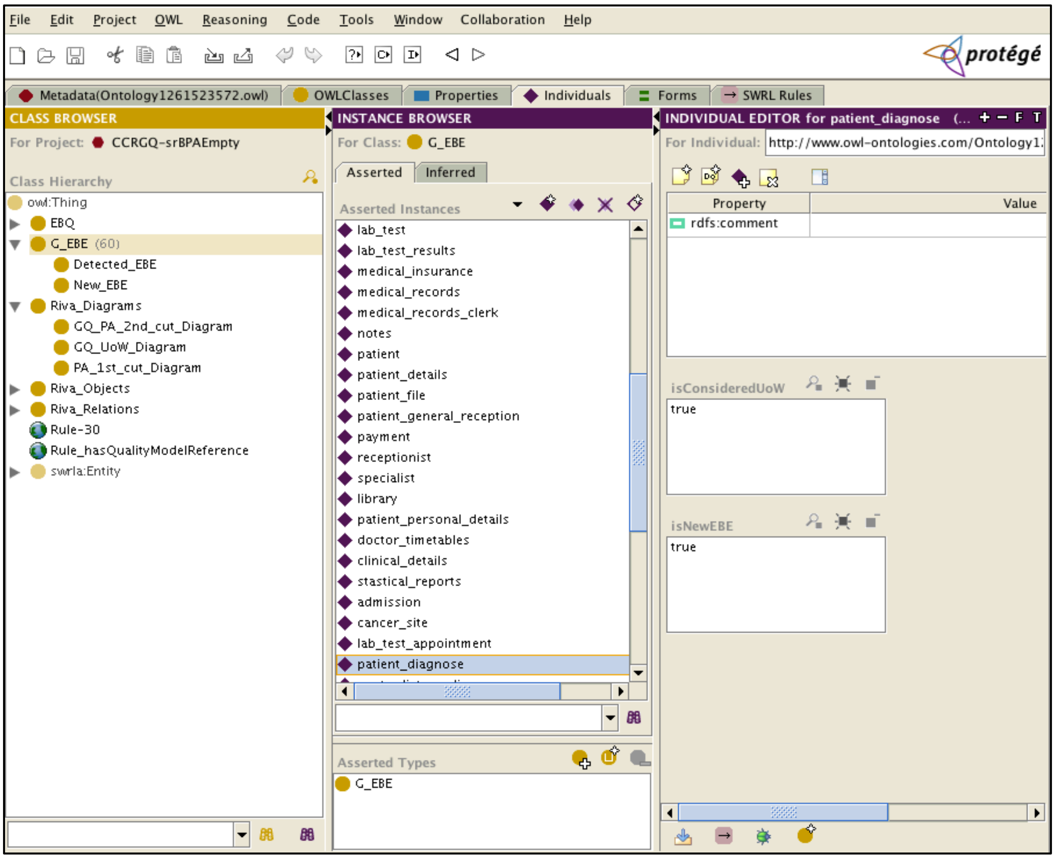


Figure R.1: Window Example Captured from the GQ-srBPA_CCR Ontology.

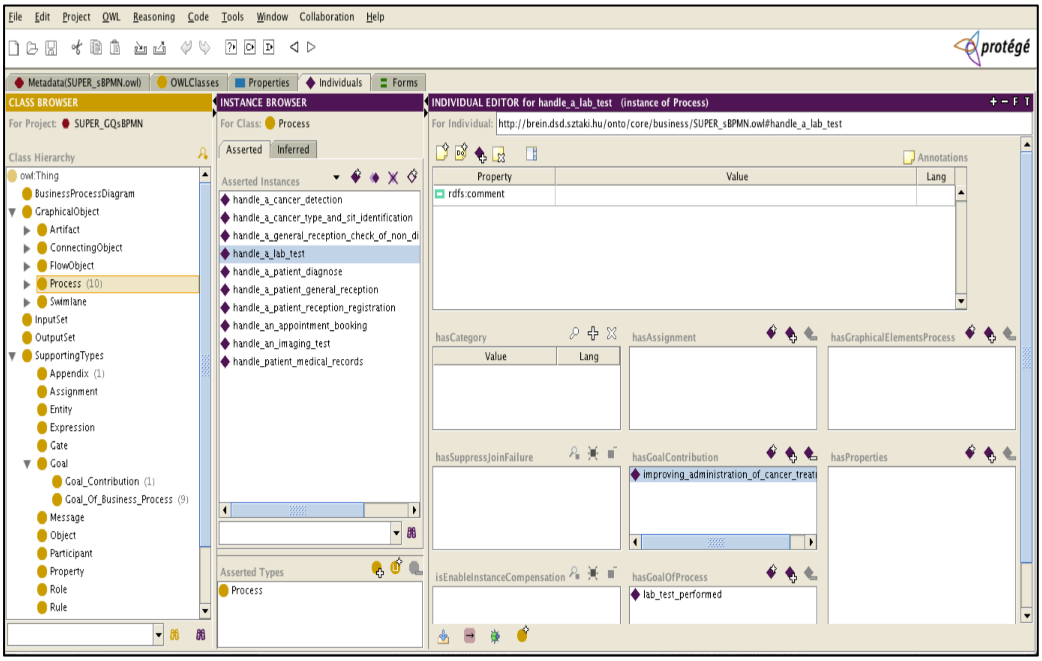


Figure R.2: An Instantiation Example of the SUPER-sBPMN_CCR Ontology.

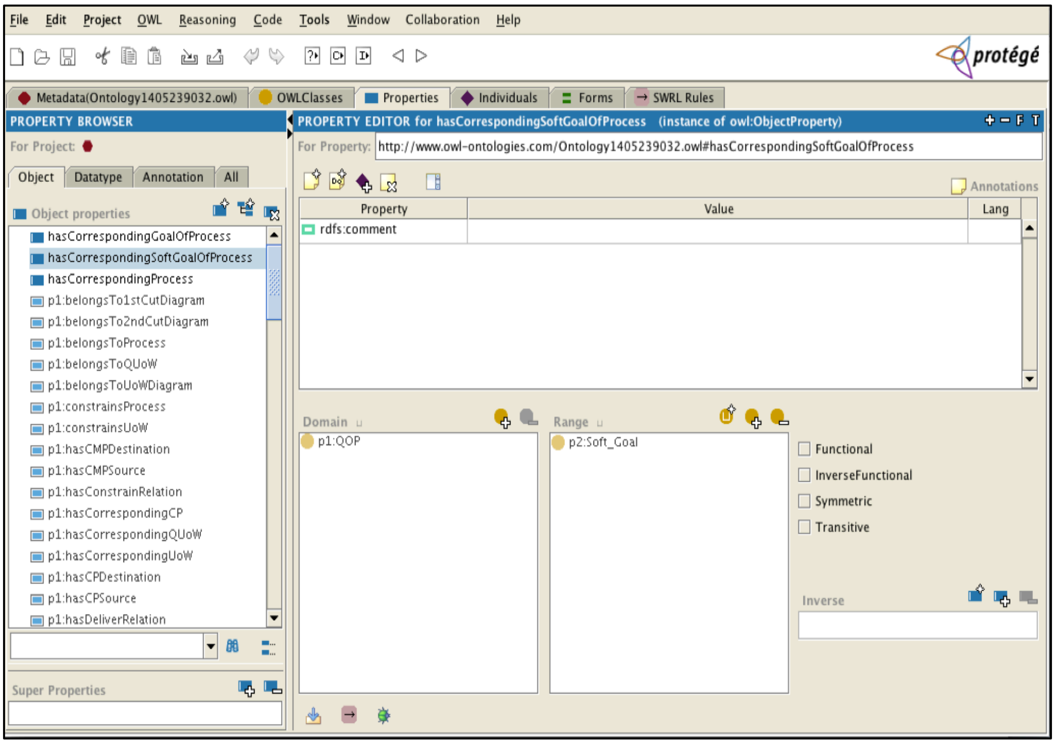


Figure R.3: Merging the Instantiation of GQ-srBPA GQ-sBPMN Ontologies into GQ-BPAOnt_CCR

Appendix S: Applying the GQ Software Service Identification Layer Using the CCR Representative Processes

This appendix shows an application of the third layer in the GQ-BPAOntoSOA framework using the selected processes as case study as presented in Section 7.3.1. The GQ-BPAOnt_CCR is a primary input to GQ-software service identification layer, where the GQOnt_CCR is a secondary input (i.e., needed to enter the QoS identifier component). The output is the identified candidate software services, their associated capabilities and QoS requirements. The GQ-Riva BPA for the selected case study is recalled (Figure N.22), where the conditional and normal relationships are identified as suggested in (Yousef, 2010) in order to derive the clusters. 19 request and deliver relationships were set as conditional, where only one remained as normal relationship after analysing the 17 CPs involved in the 20 request and deliver relationships shown in Appendix N (Figure N.22). Figure S.1 shows the GQ-Riva BPA for the selected case study after deleting the conditional relationships. The work of Sections 6.3 and 6.4 is applied and has generated 16 standalone clusters and one related cluster (i.e., have two members), as shown in Figure S.2. The semantic representation of these clusters is carried out using the Protégé ontology editor, as shown in Figure S.3.

Each cluster in Figure S.2 is treated as a candidate software service (i.e., its members are BPMs in the GQ-Riva BPA), where its QoS requirements are derived from the GQ-BPAOnt and GQOnt ontologies, as presented in Section 6.5. The capabilities of the cluster are extracted from the GQ-BPAOnt_CCR and particularly from the classes UserTask and SendTask, (task instances of the classes ReceiveTask and Manual are excluded) in the corresponding instantiation of the member using the GQ-sBPMN ontology. It is necessary to recall that capabilities are classified into goal-based and quality-driven. The former is the original as in (Yousef, 2010), where the quality-driven capabilities are derived from the dynamic operationalisations in the relevant NFR framework models, as presented in Section 6.6. Table S.1 shows each cluster, its associated members, goals, quality of cluster and capabilities. Each entry in table is shaded with respect to the colours in Figure S.2. Pink entry means novel cluster, where yellow refers to as-is cluster with some needed modifications required because of the integration of goals and quality requirements. The white entries for the pre-existing clusters with very light refinements.

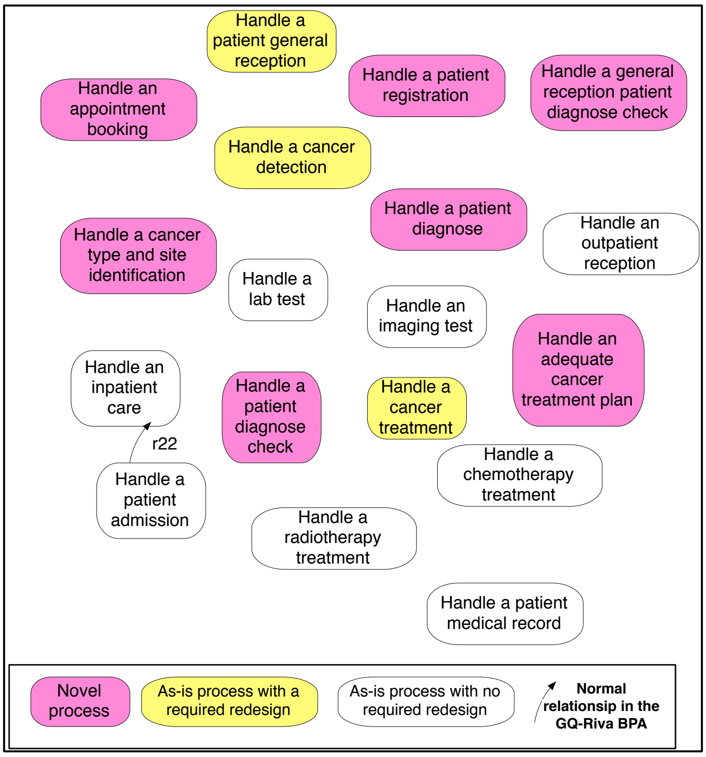


Figure S.1: The GQ-Riva BPA After Deleting Conditional Relationships For the Selected Processes in the CCR Case Study

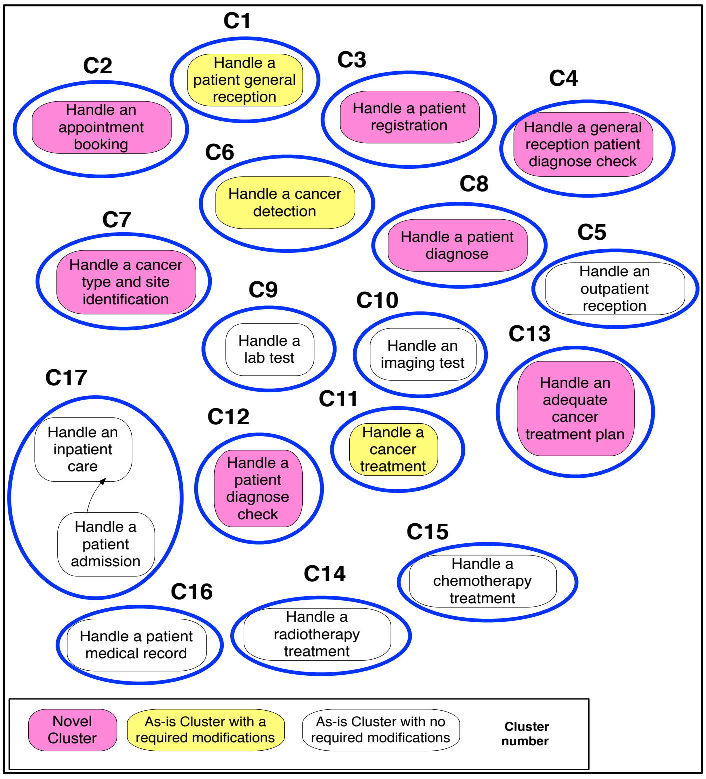


Figure S.2: The GQ-RPA Clustrs for the Selected Processes in the CCR Case Study.

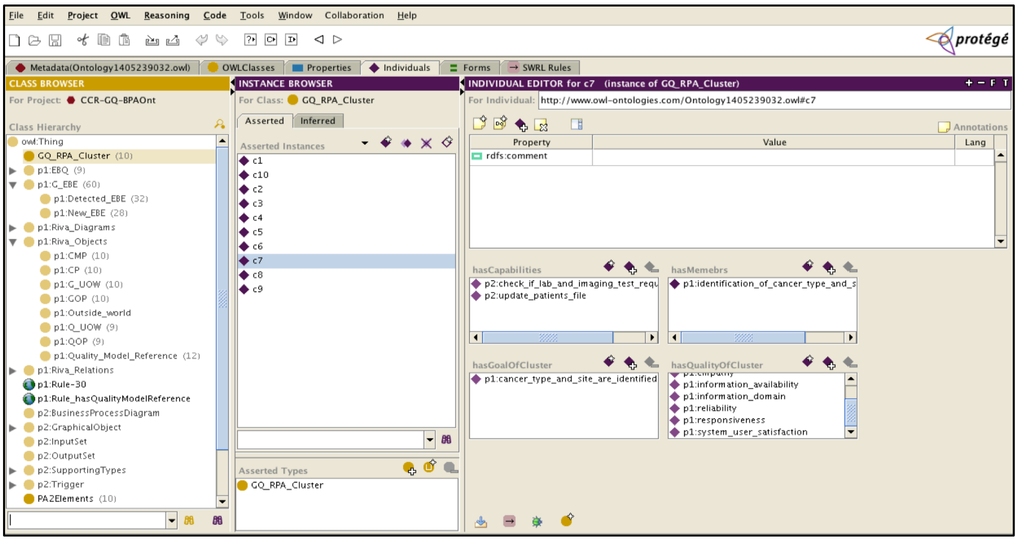


Figure S.3: GQ-RPA Clusters for the CCR Case Study as Depicted from the Protégé GQ-BPAOnt_CCR Window

Table S.1: Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C1: General Reception	Patient Reception	Handle a patient general reception	Patient Reception.	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Information availability [1c]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.	Goal-driven Capabilities (reduced from 8 to 5): 1- Check if patient has an appointment. 2- Check if emergency. 3-Transfer patient to emergency. 4-Check if patient is diagnosed. 5- Request appointment Quality-driven Capabilities: 1) ID: i) Create manual files ii) Generate statistical reports 2) System User Satisfaction (SUS): i) support enquiries. ii) help in producing statistical reports. 3) Confidentiality: i) Number of sent and received files must be recorded. ii) Names of persons

					who send and receive files must be recorded iii) Link electronic to manual files iv) Unauthorized access must not be allowed v) Transferred data must be encrypted.
C2: Appointment Booking	Appointment is booked	Handle an appointment booking	Appointment booked.	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Fast/reliable booking. [1a] 4- Information availability. [1c] 5-Confidentiality [1e]. 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability.	Goal-driven Capabilities: 1-Book an appointment by phone. Quality-driven Capabilities 1- ID of receptionist: i)book an appointment. 2- SUS: i) Support enquiries using different search keys. ii) help in producing statistical reports. 3- Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C3: Patient Registration	Patient is registered	Handle a patient registration	Patient is registered.	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Fast/reliable booking. 4- Information availability. [1c] 5-Confidentiality [1e]. 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability.	Goal-driven Capabilities: 1-Check if patient is in DB. 2- Inform patient to visit specialist. 3-Request patient's details. 4- Register patient's details. 5- Check if patient has an appointment. 6-Make an appointment. 7-Provide information related to patients Quality-driven Capabilities 1- ID of receptionist: i)book an appointment. ii) register patient details. 2- SUS: (Same capabilities in General Reception Cluster) 3- Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C4: General Reception Patient Diagnose Check	General reception patient diagnose is checked	Handle a general reception check of patient diagnose	General reception patient diagnose is checked	1- ID of receptionist. [1b] 2- System user satisfaction of receptionist. [1d] 3- Information availability. [1c] 4-Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.	Goal-driven Capabilities: 1-Check if patient is diagnosed. 2- Inform patient to visit cancer detection unit. Quality-driven Capabilities 1- SUS: (Same capabilities in General Reception Cluster) 2- Confidentiality (Same capabilities in General Reception Cluster)
C5: Outpatient Reception	Outpatient clinic reception	Handle an outpatient clinic reception	Outpatient clinic reception	1- ID of receptionist. [2b] 2- System user satisfaction of receptionist. [2g] 3- Information availability. [2c] 4-Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.	Goal-driven Capabilities: 1-Check patient appointment 2- Check if patient have medical insurance. Quality-driven Capabilities: 1-ID of receptionist: i)- Arrange appointments ii)- Generate statistical reports iii)- register patient personal details. 2-SUS: i) Support enquiries using different search keys. ii) Help in producing statistical reports iii) Register patient personal details. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C6: Cancer detection reception	Cancer is detected	Handle a cancer detection	Cancer is detected	1- System user satisfaction of doctor [2e]. 2- System user satisfaction of receptionist [2g]. 3- Information availability [2c]. 4- Reliable/fast appointment request and payment method. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability. 9- Information domain [2b] 10-Confidentiality [1e].	Goal-driven Capabilities (reduced from 9 to 3): 1- Check if patient is medically insured. 2- Check if patient in DB. 3-Register patient's details. Quality-driven Capabilities: 1-ID of receptionist: (Same capabilities in Outpatient Reception Cluster) 2- System user satisfaction for receptionist [2g]: i) Support enquiries using different search keys. ii) Help in producing statistical reports iii) Register patient personal details. 3- System user satisfaction for doctor [2e]: i) Support enquiries using different search keys. ii) Do not accept contradictory diagnoses. iii) Add the details and results of diagnostic process. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C7: Cancer Type and Site Identification	Cancer type and site are identification	Handle a cancer type and site identification	Cancer type and site are determined.	1- Information domain of doctor [2b]. 2- System user satisfaction of doctor [2e]. 3- Confidentiality [1e]. 4- Information availability [2d]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability.	Goal-driven Capabilities: 1- Check if lab and imaging test require review. 2-Referral to special combined clinic. 3- Update patient file. Quality-driven Capabilities: 1-ID of doctor: i) Provide diagnoses details. ii) Name combined clinic. 2- SUS of doctor: (Same capabilities in cancer detection reception cluster). 3-Confidentiality (Same capabilities in General Reception Cluster)
C8: Patient Diagnose	Patient is diagnosed	Handle a patient diagnose	Patient is diagnosed for cancer detection.	1- Reliable/Fast appointment request 2- Information domain of doctor [2b]. 3- System user satisfaction of doctor [2e]. 4- Confidentiality [2g]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability. 9- Information availability [2d].	Goal-driven Capabilities: 1-Book an imaging test appointment for patient. 2- Order tests. Quality-driven Capabilities: 1- SUS of doctor: (Same capabilities in cancer detection reception cluster). 2-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C8: Lab test	Lab test is performed	Handle a lab test	A lab test is performed	1-Information domain of lab [2b]. 2-Information availability [2c]. 3-System user satisfaction of lab [2f]. 4-Confidentiality [1e]. 5-Empathy. 6-Responsiveness. 7-Assurance. 8- Reliability.	Goal-driven Capabilities (remained capabilities): 1- Check if patient is medically insured. 2- Inform patient to visit doctor. 3- Add test results. Quality-driven Capabilities: 1- IR of lab: i) Provide results of lab. ii) Generate statistical reports. 2- SUS: i) Support enquiries using different search keys. ii) Arrange appointments. iii) Add the results of investigation. iv) Generate statistical reports. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C9: Imaging test	Imagining test is performed	Handle an imaging test	An imaging test is performed	6- Information domain of imaging department [2b]. 7- Information availability [2c]. 3-System user satisfaction of imaging department [2f]. 8- Confidentiality [1e]. 9- Empathy. 10- Responsiveness. 11- Assurance. 8- Reliability.	Goal-driven Capabilities (remained capabilities): 1- Check if patient is medically insured. 2- Inform patient to visit doctor. 3- Add and report test results. 5- Check patient's appointment. 6- Book appointment for patient. Quality-driven Capabilities: 1- IR of imaging dept: i) Provide results of imaging dept. ii) Generate statistical reports. iii) Arrange appointments. 2- SUS: i) Support enquiries using different search keys. ii) Arrange appointments. iii) Add the results of investigation. iv) Generate statistical reports. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C11: Cancer Treatment	Patient reception for cancer treatment.	Handle a patient treatment	Patient reception for cancer treatment.	2- System user satisfaction of specialists [3e]. 3- Information domain of receptionist [2b]. 4- Empathy. 5- Responsiveness. 6- Assurance. 7- Reliability. 8- Information availability [3d]. 8- Confidentiality [1e].	Goal-driven Capabilities (reduced from 9 to 1): 1-Check if patient needs diagnose Quality-driven Capabilities: 1- SUS for specialists: i)Support enquiries using different search keys. ii) Do not accept contradictory diagnoses. iii) Add the results of treatment. 2- ID of receptionist: (Same capabilities in Outpatient Reception Cluster) 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C12: Patient Diagnose Check	Patient diagnose is checked	Handle a patient diagnose check	Patient diagnose is checked	1- Reliable/fast appointment requesting method. 2- Information domain . [3b] 3- Information availability [3d]. 4- System user satisfaction. [3e]. 5- Confidentiality. [1e] 6- Empathy. 7- Responsiveness. 8- Assurance. 9- Reliability	Goal-driven Capabilities: 1- Review patient's history and investigations. 2- Request admission 3- Order tests Quality-driven Capabilities: 1- SUS for specialists (Same capabilities in Patient Treatment Cluster) 2-ID of specialists: i) Provide treatment details. ii) Provide admission duration. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C13: Adequate Treatment Plan	Adequate treatment is received.	Handle an adequate treatment plan	Adequate treatment is received.	1- Reliability. 2- Information domain. [3b] 3- Information availability [3d]. 4- System user satisfaction. [3e]. 5- Confidentiality. [1e] 6- Empathy. 7- Responsiveness. 8- Assurance	Goal-driven Capabilities: 1-Device plan for treatment 2 Inform patient to visit radiotherapy department 3- Book appointment for radiotherapy treatment 4- Inform patient to visit chemotherapy department 5- Book appointment for chemotherapy treatment Quality-driven Capabilities: 1- SUS for specialists (Same capabilities in Patient Treatment Cluster) 2-ID of specialists: i) Provide treatment details. ii) Provide admission duration. 3-Confidentiality (Same capabilities in General Reception Cluster)
C14: Radiotherapy Treatment	Patient radiotherapy treated.	Handle a radiotherapy treatment	Patient radiotherapy treated.	1- Empathy. 2- Responsiveness. 3- Assurance. 4- Reliability.	Goal-driven Capabilities (remained capabilities): 1- Check if patient has appointment 2- Check if patient is medically insured 3- Add results 4- Transfer patient.

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C15: Chemotherapy Treatment	Patient chemotherapy treated.	Handle a chemotherapy treatment	Patient chemotherapy treated.	1- Empathy. 2-Responsiveness. 3-Assurance. 4-Reliability.	Goal-driven Capabilities (remained capabilities): 1-Check if patient has appointment 2-Check if patient is medically insured 3- Add results
C16: Patient Medical Record	Patient medical record is handled	Handle a patient medical record	Medical record of patient is handled.	1-ID of receptionist and medical records. [1b] 2-System user satisfaction of receptionist and medical records. [1d] 3-Information availability. [1c] 4-Confidentiality [1e]. 5- Reliability.	Goal-driven Capabilities: 1-Find patient's file. 2-Register file's details. 3- Send patient's file. 4- Open file. 5-Save patient's file in library. Quality-driven Capabilities 1- ID of receptionist and medical records: i) Create manual files ii) Create e-file. iii) Generate statistical reports. 2- SUS: i) Support enquiries using different search keys. ii) Help in the production of statistical reports. iii) Personal patient's details registration. 3-Confidentiality (Same capabilities in General Reception Cluster)

Table S.1 (Cont'd): Identified GQ Candidate Sotware Services for the Selected Processes in the CCR Case Study

GQ-Candidate Software Service	Goal of Cluster (GoS)	Software Service Members (GQ-CPs and GQ-CMPs within GQ-RPA cluster)	Goal of Service Member (GoS)	QoS Member	Capabilities including quality-driven Capabilities
C17: Patient admission and Care	Patient admission and inpatient care is received.	Handle a patient admission	Patient is admitted.	1- Information domain of admission clerk [3b]. 2- Information availability [3c]. 3-System user satisfaction of admission clerk [3f]. 4- Confidentiality [1e]. 5- Empathy. 6- Responsiveness. 7- Assurance. 8- Reliability	Goal-driven Capabilities: 1- Check room availability 2- Add patient to waiting list 3- Complete paper work Quality-driven Capabilities 1- ID of admission clerk: i) Arrange admission appointments ii) Generate statistical reports 2- SUS for admission clerk: i) Organize appointments. ii) Make queries using different search elements. iii) Make statistical reports. 3-Confidentiality (Same capabilities in General Reception Cluster)
		Handle an inpatient care	Inpatient care is received	1- Empathy. 2- Responsiveness. 3- Assurance. 4- Reliability.	Goal-driven Capabilities: 1- Open admission file. 2- Add notes to file. 3- Update patient file.

Appendix T: Informing the 3Cs for the GQOnt Ontology Using the CCR Representative Processes

The work of this appendix is employed in the first level of the research evaluation framework presented in Section 7.3 in order to pave the way answering the research questions at the end of Chapter 7. This appendix presents three tables, where each table aims at informing the 3Cs of each component in the GQOnt ontology instantiation layer using the CCR case study. Yes/No informs the 3Cs for each entry in the tables. Therefore, Table T.1 informs the 3Cs for the siGoal_CCR ontology, where Table T.2 informs the 3Cs of the sQuality_CCR ontology. Finally, Table T.3 presents informing the 3Cs for the linker component.

Table T.1: The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	siGoal_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Organisation	One agreed boundary was determined, namely KHCC	One instance was created with its associated “aim to” relationships and HBGs.	✔	✔	✔
HBG	One HBG was identified for the agreed organisation.	One instance was created, namely improving CCR business process.	✔	✔	✔
Aims to relationship in the BS model	One relation was identified from the agreed organisation to the identified HBG.	One instance was created of the Class Aims_To with sources and destinations properties were presented.	✔	✔	✔
BS model	One model was created in order to represent first GO model.	One instance was created.	✔	✔	✔

Table T.1 (Cont'd): The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	siGoal_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
HSD actors	Two actors were designed in the HSD model, where each had their intentions and abilities that are designed in the form of IH-Gs and IH-SGs.	Two instances from this class with their IH-G and IH-SG intentions and abilities were created.	✔	✔	✔
IH-G dependency relationships	One IH-G dependency relationship was designed from the patient actor to the treatment team actor.	One instance and its associated source and destination properties were created.	✔	✔	✔
IH-G dependum	One IH-G dependum was designed between the two actors	One instance was created of the class IH_G_Dependum.	✔	✔	✔
IH-SG dependency relationships	8 IH-G dependency relationships were designed from the patient actor to the treatment team actor.	8 instances and their associated source and destination properties were created.	✔	✔	✔
IH-SG dependum	8 IH-G dependums were designed between the two HSD actors that constrain the associated IH-G.	8 instances were created of the class IH_SG_Dependum with their associated property (*constrainsIHG*).	✔	✔	✔

Table T.1 (Cont'd): The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	CCR_siGoal Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
HSD model	One HSD model was designed.	One instance was created of this class.	✔	✔	✔
Goal of SD	Three goals were designed as sub goals of the IH-G "administration of cancer treatment is improved".	Three instances were created of the class SD_Goal.	✔	✔	✔
SD model	Three SD models, that correspond the three Goals of SD, are designed.	Three instances were created of the class SD_Model.	✔	✔	✔
Actors	13 actors were designed in the i* framework models for the selected case study.	13 corresponding instances were created along with goal abilities, goal intentions, soft goal abilities, soft goal intentions, addressing goals and soft goals properties.	✔	✔	✔
Goal dependums	14 goal dependums were designed in the goal dependency relationships in the i* framework models.	14 instances were created of the class Goal_Dependum that are derived automatically from the class Goal using the SWRL rule: Creating_SD_Goal_Dependency.	✔	✔	✔
Soft goal dependums	3 soft goal dependums were designed in the three i* framework models.	3 instances were created of the class Soft_Goal_Dependum that are derived automatically from the class Soft_Goal using the SWRL rule: Creating_SD_SoftGoal_Dependency.	✔	✔	✔

Table T.1 (Cont'd): The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	CCR_siGoal Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Goal dependency relationships	26 goal dependency relationships were designed employing the 14 goal dependums between the identified actors.	26 instances were created of the class Goal_Dependency along with their source and destination properties.	✔	✔	✔
Soft goal dependency relationships	4 soft goal dependency relationships were designed employing the above three soft goal dependeums.	4 instances of the class Soft_Goal_Dependency were created along with their source and destination properties.	✔	✔	✔
Goal	19 goals were designed in the entire i* framework including the third and the fourth levels in the goal network.	19 goals were manually created of the class Goal.	✔	✔	✔
Soft goal	11 soft goals were designed in the entire GO view	11 soft goals were manually instantiated from the class Soft_Goal	✔	✔	✔
Task	107 tasks were designed in the i* framework models.	107 tasks were manually created in siGoal ontology sing the Class Task	✔	✔	✔

Table T.1 (Cont'd): The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	CCR_siGoal Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Resource	1 resource (payment) was designed in the i* framework.	One instance was manually created of the class Resource.	✔	✔	✔
GT mean end relationship	13 relation was designed between goals (ends) and their related tasks (means)	13 instances were created in Protégé of the Class GT_MeanEnd. The source and the destination were automatically created using the SWRL rule: Rule-Creating_GT_Relation	✔	✔	✔
Task decomposition into sub task relationship	133 relationships were designed in the SR models.	133 instances were created from the class Task_Decomposition_subTask along with their sources and destinations using the Rule-Creating_Task_Decomposition_subTask_Relation	✔	✔	✔
Task decomposition into sub goal relationship	37 relationships were designed in the SR models.	37 corresponding instances were created from the class Task_Decomposition_subGoal along with their sources and destinations using SWRL rule: Rule-Creating_Task_Decomposition_subGoal_Relation	✔	✔	✔
Task decomposition into sub resource relationship	1 relationship is designed in the SR models.	One instance was created along with its source and destinations using Rule-Creating_Task_Decomposition_ResourceFor_Relation	✔	✔	✔

Table T.1 (Cont'd): The 3Cs Values For the siGoal_CCR Ontology

siGoal Ontology Elements	GO View for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	CCR_siGoal Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Task decomposition into sub soft goal relationship	47 relationships were designed in the SR models.	47 corresponding instances from the class Task_Decomposition_SoftGoalFor using: Rule-Creating_Task_Decomposition_subSoftGoal_Relation	✔	✔	✔
First level in Goal network	2 relationships are designed in this level.	Rule-Creating_First_Level_Goal_Network	✔	✔	✔
Second level in goal network	3 relationships created the second level in the goal network.	Three instances were created using: Rule-Creating_Second_Level_Goal_Network	✔	✔	✔
Third level in goal network	17 relationships created the third level in the goal network.	17 instances were created using the SWRL rule: Rule-Creating_Third_Level_Goal_Network	✔	✔	✔
Fourth level in goal network	12 relationships created the fourth level in the goal network.	12 instances were created using: Rule-Creating_Fourth_Level_Goal_Network	✔	✔	✔

Table T.2: The 3Cs Values For the sQuality_CCR Ontology

sQuality ontology elements	Quality view for the representative case study (using the work of (Aburub, 2006))	sQuality_CCR ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
SIG diagram	15 SIG diagrams designed in (Aburub, 2006)	15 corresponding instances were created from the Class: SIG_Diagram.	✔	✔	✔
NFR type soft goal	4 NFR type soft goals were identified in the 15 SIG diagrams in (Aburub, 2006)	4 instances were created from the class: NFR_Type_SoftGoal.	✔	✔	✔
NFR soft goal	25 NFR soft goals were designed in the relevant NFR framework models.	25 instances were manually created from the Class NFR_SoftGoal.	✔	✔	✔
Operationalisation soft goal	41 operationalisation	41 instances of the the Class Operationalisation_SoftGoal	✔	✔	✔
Static operationalisation	15 operationalisation soft goals were determined as static.	15 instance were automatically generated using SWRL rule: Determining_Static_Operationalisation	✔	✔	✔
Dynamic operationalisation	26 operationalisation soft goals were determined as dynamic.	26 instance were automatically generated using SWRL rule: Determining_Dynamic_Operationalisation	✔	✔	✔
NFR type soft goal decomposition relationship	32 were designed to decompose a main NFR into sub soft goals.	32 instances were created along with their associated properties that determine parent and offspring soft goals.	✔	✔	✔
Explicit interdependency relationships	85 were designed between the operationalisation soft goals	85 instances of Class Explicit_Interdependency were created along with their associated properties that determine mean and end soft goals.	✔	✔	✔

Table T.3: The 3Cs Values For the Linker

Linking cases	BSV view for the representative case study	CCR_GQ ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Linking case 1 (When a soft goal dependum is addressed using a NFR framework)	No linking need using this case	No linking is instantiated for this case	✔	✔	✔
Linking case 2 (When a soft goal is sub from a task in order to work as a constraint)	23 linking performed in the entire BSV for the CCR process using this case.	23 linking cases were semantically represented in the CCR_GQOnt.	✔	✔	✔
Linking case 3 (When a dynamic operationalization is matched with a task)	8 linking situations needed in the CR BSV.	8 linking instantiations were carried out in the CCr_GQOnt for this case.	✔	✔	✔
Linking case 4 (When a staticoperationalization is matched with a resource)	No linking need using this case	No linking is instantiated for this case	✔	✔	✔

Appendix U: Informing the 3Cs of the GQ-BPAOnt Ontology Using the CCR Representative Processes

This appendix presents the evaluation work needed in Chapter 7 (Section 7.3.4) that concerns with informing the 3Cs for the semantic representation of the GQ-Riva BPA and the proposed merging rules with associated GQ-BPMNs. This work is shown in the two Tables U.1 and U.2.

Table U.1: Informing the 3Cs for the CCR Representative Processes GQ-Riva BPA and their CCR_GQ_srBPA Ontology

GQ-srBPA Ontology Elements	GQ-Riva BPA for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	GQ_srBPA_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
EBEs	77 EBEs were identified.	77 instances were created of the Class G_EBE.	✔	✔	✔
EBQs	9 EBQs were identified.	9 instances were created of the Class EBQ.	✔	✔	✔
UoWs	18 UoWs were filtered from the 77 EBEs and designed in the UoW diagram.	18 instances were classified from the 77 EBEs using the SWRL rule: Rule_UOW_Instnaces.	✔	✔	✔
Generate	19 generate relationships were designed in the GQ-UoW diagram.	19 instances were created of the Class Generate along with their associated sources and destinations.	✔	✔	✔
Q-UoWs	The 9 EBQs are Q-UoWs for the 18 UoWs.	9 corresponding Q-UoWs were created using the SWRL Rule: Rule_Q_UoW_Instances	✔	✔	✔

Table U.1 (Cont'd): Informing the 3Cs for the CCR Representative Processes GQ-Riva BPA and their CCR_GQ_srBPA Ontology

GQ-srBPA Ontology Elements	GQ-Riva BPA for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	GQ_srBPA_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Quality model reference	15 NFR framework models associated the elaborative Q-UoWs	15 corresponding instances were created of the Class Quality_Model_Reference	✔	✔	✔
Constrain	135 relationships were created from Q-UoWs to G-UoWs.	135 corresponding instances were created of the Class Constrain along with their associated sources and destinations using the SWRL rule: Rule_hasConstrainRelation	✔	✔	✔
Request	19 were designed in the 1st and the 2nd cut architecture.	19 instances were created of the Class Request using the original SWRL Rule: Rule_1st_cut_translated_relations	✔	✔	✔
Start	19 relationships between the CPs in the 1st cut architecture.	19 instances were created of the Class Start using the original SWRL Rule: Rule_1st_cut_translated_relations	✔	✔	✔
Deliver	19 deliver relationships were designed in the 1st and the 2nd cut architecture.	19 instances were created of the Class Deliver using the original SWRL Rule: Rule_1st_cut_translated_relations	✔	✔	✔
CP	18 CPs are generated from the corresponding 18 UoWs in the 1st and 2nd cut architecture.	18 instances of the Class CP.	✔	✔	✔

Table U.1 (Cont'd): Informing the 3Cs for the CCR Representative Processes GQ-Riva BPA and their CCR_GQ_srBPA Ontology

GQ-srBPA Ontology Elements	GQ-Riva BPA for the Representative Case Study (using Omni Graffle Professional Diagramming Tool)	GQ_srBPA_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
CMP	18 CMPs in the 1st cut architecture.	18 instances of the Class CP	✔	✔	✔
GoP	18 GoP were created for the 18 processes in the 2nd cut architecture.	18 instances were created using the SWRL rule: Rule_GoP_CP_or_CMP_in_PA2_Diagram	✔	✔	✔
QoP	9 QoP were identified to constrain the processes in the 2nd cut architecture.	9 instances were created of the Class QoP along with their associate quality reference model.	✔	✔	✔
GQ 2nd cut architecture	The Riva BPA constitutes 18 CP along with their associated GoP and QoP.	18 CMPs were deactivated where the 18 CPs are remained along with their associated properties hasGoP and hasQoP.	✔	✔	✔

Table U.2: Informing the 3Cs for the Merging Rules and their Semantic Representation for the Selected Case Study

GQ_BPAOnt Ontology Merging	Merging Rules	GQ_BPAOnt_CCR Ontology (Protégé ontology editor)	Remarks		
			Correctness	Completeness	Consistency
Merging CCR_GQ-srBPAOnt and GQ-CCR_sBPMN into CCR_GQ_BPA Ont	Each of the 18 CPs in the GQ Riva BPA encapsulates a corresponding BPMN model as shown in Appendices O and P. Also, each GoP and QoP is related to a goal and soft goal in BPMN	Each of the 18 CPs in the GQ-srBPAOnt is related to a corresponding instance of the Class Process. Also, the GoP and QoP in each CP are related to a corresponding instance of the Class Goal and Soft Goal, respectively.	✔	✔	✔

Appendix V: Glossary

3Cs: Completeness, Correctness and Consistency.

Actor: is defined as an active entity - a human (e.g., doctor) or non-human, (e.g., flight reservation system)- that conducts a set of actions to fulfil a goal.

BE: Business Entities (BE) is the set of the sets (i.e., Riva original concepts) that were presented in the original Riva method

BIA: business IT alignment is the amount to which the IT applications, infrastructure and organization, the business strategy and processes enables and shapes, as well as the process to realize this

BPAOnt Ontology Instantiation Layer: is the first layer in the BPAOntoSOA framework that is established in order to instantiate the semantic representation of a Riva BPA and its associated BPMs using OWL-DL, namely the BPAOnt

BPAOntoSOA framework: is a semantic framework that derives candidate software services and capabilities from a Riva BPA and its associated BPMs for an organisation

BQ: the Business Qualities (BQs) or the Quality of the Business is the second main sub set in BU that mirrors its sibling, which is the BE, but from the quality point of view.

BS Model: This model appears as the first model within the first-level of the refined *i** framework modelling stages.

BSV: captures the strategic goals that drive an organisation forward. The goals may be decomposed into various tactical approaches for achieving these goals and for providing traceability through the organisation. These strategic goals are mapped to metrics that provide on-going evaluation of how successfully the organisation is achieving its goals.

BU: the Business Universe (BU) represents the set of elements that are classified and appropriately related based on the Riva method guidelines/rules in order to manage the systematic derivation of the GQ-Riva BPA (i.e., the 2nd cut architecture)

CG: that is collective goals (i.e., union of goals) among the GQ-CPs and/or GQ-CMPs in the entire cluster.

CP: case process that represents instances from the original process.

CMP: case management process manages the function of the CP instances.

DBE: embodies the way the stakeholders chose to do the business, and it is not a subject of the matter.

DBQ: Designed Business Quality (DBQ) represents the set of quality attributes/characteristics or sub characteristics that manifest how a given organisation choose to achieve an EBQ.

EBE: is identified as the entity that substantially characterises the business that the organisation is in and forms a subject of matter of the business.

EBQ: The Essential Business Quality (EBQs) requirements are identified and they simply refer to the main quality requirements, NFRs or soft goals (e.g., security and information availability) within the agreed organisation boundary.

GO Approach: employ the concept of goal in order to articulate the strategic view of an enterprise. The GO approach is one of the methods used to arrive at a common understanding of a strategic view.

It is defined as an analytical method used in RE that "encourages the modelling of goals in order to understand or describe problems associated with business structures and processes and their supporting systems" .

Goal and Quality based Software Service Identification Layer: This layer derives from the software service identification layer in the original BPAOntoSOA framework. It carries out the generation of the semantic identification of software services from the input GQ-BPAOnt ontology instantiation for an organisation

Goal of SD: The GBP generated from the decomposition in the BSV.

GORE: goal-oriented modelling in requirements engineering research area.

GoS: Goal of a Service.

GQ-BPAOnt Ontology Instantiation Layer: This layer stems from the original BPAOntoSOA framework, and extended using the above GQOnt ontology instantiation layer

GQ-C: the goal-based and quality-linked RPA clusters.

GQOnt: consists of two sub ontologies, siGoal ontology represents relevant GO models following the i* framework, and the sQuality ontology represents the NFR framework.

GQOnt Ontology Instantiation Layer: The layer is required to semantically represent a business strategy view for a business organisation.

GQR-BPMs: recent Goal-based, Quality-linked and Role-driven Business Process Models

GQ-Riva BPA: a Goal-based and Quality-linked Riva BPA.

GQ-RPA Cluster: RPA clusters are informed by the associated goals and quality requirements.

GQ-sBPMN Ontology Instantiator: This component is reused and lightly refined from sBPMN ontology instantiator in the BPAOntoSOA framework.

Goal-based and Quality-linked Service Identifier: This component inputs the GQ-BPAOnt ontology instantiation for an organisation and outputs the corresponding goal-based and quality-linked -RPA clusters

GQ Service Capabilities Identifier: The purpose of this component is the same as in the original BPAOntoSOA framework, but taking into account the integration of goals and quality requirements.

GQ-SI: Goal-based and Quality-linked Service Identification approach.

GQ-srBPA Ontology: Goal-based and Quality-linked srBPA is a refined ontology that encapsulates the srBPA ontology.

GQ-srBPA-sBPMN Ontology Merger: This component carries out the integration of the two ontologies, which are the GQ-srBPA and GQ-sBPMN.
HBG: is the ultimate goal of the agreed organisation.

HSD model: this is the second model within the first-level in the refined *i** framework design that results the first goal-oriented dependency model.

***i** Framework:** is a GO approach that was categorised within the approaches proposed for understanding the current situation of an enterprise in relation to its goals as part of the RE elicitation activity.

IEEE-380: is a standard as another example of SORE.

IH-G: is defined as the set of the immediate decomposed goals that make up the HBG parent. Also, the IH-G is defined as the main objective for a number of collaborating GBPs that aim to meet the IH-G parent

NFRs: restrictions and constraints among system services

NFR Framework: is a soft GO approach that aims to link soft goals (i.e., non-functional requirements) into software systems using Softgoal Interdependency Graphs (SIGs) with full bidirectional traceability.

OWL-DL: Ontology Web Language Description logic.

PORE: approaches that are required to understand the current problem

QoS: quality of service requirements are NFRs in the service-oriented context.

Quality of Service Requirements (QoS) Identifier: The purpose of this component is to identify the QoS requirements that are associated with the identified software services capabilities.

Reconsidered BSV: The strategic view for an individual organisation is a set of interrelated GO models that elaborate each others, starting from the highest goal until getting the lowest goal in abstraction within the organisation, in order to facilitate the early understanding of the business organisation from the goals' point of view. The elaboration is represented in the form of network of goals that drive the organisation forward. The soft goals associate GO component in the GO models as their constraints or desired quality requirements and represented using their relevant methods if needed. Both hard and soft goals, that are strategical elements, are decomposed within their models into tactical elements or approaches where their run would fulfil any/both of the strategic goals such as tasks, hard resources (i.e., physical) and soft resources (i.e., data and information). The representation of strategical goals is aided with traceability in both directions (i.e., from the highest soft/goal to the lowest soft/goal in abstraction and vice versa) and with mapping to appropriate metrics (i.e., qualitative or quantitative) 'that provide ongoing evaluation of how successfully the organisation is achieving its goals' (OMG, 2013). The GO models must consider the participation of the hard and soft goals' holders and their interaction within the models as required active entities.

sBPMN: the function of this component is to semantically represent the associated BPMs using the sBPMN ontology borrowed from the super project

SD Model: describes the intentional structure of the organisation through configuring a network of dependency relationships between organisational actors.

Service capability identifier component: this component uses the sBPMN ontology in the BPAOnt in order to identify the functional boundary through extracting the designed capabilities (e.g., send task, receive task and user task) of each member and considered them for the cluster's capabilities

Service identifier component: this component involves the application of the novel Service Identification (SI) approach proposed based on the Riva Process Architecture (RPA)

SIG: the NFR framework models that are independently generated and organised in the form of catalogues that displays the soft goal-oriented models of the organisation for stakeholders

siGoal Ontology Instantiator: The function of this component is to produce the formal semantic representation of interrelated GO models for an organisation considering that the *i** framework is the backbone for the semantic representation

siGoal-sQuality Ontology Linker: The function of this component is to bridge the input that comprises of the instantiated siGoal Ontology and the instantiated sQuality Ontology for a particular organisation.

SOA: Service-Oriented Architecture is a distinct form of technology architecture designed in support of service-oriented solution logic, which is comprised of services and service compositions shaped by and designed in accordance with service-orientation

SOC: Service-oriented computing represents a new generation computing platform that encompasses the service-oriented paradigm and service-oriented architecture with the ultimate goal of creating and assembling one or more service inventory.

Software Service Identification Layer: is the second layer in the BPAOntoSOA framework is that uses the resultant BPAOnt from the first layer in order to identify the candidate software services and their associated capabilities.

SORE: the second school of requirements engineering that is concerned with generating solutions.

sQuality Ontology Instantiator: This component has the ability to formally represent a quality-oriented model such as the NFR framework

SR Model: elaborates the SD model by articulating the rationales that organisational actors have to address in addition to its role in showing the internal link between the strategic dependency relations designed in the SD model.

srBPA ontology instantiator component: its main function is conceptualising a Riva BPA for a particular organisation using the developed srBPA ontology that implements (i.e., ontologises) the concepts of the Riva BPA, the relations between them and the rules that automate the Riva method steps

srBPA-sBPMN ontology merger component: the purpose of this component is to merge the two instantiated ontologies resulted from the above two components (srBPA ontology and sBPMN ontology instantiator components) in order to derive the BPAOnt ontology for an organisation

Tropos: is a software development methodology that aims to generate a system to-be in an environment using a GO approach.

UoW: is EBE with lifetime.

Volere: requirements specification template, as a SORE.