A Semantically- Enriched Quality Governance Framework in the System of Systems context Applied to Cancer Care

Eman Ghaleb Yousef Qaddoumi

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 Research Group,
Faculty of Environment and Technology,
University of the West of England, Bristol

2020

Abstract

Organisations are becoming more complex with diverse businesses, and therefore accomplishing their business objectives entails the need to develop System of Systems (SoS) with new capabilities based on existing monolithic systems of different domains. Regardless of the business objectives of these organisations, they can only be achieved if the right level of quality is ensured across the SoS arrangement.

In order to deliver new SoS capabilities, interoperability between the SoS's Constituent Systems (CSs) is required. Semantic inconsistencies at different levels of SoS's constituent systems causes various challenges which can degrade the level of quality governance among the SoS arrangement. These inconsistencies mainly are due to the domain process' heterogeneities, multiple standards followed, policies and varying levels of quality requirements of the CSs, and hence the level of interoperability affecting the anticipated quality.

To respond to the above challenges, this research is aimed at investigating the effectiveness of semantically-enriched quality governance in relation to policies, processes, standards and quality requirements of the constituent systems in a SoS arrangement. For this purpose, a semantically enriched framework for the quality governance of SoS, i.e. OntoSoS.QM.Gov (Ontology-based System of Systems Quality Management Governance) has been developed and evaluated incrementally using an adaptation of the Design Science Research Methodology (DSRM). A sufficient and representative case study has been utilised in the DSRM process increments from the SoS cancer care domain, in particular, the Cell Therapy and Applied Genomics (CTAG) at the King Hussein Cancer Centre (KHCC), Jordan.

The OntoSoS.QM.Gov framework consists of four ontological models: (i) the SoS standards ontology model (OntoSoS.Stand), (ii) the SoS quality requirements ontology model (OntoSoS.QR), (iii) the SoS process ontology model (OntoSoS.Process), and (iv) the SoS policies ontology model (OntoSoS.Policy). They are linked together using a fit-for-purpose governance process in managing the semantics of the relevant quality governance areas.

The outcomes of demonstrating the OntoSoS.QM.Gov framework using the CTAG case study and evaluating it with the cancer care domain experts revealed the following. First, semantic heterogeneities between CSs and SoS in relation to their policies, processes, quality requirements and standards have been resolved. Second, the fit- for- purpose quality governance process was observed to mostly determining and resolving conflicts with minimum human intervention. Third, the adequacy of the four ontological governance models in capturing the semantics of governance in relation to policies, processes, quality requirements and standards not only for CSs but also as stand-alone models that may further be utilised in different contexts or domains.

Finally, this research has been able to identify further research areas to explore in relation to the governance of change management of constituent systems' processes, policies, standards where their business processes change.

To My Parents

To My Husband

To My Beloved Daughters

To My Brothers

To My Sister

Acknowledgements

All praise to Allah

I would like to thank my supervisory team Prof. ***Mohammed Odeh****, Dr.* ***Mario Kossmann*** *and Dr.* ***Zaheer Khan*** *for their continuous support and valuable feedback regarding my PhD research. I would also like to thank Prof.* ***Fairouz Kamareddine*** *and Dr.* ***Mehmet Aydin*** *for their valuable comments. Also, I would like to thank Dr.* ***Rawad,*** *Dr.* ***Hamzeh and*** *Dr.* ***Ahmad*** *for their advises regarding my research.*

I would also like to thank all staff members of the Graduate School at the University of the West of England, especially Marisa Downham for her prompt support.

I would like to acknowledge the support of all staff at King Hussein Cancer Centre, Amman, Jordan, for their support during my PhD, especially, ***Dr. Asem Mansour****,* ***Dr. Abdelghani Tbakhi****,* ***Eng. Dana Nashawati*** *and* ***Ms Ronza Al-Dwiri****.*

My sincere thanks to my colleagues who shared with me this long journey. ***Mais****,* ***Mohammad****,* ***Suhair****,* ***Abdullah****,* ***Shadan****,* ***Dalia****,* ***Noura****,* ***Bushra****,* ***Barakah and Maryam****. Also, I warmly thank my friend* ***Dr. Nashwa*** *for her support and for being my best friend.*

I am deeply grateful to my teacher and father ***Ghaleb*** *and my mother* ***Lateifah*** *for their prayers, love and their continuous support. Also, I would like to thank my beloved sister* ***Amani****, brothers* ***Ezidin****,* ***Yousef*** *and* ***Saif****, brother in law* ***Jalal*** *and sisters in law* ***Maha****,* ***Lena*** *and* ***Toqa****, and respected mother and father in law for their support and care.*

I owe thanks to a very special person, my husband ***Sami Abuezhayeh*** *for his support, love and everlasting encouragement throughout my PhD journey. I am so thankful that you kept me going strong. Also, my heartfelt thanks to my amazing daughters* ***Dua'a****,* ***Alaa'*** *and* ***Sadin*** *for their patience, unconditional support, wishes, and for filling my life with positivity and love.*

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

ADT: Admission, Discharge and Transfer 39, 40, 58, 92

BMT: Blood and Marrow Transplant 39

BPM : Business Processes Modelling...25

CAP: College of American Pathologists...38, 39

COBIT : Control Objective for Information and related Technologies...12

CSs: Constituent Systems ii, 1, 2, 5, 9, 10, 13, 24, 26, 27, 44, 51, 87, 91, 92, 94, 95, 115, 130, 134, 140, 146

CTAG: Cell Therapy and Applied Genomics...39, 40, 92, 116

FC: Flow Cytometry 39

GQGov: global quality governance 47, 52, 56

HCAC: Healthcare Accreditation Council...38, 39

JCI: Joint Commission International 38, 39

LQGov: local quality governance...47, 56

MDI: Molecular Diagnostics Immunogenetics..39

NHS : National Health Service 11, 14, 25, 27

OMG: Object Management Group 22, 25

OntoSoS.Policy: Ontology-based Systems of Systems Policy Model...... ii, 5, 40, 43, 44, 45, 47, 82, 84, 104, 105, 106, 108, 127

OntoSoS.Process : Ontology-based Systems of Systems Processes Model ii, 5, 40, 43, 44, 46, 108, 109, 110, 119, 124, 125, 127, 183

OntoSoS.QM.Gov : Ontology-based Systems of Systems Quality Management Governance Frameworkii, 5, 6, 29, 30, 31, 39, 40, 41, 43, 44, 45, 46, 47, 48, 50, 80, 82, 87, 106, 125, 127, 151, 156, 163

OntoSoS.QR: Ontology-based Systems of Systems Quality Requirements Model ii, 5, 40, 43, 44, 45, 47, 60, 61, 65, 66, 67, 68, 70, 71, 78, 79, 82, 103, 108, 123, 127

OntoSoS.Stand : Ontology-based Systems of Systems Standards Model .. ii, 5, 127, 128, 129, 130, 147, 150, 151, 183

QM : Quality Management ... ii, 2, 5, 6, 10, 29, 30, 31, 39, 40, 41, 43, 44, 45, 46, 47, 48, 50, 80, 82, 87, 106, 125, 151, 156, 163

RAD: Role Activity Diagrams...26

SoI : System of Interest...10

SoS : Systems of Systems ... ii, 1, 2, 3, 5, 6, 8, 9, 11, 12, 13, 14, 15, 22, 23, 24, 26, 27, 28, 29, 31, 33, 36, 37, 38, 39, 43, 44, 47, 48, 50, 60, 61, 63, 64, 66, 71, 73, 74, 85, 86, 90, 100, 111, 115, 121, 129, 134

UML: Unified Modeling Language 57

V&V: Verification and Validation..53

1. Chapter One: Introduction

This chapter introduces the subject area of this research; it starts by presenting a brief overview for the field of the study, quality governance in the context of System of Systems (SoS), in Section 1.1. Then, the current research problem and the motivation behind it are introduced in Section 1.2. In Section 1.3, the main research aim and its objectives are presented, followed by the research questions and hypothesis in Section 1.4. The research main contributions are listed in Section 1.5. At the end of this chapter, a schematic overview of the structure of the PhD thesis is provided in Section 1.6.

1.1. Overview

The complex nature of today's organizations and their demand to accomplish new objectives entail the need to develop System of Systems (SoS) based on cooperation between existing monolithic systems of different domains but delivering new capabilities (*JCIDS*, 2005; Maier, 1998; Nielsen et al., 2015). A monolithic system "is called monolithic if distinguishable services are not clearly separated in the implementation but are interwoven" (Ceccarelli et al., 2016).

SoS can be defined as "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" (*Defense Acquisition University*, 2011). Some of these capabilities cannot be achieved without an interoperability between two or more heterogeneous Constituent Systems (CSs) (Bianchi et al., 2015; Gansler et al., 2012; *DoD*, 2008). This heterogeneity can be attributed to the constituent systems' different domains, associated processes, resources, etc. (DoD, 2008).

In order to provide new capabilities and to successfully deliver SoS business objectives, it is necessary to provide means of coordination between CSs (Gansler et al., 2012; Gheorghe et al., 2018). However, the heterogeneity of the CSs can negatively affect several quality-related objectives to be fully satisfied and managed, e.g. safety, security and performance related objectives (Bianchi et al., 2015; Dillon et al., 2011; "SEBok," 2015; "SEBok," 2017). Therefore, managing the quality and adhering to SoS quality requirements is a challenging task that needs to be effectively attained to (Bianchi et al., 2015; *DoD*, 2008; Wagner, 2013).

In the case of SoS, the synergism between heterogeneous CSs may lead to many conflicts between them (Nielsen et al., 2015). For instance, conflicts may occur when an

SoS arrangement is inspected against adherence to security policies, processes, and standards (Nielsen et al., 2015). Such conformance assessment related to quality requirements in SoS context has led the way to identify the research gap attended in this thesis.

One example of the SoS arrangement is the cooperation between several monolithic CSs (e.g. pharmacy system, treatment systems, cancer registry system, etc.) to form a cancer care SoS arrangement. This arrangement provides a holistic approach to cancer care and provides new capabilities which cannot be achieved by using monolithic systems.

1.2. Research Problem and Motivation

The dynamics in the evolving structure of the SoS and constituent systems' heterogeneity introduce new challenges (*JCIDS*, 2005). Some challenges can overlap with the existing challenges specific to monolithic systems, while others may be due to the emergent behaviour of the SoS arrangement, e.g. in terms of interoperability between the SoS constituent systems and the complexity of the SoS authority structure ("SEBok," 2017). Many of these challenges are closely related to the Quality Management (QM) aspects of the SoS ("SEBok," 2017) and to control and management issues (DoD, 2008; Jamshidi; 2008). This research is motivated by such challenges and more specifically quality management challenges and their constraints in an SoS context.

Quality Management (QM) is defined as "the process of ensuring that the required level of quality is achieved" ("SEBok," 2015). The goal of QM is to ensure that the implementation of quality management meets organizational and project quality objectives (*INCOSE*, 2015). To ensure quality is achieved, the definition of procedures, processes, and standards that are aimed at ensuring that software quality is achieved (Sommerville, 2011).

One approach to handle QM-related issues is via governance approaches (Martins et al., 2017). In general, the need for SoS governance is grown (Gheorghe et al., 2018). Governance is defined as the intentional usage of policies, procedures, and organizational structures to make decisions to achieve the desired business objectives (Bieberstein et al., 2008).

(Campbell et al., 2001; Martins et al., 2017) show the importance of quality governance in adhering to quality requirements and improving quality in complex systems. Quality governance is one of the main challenges in the SoS context for the purpose of fulfilling quality requirements (Holley et al., 2006; Holley and Arsanjani, 2010a) and it is a more complex process in SoS compared to monolithic systems (Cadavid et al., 2020;

DoD, 2008). This research investigates the effectiveness of developing an SoS quality governance framework to inform adherence to SoS quality requirements.

The complexity of SoS arrangements and interoperability issues initiate the need for semantic representation of variance governance elements in order to have a consensus amongst the constituent systems. One example of an interoperability issue in SoS arrangement is the mismatch between the constituent systems in SoS arrangement that is related to understanding the concepts provided by them which resulted because of the situations of operating each constituent system are different from each other (Dahmann, 2014; *INCOSE*, 2015). Thus, it is recommended to use semantic representation to represent knowledge between heterogeneous domains in order to detect and resolve semantic conflicts resulting from the interoperability between constituent systems (Benammar et al., 2015).

Many governance frameworks (Musa et al., 2014; Sloane et al., 2008) have been reported in the literature. However, it appears that there is a lack of SoS quality governance frameworks (Qaddoumi et al., 2018b). Many available frameworks have been developed for very specific domain systems, e.g. IT governance in an education system in a university (Musa et al., 2014), have deficiency in covering important elements of adherence to quality governance in relation to processes and standards, or are only theoretical frameworks (Calida et al., 2016).

1.3. Research Aim and Objectives

The current research is aimed at investigating whether a semantically enriched approach to manage quality governance in system of systems context results in identification and resolving semantic heterogeneities emerging from the interactions between the individual constituent systems in relation to their policies, processes, standards and quality requirements.

In order to respond to the main research aim, the following research objectives have been formulated:

1. Investigate the main areas and related issues in the literature that have not been addressed and the impact of their current limitations for the purpose of informing SoS quality governance;
2. Investigate the developing of a SoS quality governance related models that represent the identified issues using a semantically enriched approach;
3. Investigate the process used to implement the interactions between the semantically enriched models to identify and resolve semantic heterogeneities; and

4. Investigate the effectiveness of the developed process to identify and resolve semantic heterogeneities in the SoS context.

The above aim and associated objectives have been established as the basis to bridge the gap in literature in relation to SoS quality governance and for the development of a novel quality governance framework in the SoS context employing semantic modelling approach to capture the required semantics of the policies, business processes, standards, and quality requirements in the constituent systems, and then, to identify and resolve heterogeneities emerged from the interoperability between them to enable the adherence to quality requirements. The research aim and objectives have been utilised to formulate the research hypothesis and its associated research questions as presented in the next section.

1.4. Research Hypothesis and Questions

The research hypothesis asserts that "***Using semantics to represent the interaction between policies, standards, processes and quality requirements of constituent heterogeneous systems in a System of Systems arrangement results in an effective SoS quality governance framework***".

Some terminologies mentioned above need to be clarified according to the research context. Firstly, *semantic* is defined as "the relationships of symbols or groups of symbols to their meanings in a given language"("ISO 9241-210," 2010). Some researchers indicated that the best approach to semantic specification is based on an ontology-based description (Benammar et al., 2015; Sommerville, 2010). Ontology is "an explicit specification of a conceptualization" (Gruber, 1993). Ontologies are the most used approach to represent semantic knowledge between heterogeneous domains and to detect semantic conflicts resulting from the interoperability between constituent systems (Benammar et al., 2015). Hence, a quality governance framework is anticipated to benefit from the ontology-based description, where the precise meaning of terms and the relationships between different terms in a description is defined in an ontology. Secondly, *heterogeneity* occur when there is a disagreement about the meaning, interpretation, or intended use of the same or related data (Sheth and Larson, 1990). According to Oxford dictionary, heterogeneity means "the quality or state of being diverse in character or content".

The above hypothesis triggers the formulation of the following Research Questions (RQs) listed in Table 1-1. All RQs will be answered in the coming chapters in order to prove or disprove the research hypothesis.

Table 1-1 Research Questions

	Research Questions (RQs)
RQ1	What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context?
RQ2	How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach?
RQ3	How will the semantically- enriched models of policies, processes, standards and quality requirements interact in the system of systems context to identify and resolve semantic heterogeneities?
RQ4	How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities?

1.5. List of Research Contributions

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

1- The **Onto**logy-based **S**ystem **o**f **S**ystems **Q**uality **M**anagement **Gov**ernance Framework (i.e. OntoSoS.QM.Gov): the principal artefact resulted of this research study. This framework consists of four main models: standards, policy, quality requirements and processes. These models are linked to each other to formulate the quality governance framework in order to adhere to quality governance by resolving semantic heterogeneities between CSs and SoS in relation to the four models.

2- Ontological representation of the four key SoS quality governance components: (i) SoS standard ontology model (i.e. OntoSoS.Stand), (ii) SoS quality requirements ontology model (i.e. OntoSoS.QR), (iii) SoS policies ontology model (i.e. OntoSoS.Policy) and (iv) SoS processes ontology model (i.e. OntoSoS.Process). Each ontological model can be used as a stand-alone model that may be utilised in different contexts or domains.

3- SoS quality governance process that can employ selecting quality governance related policies, processes, standards and quality requirements in SoS context to determine and resolve the conflicts between the CSs and SoS in relation to their policies, processes, standards and quality requirements.

4- The research design framework developed and followed in this research that can be adopted and reused in similar research. It is featured with its incremental and iterative nature following the Design Science Research Methodology methodology combined with literature and case study methods.

1.6. Thesis Overview

After introducing the research in this chapter, the background and literature review are discussed in **Chapter 2**. In particular, governance, SoS, ontologies and semantic representation, the research gap analysis and quality governance- related areas. **Chapter 3** presents: (i) the research design with emphasis on Design Science Research Methodology (DSRM) utilised in this research (ii) the main research artefact "OntoSoS.QM.Gov framework", and (iii) the selected research case study. **Chapter 4** discusses the design, demonstration and evaluation of the first increment of the OntoSoS.QM.Gov framework employing the SoS Quality Requirements (QR) model. **Chapter 5** discusses the design, demonstration and evaluation of the SoS policies model and the interaction process between the SoS policies model and the SoS quality requirements model. **Chapter 6** discusses the design, demonstration and evaluation of the SoS processes model and the interaction process between the SoS processes model, SoS policies model and SoS quality requirements model. The design, demonstration and evaluation of the SoS standards model and the interaction process between the SoS standards model, SoS processes model, SoS policies model, and the SoS quality requirements model are discussed in **Chapter 7**. Finally, answering research questions, summary of this research outcomes and the suggested future research directions are presented in **Chapter 8**. At the end of this thesis, the references and appendices are listed.

Figure 1-1shows the thesis roadmap to answer the research questions. The first RQ is fully answered in Chapter 2 by surveying the literature to report the gaps in the literature in relation to SoS quality governance. RQ2 is partially answered in Chapter 2. However, due to the nature of the incremental development and evaluation of the quality governance framework, the RQ2, RQ3, and RQ4 will be incrementally answered in Chapters 4,5,6 and 7.

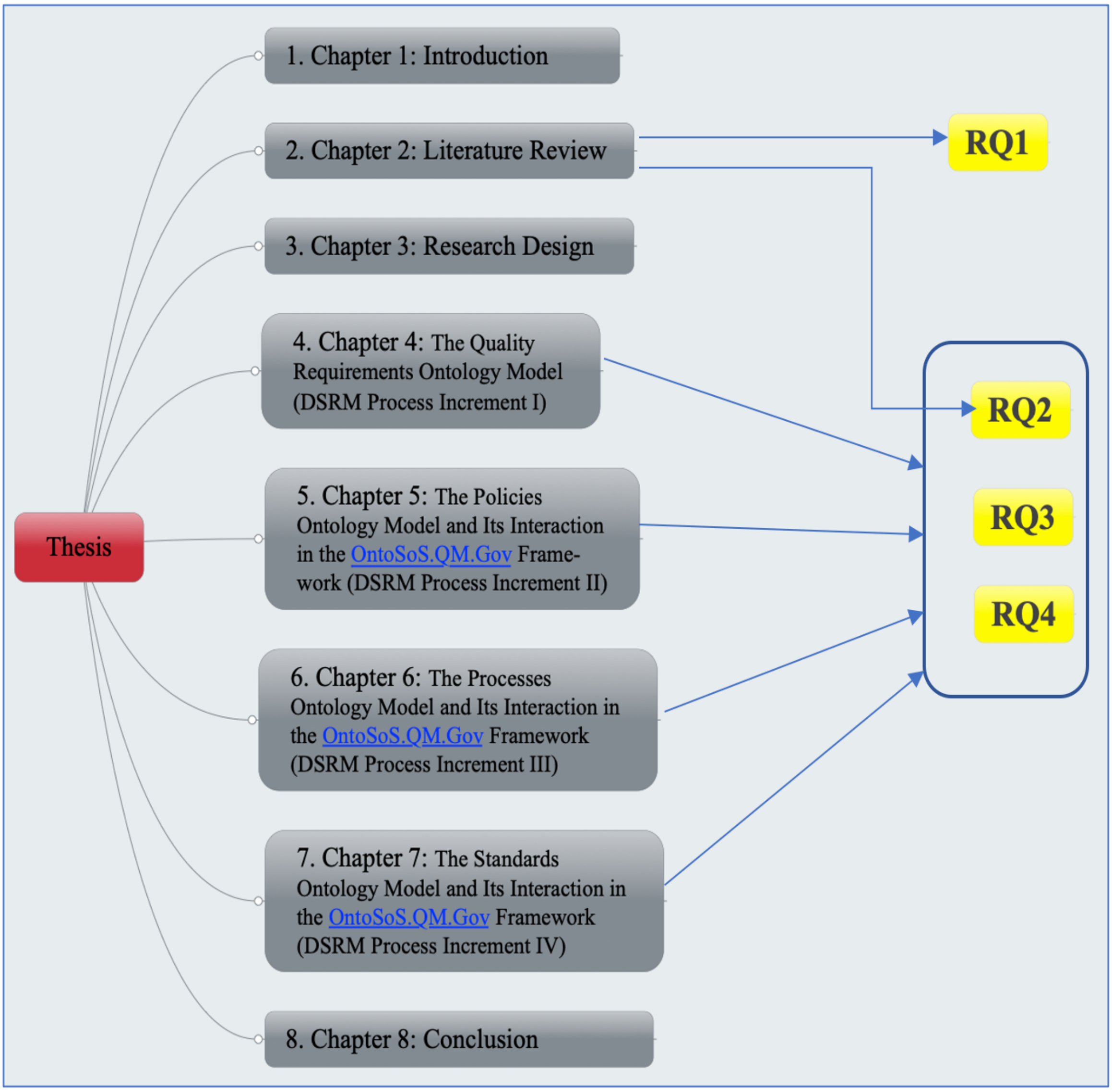

Thesis
1. Chapter 1: Introduction
2. Chapter 2: Literature Review
3. Chapter 3: Research Design
4. Chapter 4: The Quality Requirements Ontology Model (DSRM Process Increment I)
5. Chapter 5: The Policies Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment II)
6. Chapter 6: The Processes Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment III)
7. Chapter 7: The Standards Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment IV)
8. Chapter 8: Conclusion
RQ1
RQ2
RQ3
RQ4

Figure 1-1: Thesis Roadmap to Answer Research Questions

2. Chapter Two: Literature Review

2.1. Introduction

In order to identify and investigate the gaps in the current related research areas to SoS quality governance, this chapter begins by firstly providing background knowledge about all related aspects of this research. Hence the main focus is on the SoS, quality management, governance, ontologies and semantic representation.

Then, the issues and current limitations of the models of quality governance-related areas (i.e. quality requirements, policies, processes and standards) and the models are reported and assessed based on several aspects, (i) the semantic representation of the models, (ii) the existence of the SoS-related concepts, (iii) the interactions between the different concepts with other models, (iv) the existence of detailed information regarding the design and implementation of the models, (v) the application industry of the models and (vi) the evaluation methods followed.

A reflection on the research gap analysis is detailed in Section 2.12. This led to identify the research directions, to formulate the definition of the governance and the components of the governance framework which led to formulate the research questions and research hypothesis introduced in Chapter 1. Finally, a summary and conclusion section will be presented in Section 2.13.

Figure 2-1 shows the process followed for literature review. After the main research problem has been identified, firstly, the general areas related to the current study have been identified. These areas are SoS concepts and associated issues, quality management and quality requirements, governance related issues, and semantics and ontology construction methods. Secondly, the first step has led to identify and to explore the main areas and models related to SoS quality governance: (i) policies, (ii) standards, (iii) quality requirements and (iv) processes. The main limitations of these models were identified and the interactions points between them were determined. Then, the gaps resulted of the previous step were reported. Identifying these gaps led to determine the main aspects related to the design and evaluation of the suggested solutions.

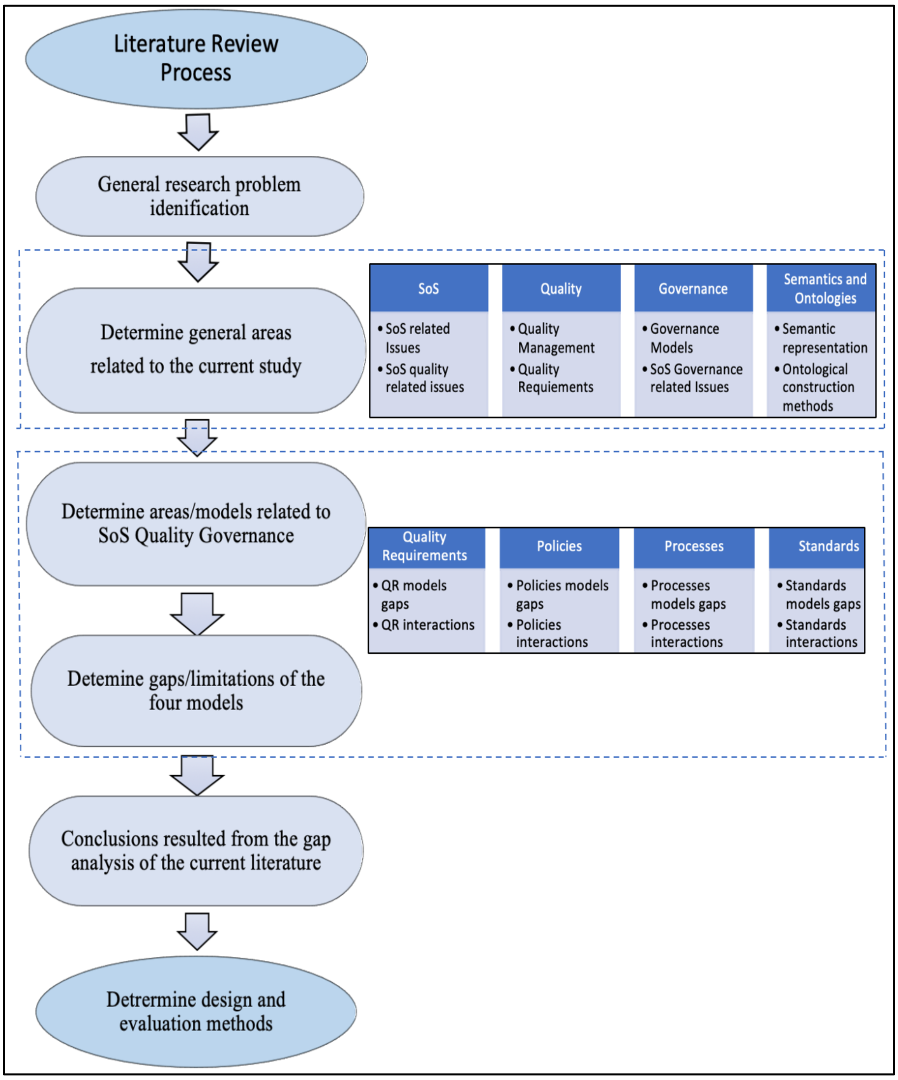

Literature Review Process
General research problem idenification
Determine general areas related to the current study
SoS
• SoS related Issues
• SoS quality related issues
Quality
• Quality Management
• Quality Requiements
Governance
• Governance Models
• SoS Governance related Issues
Semantics and Ontologies
• Semantic representation
• Ontological construction methods
Determine areas/models related to SoS Quality Governance
Quality Requirements
• QR models gaps
• QR interactions
Policies
• Policies models gaps
• Policies interactions
Processes
• Processes models gaps
• Processes interactions
Standards
• Standards models gaps
• Standards interactions
Detemine gaps/limitations of the four models
Conclusions resulted from the gap analysis of the current literature
Detrermine design and evaluation methods

Figure 2-1: Roadmap to Literature Review

2.2. System of Systems

A system is “an integrated set of elements or subsystems that accomplish a defined objective” (INCOSE, 2015). These elements consist of products, processes, people, information and any other support elements (INCOSE, 2015). However, organizations are becoming more complex with diverse businesses (e.g. transportation, shipping, security, product customization, enhancing quality of services after sale, etc.) and hence accomplishing their objectives entails the need to develop SoS based on a form of cooperation between existing monolithic systems of different domains in order to deliver new capabilities. JCIDS (2005) defines SoS as “a set of systems that are connected together to provide capabilities that the loss of any of the CSs will affect the performance or capabilities of the whole”.

Despite much research being done in the field of SoS, there is still no universally agreed definition for SoS (Gandhi et al., 2012). Many definitions are available in the literature. For instance, in (Jamshidi, 2008), the author defines the SoS as “a collection of individual, possibly heterogeneous, but functional systems integrated together to enhance the overall robustness, lower the cost of operation, and increase reliability of the overall complex SoS system". Another definition is introduced by the systems engineering handbook of the International Council on Systems Engineering (INCOSE): "SoS is an System of Interest (SoI) whose elements are managerially and/or operationally independent CSs. These interoperating and/or integrated collections of constituent systems usually produce results unachievable by the individual systems alone" (INCOSE, 2015).

The distinguishing characteristics to realise a SoS were introduced by Maier (Maier, 1998): operational and managerial independence of CSs, emergent behaviour, evolutionary development processes and geographical distribution. Therefore, the dynamics in the evolving structure of the SoS, constituent systems' heterogeneity, and emergent behaviour introduce new challenges (Jamshidi, 2008; Samad and Annaswamy, 2011). Some challenges can overlap with the existing challenges specific to monolithic systems, while others may be due to the emergent behaviour of the SoS arrangement, e.g. in terms of interoperability between the SoS' constituent systems (Sledge, 2010) and the complexity of the SoS authority structure (DoD, 2008). Many of these challenges are related to control and management issues (Jamshidi, 2008; DoD, 2008), e.g. governance and to the Quality Management (QM) aspects of the SoS ("SEBok," 2015), e.g. adhering to software quality requirements.

The literature reported the need for monitoring and continuously checking the adherence of systems to their software quality requirements, e.g. privacy requirements (Robinson, 2006; Vierhauser et al., 2016) as these requirements need to be consistent with their policies and standards (Robinson, 2006). Another example of such emerging properties is the safety as a system quality requirement (J. Black and P. Koopman, 2009). Safety levels needed at the SoS level and their meanings do not have the same meaning in the monolithic CS. In general, SoS quality requirements are not addressed well in the recent research (Cadavid et al., 2020). However, the literature showed the importance of interoperability as a quality requirement in the SoS field (Cadavid et al., 2020).

SoS can be created within wide variety of domains, e.g. industry, transport, defense, health care, energy, etc. However, there is a lack of the research that is related to SoS health care domain (Cadavid et al., 2020). One example of a health care domain is the cancer care. It has been proven in the literature that cancer care is a complex SoS (Gorod et al., 2018). So, a case study from the cancer care domain is considered a beneficial case study to this research.

2.3. SoS Quality Management

Quality- related issues of the SoS is a vital challenge that has not yet been overcomed in the traditional software engineering (Cadavid et al., 2020; Nielsen et al., 2015; *DoD*, 2008). Quality Management (QM) has been identified as "the process of ensuring that the required level of quality is achieved in software deliverables" ("SEBok," 2015). The goal of QM is to outline the policies and procedures required to improve and control the different processes within the organisation that lead to improve business performance (INCOSE, 2015).

Many researchers indicated that business performance is affected by governance, which deals with policies and procedures (Hendrikse and Hendrikse, 2003). Currently, one of the main QM challenges for SoS is to govern SoS arrangements (Holley et al., 2006; Holley and Arsanjani, 2010b; Martins et al., 2017; *DoD*, 2008) as they are more complex compared to monolithic systems (DoD, 2008). Therefore, quality governance is a key research area for the development and the operational support of SoS.

2.4. Governance

The verb "to govern" is defined in the Cambridge Dictionary ("Cambridge Dictionary," 2018) as "to control and direct, or to have a controlling influence on something". The concept "governance" has several definitions in the literature. For instance, according to (Bieberstein et al., 2008), governance is defined as the intentional usage of policies, plans, procedures, and organizational structures to make decisions and control an entity to achieve the desired business objectives.

A definition of SoS governance was proposed in (Mansouri and Mostashari, 2010), introducing enterprise systems' governance as a new systemic approach to governance within the SoS environment. The authors defined SoS governance as "a process that uses the infrastructure of authority for defining restrictions and allocating resources within the boundary of extended enterprise systems in order to coordinate activities, facilitate communications, increase interoperability, resolve conflicts, assist holistic decision or policy making, enhance SoS-level legitimacy for belonging, and improve effective participation as well as interconnectivity among constituent systems of the enterprise network".

The UK's National Health Service (NHS) (NHS, 2011) introduced a quality governance framework that consists of four components, which are important to increase the quality of patients' care in terms of safety and positive patient experience. These components are strategy, capabilities and culture, processes, and measurement. NHS

(2011) refered to quality governance as “the values, behaviour, structure and processes that are necessary to help the members who are responsible of quality governance in the organisation to ensure that levels of quality and safety are met”.

A widespread governance framework is Control Objective for Information and related Technologies (COBIT) framework (ISACA, 2012; Oliver and Lainhart, 2012). It is a comprehensive IT governance framework that covers all management levels in the organization and it is well suited to enterprises focused on risk management (ISACA, 2012).

The semantic/ontology modelling was not taken into consideration in both the NHS and COBIT frameworks. Also, the perspective of SoS is not clearly attained in the documentation of the aforementioned models. The main quality requirements that these models focus on are safety in the case of the NHS framework and security in the case of COBIT framework. Despite the widespread of the COBIT framework in information systems organisations, there is limited academic research that analyses COBIT and detail the specifications of the connections between its processes (Bartens et al., 2015, p. 4558).

The authors in (Calida et al., 2016) proposed a theoretical SoS governance model by providing a set of systems- based concepts to represent SoS governance. Their contribution was to provide some main governance- related concepts which enhanced the understanding of the associated phenomena of the design of the complex SoS governance. For instance, they suggested to focus on interoperability and to have a flexible governance design for the constituent systems to deal with emergence principles resulted from the dynamic integration between the constituent systems. Also, they said that there is a need to build a central repository to store all governance-related data. The authors said that one of SoS challenges is the conflict between roles of different actors in each constituent system. Two limitations to their work is the theoretical side of this model and the lack of quality management concepts of SoS. However, the authors said that a future development to their work is to apply their theoretical framework to understand how it might be deployed.

Most of the former governance frameworks and definitions deal with general and theoretical governance, e.g. the framework of (Calida et al., 2016). Hence, it might be vital to investigate a more general quality governance definition in the SoS context.

2.5. SoS Quality Requirements and Related Issues

Quality requirements (QRs) are vital inputs in many governance frameworks and standards in literature (e.g. ISO 38500, COBIT 5 and ISO 9001).

A Systematic Literature Review (SLR) (Bianchi et al., 2015) has been conducted and identified the most common challenging quality attributes in SoS. Also, the study concluded that the current quality models cannot fully address most of the quality attributes in the SoS context. This initiates the need to develop a more generalised model that can conceptualise the common challenging quality attributes in SoS. This SLR investigated the literature that mainly have keywords that are focusing on SoS and QRs. The main reported SoS quality requirements are security, interoperability, performance, reliability and safety. The heterogeneity of the constituent systems and the related interoperability issues are the main challenge to deal with in the SoS arrangement (Cadavid et al., 2020; Nielsen et al., 2015). Therefore, a solution is needed to conceptualise and resolve the heterogeneity and interoperability related issues in the SoS arrangement.

2.6. SoS Governance Related Issues

To successfully govern SoS operations, the above challenges need proper solutions. One way to manage a challenge is to find the suitable procedures, resources and frameworks to degrade or prevent the influence of the challenge on satisfying the goals of the SoS, this done after identifying the factors leading to a challenge.

Firstly, there are many SoS challenges related to quality aspects (SEBok, 2015). Typically, SoS are comprised of multiple independent systems with their own quality requirements, working toward broader capability objectives. In many cases, the SoS arrangement goals and needs may not be consistent with the requirements of the constituent systems (Dahmann, 2014). For instance, the security requirement has different definitions and variant levels (INCOSE, 2015) and the challenge of identifying and documenting all stakeholders' quality needs in the SoS arrangement (INCOSE, 2015). Because of the emergent characteristics of SoS, some new quality requirements may appear to overcome emergent quality constraints issues (DoD, 2008). So, system engineers and the people who are responsible of the process for applying quality governance need to identify stakeholder's quality requirements at SoS level and monolithic systems level (INCOSE, 2015; DoD, 2008). The former challenges initiate the need to govern the quality requirements related aspects at both SoS and CSs levels. Also, the author in (Keating, 2015) concluded that governance can assist SoS engineers to deal with its associated issues.

Secondly, Controlling and management activities of organisations are usually facilitated by a set of policies and internal controls (Van Arnum, 2004). In the case of SoS, many inconsistencies may arise among the policies of monolithic systems. For instance, the conflicts that may emerge from policy specifications due to the conditions under which a conflict in the policy specifications may occur, specific rules can be defined to detect conflicts (Flegkas et al., 2005). The policy is “a set of rules used to guide and determine decisions” (Susanti and Sembiring, 2011). Also, as in (Mansouri et al., 2009) policy provides quality assurance guidelines for the system’s activities and processes. There are several policy-related activities and artefacts that are needed to be governed. For example, a vital element to be governed is identifying the roles and responsibilities of the different types of the stakeholders. The roles and responsibilities of individuals and groups are outlined by the policies of the organisation (Musa et al., 2014). The former challenges initiate the need to govern the SoS policies along with the associated roles and responsibilities.

Thirdly, one challenge is that processes have to be defined to monitor the use of the standards and check that they have been followed (Sommerville, 2001). Processes are “collections of interrelated tasks that are enacted to accomplish the purpose of the governance” (Bannerman, 2009). Effective processes and structures that are recognized and understood by a governing body will help to successfully govern and satisfy quality constraints (NHS, 2011). Also, the multiple authority levels (Dahmann and Baldwin, 2008) and the characteristics of SoS (Boardman and Sauser, 2006; Maier, 1998) raise several challenges to SoS. These challenges entail the need for governing processes and related activities.

Another issue that is more challenging in SoS than it is in systems is the interoperability between the constituent systems of the SoS arrangement (Sledge, 2010). Most quality problems can be addressed by deploying the correct metrics and standards. For example, metrics should be utilised to provide the systems engineer a feedback on the state of SoS capabilities being formed (DoD, 2008). This initiate the need to govern the standards of SoS arrangement.

SoS systems engineers have a wider role than in the case of monolithic systems and therefore, they need to take into wider context considerations both technical (e.g. control, test, evaluate) and non-technical (e.g. policy, strategy, resources) issues (Dahmann et al., 2009). They need to work through the constituent systems to investigate the top-level capability needs as well as achieving the required coordination for the development and evolution of the constituent systems (Dahmann and Baldwin, 2008). This coordination is vital to achieve the required governance (Musa et al., 2014). Also, DoD (DoD, 2008) focused on the importance of integrating the efforts across multiple independent constituent systems in order to have an effective SoS governance. This show the importance of a consensus on the different aspects between the constituent systems.

Also, “because the systems were developed and operated in different situations, there is a risk that there could be a mismatch in understanding the concepts provided by one system to the SoS if the particular systems’ context differs from that of the SoS” (Dahmann, 2014; INCOSE, 2015). So, a consensus on quality governance concepts needs to be deployed for both constituent systems level and SoS level. This shows the importance of the semantics representation of the SoS quality governance related concepts.

In summary, the heterogeneous SoS arrangement introduces many quality governance- related challenges. These challenges can be categorised into several areas. As discussed in this chapter, some of these areas are related to (i) quality requirements, (ii) standards (iii) policies (iv) processes and (v) heterogeneity and interoperability. The interoperability issue of the SoS arrangement initiates the need for semantic representation modelling for the governance elements in order to have a common understanding among the constituent systems. Ontologies are the most used approach to represent semantic knowledge between heterogeneous domains, and to detect and resolve semantic conflicts resulting from the interoperability between constituent systems (Benammar et al., 2015). Next, the identified areas will be reviewed, and the gaps and limitations of the related models will be assessed.

2.7. Ontologies and Semantic Representation

Ontologies provide a formal description of real world objects and their relationships within a domain (Smith et al., 2004). As illustrated earlier, ontologies are the most frequently used approach to represent semantic knowledge between heterogeneous domains, and thereby provide robust mechanisms to detect and resolve semantic conflicts (Benammar et al., 2015). In general, it is recommended to use semantic modelling in the case of heterogenous systems (Bischof et al., 2014; Sciore et al., 1994; Uschold and Gruninger, 2004; Yang et al., 2017).

The origin of the term ontology is in philosophy, where it is mainly concerned with the study of the nature of existence. Ontology has become a technical concept in the computer and information science discipline, where it is typically defined as “an explicit and formal specification of a conceptualisation” (Gruber, 1993). Conceptualisation is further defined as the intended models within which a set of logical axioms are designed to account for the intended meaning of a vocabulary (Guarino, 1998). Ontologies provide a formal description of concepts and their relationships within a domain (Smith et al., 2004), which results in a shared understanding.

According to literature, there isn't a widely accepted ontology construction methodology in the literature (Iqbal et al., 2013; Fernández-López, 1999). However, in the current research, only one methodology needs to be used in the process of constructing each ontology model. There are several ontology building methodologies. A brief overview of some methodologies will be illustrated in this section. These methodologies are the methodology of Gruninger and Fox (1995), the methodology of Bernaras, Laresgoiti and Corera (1996), the methodology of Breitman and do Prado Leite (2003) and the methodology of Noy and McGuinness (2001).

Gruninger and Fox, 1995

An early methodology was introduced by Gruninger and Fox in 1995 (Grüninger and Fox, 1995). They used their methodology to build the Toronto Virtual Enterprise (TOVE) project ontology within the domain of business processes and activities modelling. They use motivating scenarios which are examples which are not adequately addressed by existing ontologies or story problems that arise in the application, so "it is an application-semidependent strategy" (Fernández-López, 1999). One drawback of this methodology is to "suppose that the ontology concepts and relationships could be easily derived from the motivation scenarios" (Breitman and Leite, 2003). However, they didn't introduce details regarding the activities and techniques that they used (Fernández-López, 1999).

Bernaras, Laresgoiti and Corera, 1996

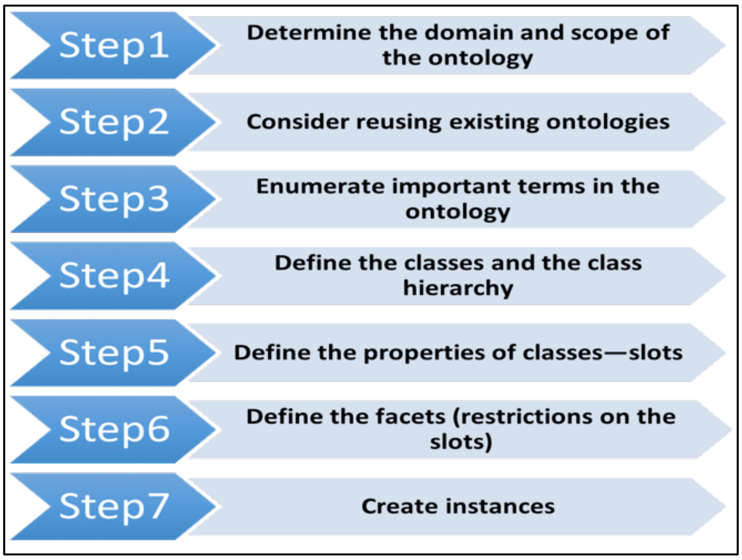
Another early methodology was introduced by Bernaras, Laresgoiti and Corera in 1996 that called KACTUS approach (Bernaras et al., 1996) . This method was used in the field of electrical networks. A main advantage of this method is that "the design process involves searching ontologies developed for other applications, which are then refined and extended for use in the new application" (Fernández-López and Gómez-Pérez, 2002). On the other hand, this approach has some drawbacks. For example, the authors provide very little details about their methodology (Fernández-López, 1999). Also, this method uses only the top-down strategy for identifying the concepts of the ontology (Fernández-López, 1999).

Breitman and do Prado Leite, 2003

Contrary to the Bernaras et.al. approach, the methodology of Breitman and do Prado Leite (Breitman and Leite, 2003) uses the bottom up strategy for constructing the ontology which may make it not suitable to all applications. Although the elicitation strategy is time consuming and application-semi dependent, a main advantage for this approach is the power of the elicitation strategy e.g. structured interviews, document reading and questionnaires.

Noy and McGuinness, 2001

Finally, Noy and McGuinness (Noy and McGuinness, 2001) had proposed a simple construction process that consists of seven steps as in Figure 2-2. This methodology allows using combination of both top-down and bottom-up approaches for building the class hierarchy which makes it suitable for many applications. The bottom-up approach starts with the definition of the most specific classes (i.e. leaves classes) based on the instances available in the real world, with subsequent grouping of these classes into more general concepts. The top-down approach starts with identifying the most general concepts, organising them into a high-level taxonomy, and proceed to more specific concepts (i.e. subclasses). Also, this methodology is suitable for brainstorming concepts and sub-concepts of a knowledge domain.



Step1 Determine the domain and scope of the ontology

Step2 Consider reusing existing ontologies

Step3 Enumerate important terms in the ontology

Step4 Define the classes and the class hierarchy

Step5 Define the properties of classes—slots

Step6 Define the facets (restrictions on the slots)

Step7 Create instances

Figure 2-2: The Steps of Building the Initial Version of The Ontology Model (adopted from [Noy and McGuinness, 2001])

In their methodology, Noy and McGuinness proposed three main rules:

(i) There is no one correct way to model a domain - there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.

(ii) Ontology development is necessarily an iterative process.

(iii) The concepts in the ontology should be close to object (physical or logical) and relationships in the domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.

The methodology of Noy and McGuiness has several advantages. For example, the rules of Noy and McGuiness methodology assist in making design decisions during ontology development (Iqbal et al., 2013). Also, it is the simplest methodology for building domain ontology (Noy and McGuiness, 2001) and it is worth mentioning that Noy and McGuiness methodology may be the most used methodology among researchers comparing to the former methods, this is reflected from both the enormous number of citations to this method comparing to other methods and the research work that was conducted by researchers who used this method, e.g. Ontology-driven Requirements Engineering Methodology (OntoRem) (Kossmann et al., 2008), ontology-based framework for identifying services from business process architecture (BPMOntoSOA) (Yousef et al., 2009a), the generic enterprise information architecture ontology framework (gEIAOnt) (Ahmad, 2015), etc.

	Methodology	**Application-independent**	**Supported with details**	**Number of citations in the literature**	**Strategies for identifying concepts**	**Simplicity (Easiness)**	**Support reusability**	**Allows Brainstorming**	**Allows overall repetition/ (Development type)**
1	Gruninger and Fox (1995) (TOVE)	-	√	1484	Middle -out	-	-	√	-- (Stage based)
2	Bernaras et.al (1996) (KACTUS)	-	- Insufficient details	346	Top - down	-	√	-	- (Modular development)
3	Noy and McGuinness (2001) (101 METHOD)	√	√	5144	Top – down, Bottom – up or Combination of both	√	√	√	√ (Evolving)
4	Breitman and do Prado Leite (2003) (LEL)	-	√	123	Bottom - up	-	-	-	-- limited iteration for identifying generalisation (Stage based)

Table 2-1 summarises some features of the former ontology methods and shows some similarities and differences between them.

Table 2-1:Comparison between several Ontology Construction Methods

	Methodology	Application-independent	Supported with details	Number of citations in the literature	Strategies for identifying concepts	Simplicity (Easiness)	Support reusability	Allows Brainstorming	Allows overall repetition/ (Development type)
1	Gruninger and Fox (1995) (TOVE)	-	√	1484	Middle -out	-	-	√	-- (Stage based)
2	Bernaras et.al (1996) (KACTUS)	-	- Insufficient details	346	Top - down	-	√	-	- (Modular development)
3	Noy and McGuinness (2001) (101 METHOD)	√	√	5144	Top – down, Bottom – up or Combination of both	√	√	√	√ (Evolving)
4	Breitman and do Prado Leite (2003) (LEL)	-	√	123	Bottom - up	-	-	-	-- limited iteration for identifying generalisation (Stage based)

2.8. Quality Requirements Models

There are many works in the literature that used ontology to model the quality requirements in the domain of service-based systems. For example, in (Glen Dobson et al., 2005), the authors developed an ontology for Quality of Services (QoS). The authors planned to use their ontology to choose between service compositions which accomplish the same task. They want to use their ontology to give the ability to combine QoS data with a workflow. This can be done by modelling the way of how several QoS metrics aggregate when multiple services are composed and produce overall QoS values for the workflow(s).

They have concentrated primarily on the dependability QR in their ontology. Also, they detailed the core concepts of QoS (e.g. attributes, metrics). Their research provides detail descriptions of non-functional requirements; reasoning behind such requirements during design, specification, monitoring, negotiation, and provision of the relevant levels of service once the system is deployed (Dobson et al., 2005). In their ontology, they adopt the dependability classification of Avizienis et. al. (Avizienis et al., 2004) which consists of six attributes: availability, reliability, safety, integrity, maintainability and confidentiality. To evaluate and demonstrate this ontology, they developed a tool for service differentiation and selection based upon QoS requirement.

One limitation of Dobson model is that it does not specify a large set of classes and properties to represent all QRs. For instant, this work lacks the concepts and attributes that related to interoperability with external systems. Fortunately, their ontology can be easily extended with new concepts, as it designed to be modular in nature, so, it will be easily used as a hole or using only one part of it (Dobson et al., 2005). So, the current research will construct the QR ontology by extending some modules of their ontology. Another drawback is that it does not show support for QoS relationships.

A second ontology model is the WS-QoSOnto model. Authors in (Tran et al., 2009) built a web services QoS ontology, called WS-QoSOnto. This ontology provides an excessive detail regarding QoS information. QoS properties describe non-functional aspects of Web services and they are used to evaluate the degree that a Web service satisfy specified quality requirements in a service request (Tran et al., 2009).

The WS-QoS ontology consists of five parts: part one describes several roles in specifying QoS information, QoS level, and QoS group. The second part explains main characteristics of QoS properties. The third part outlines relationships between and among QoS properties. The fourth part defines QoS metrics. The last part describes a set of core QoS properties.

A third ontology model is the DMAL-QoS model. Authors in (Zhou et al., 2005) designed a web service domain called DMAL-QoS ontology that mainly deals with the non-functional aspect of the system. It contains three layers: the QoS profile layer, which is used for matchmaking purposes; the QoS property definition layer, which is used for elaborating the property's domain and range constraints; and the metrics layer that provides measurement details. The main idea is to transfer the problem of judging QoS constraints conformance to the problem of judging Ontology subsumption relationships. This ontology shows some problems regarding the QoS metrics, so, they developed the layer of metrics in their work in 2005 (Zhou et al., 2005).

A drawback of this modelling approach is that the proposed ontology is quite limited and the quality vocabulary is absent. Another drawback is the lack support for QoS priority and mandatory.

A fourth ontology model is the model of Maximilien and Singh. Authors in (Maximilien and Singh, 2004) introduced QoS ontology as part of their approach of the dynamic service selection via an agent framework. Their QoS ontology lets service agents match advertised quality levels for its consumers with specific QoS preferences.

They constructed three ontologies for QoS: upper, middle, and lower. The upper ontology identifies some generic quality concepts and defines the basic concepts related to quality, such as quality relationships. The middle ontology captures the QoS attributes: Availability, Capacity, Economic (cost associated with attributes related to using the service), Robustness, Performance, Stability, Scalability, Security, Interoperability, Integrity and Reliability. This classification of the quality attributes is adapted from (Lee et al., 2003; Ran, 2003; Sabata et al., 1997). The application of the ontology using an example from the insurance and loan domains represents the lower level ontology.

A fifth ontology model is the model of Kassab et al. In order to enable effective communication and to enable integration of QRs' related activities, the authors of (Kassab et al., 2009) developed an ontology-based approach to QRs conceptualization. They identified three views of the QRs ontology: The first view details the QRs' relation with other entities of the software system being developed (e.g. association to functional requirements, to project or to resource). The second view contains the classes and properties intended to structure QRs in terms of interdependent entities e.g. QR type refinement hierarchy; namely, Quality Requirement, Design and Implementation, Economic Constraint, Operating Constraint and Political and Cultural Constraint. The third view captures the measurement process and contains the concepts used to produce measures to measurable QR.

A sixth ontology model is the model of ElicitO. In (Al Balushi et al., 2013, 2007), the authors used an ontology-based approach to build QR quality models with the objective to gather reusable requirements during QR specification. They built a tool called ElicitO to solve requirements reuse problems. ElicitO also helped with the identification of potential conflicts among desired quality attributes and facilitated the process of assessing and negotiating trade-offs towards balancing the weights attached to quality requirements across the project increasing overall stakeholder satisfaction.

The authors adopt the quality requirements' classification of ISO 9126. One drawback is the limited aspects of metrics which needs to be extended to be more generalised.

A seventh ontology model is the model of Kaiya and Saeki. In (Kaiya and Saeki, 2006, 2005), the authors used domain ontology to support software requirements description. They propose some inference rules that need to be applied after an analyst makes mapping between the document specification and the ontology model. Then, the calculation of the rules will identify four main measurements: completeness, unambiguity, correctness and consistency.

Another model is FIPA ("FIPA," 2002), which defines a QoS ontology that is specific to network aspects. This ontology is difficult to extend to other aspects of QoS. This is because it is a low-level ontology.

Another related model is the model of Tondello and Siqueira (Tondello and Siqueira, 2008). The authors developed a model that called QoS-MO ontology. It allows various quality levels offered by a service provider that a requester can choose to satisfy its demands. Its key shortcomings are weak support for QoS value types, QoS units, priority aspects, and mandatory aspects.

The former quality requirements models used different classification of the quality requirements. Table 2-2 shows some of the quality requirements' classifications used in the literature of quality requirements models. It shows that there is no agreed quality requirements' classification in literature.

Table 2-2: Non-Functional Requirements Classifications in Literature

The Ontology Model	QR classification
Dobson et.al. (2005)	Dependability classification in (Avizienis et. al., 2004)
Tran et. al. (2009)	Adapt QoS classification in (Zhou et. al., 2007)
Maximilien and Singh (2004)	Adapt several classifications, e.g. Sabata et.al. (1997), Ran (2004) and Lee (2003)
Kassab et. al. (2009)	Adapt several classifications mainly the classification of ISO/IEC 9126-1
ElicitO model (Al Balushi et. al., 2013)	ISO/IEC 9126-1
Tondello and Siqueira (2008)	Adapted the classification of OMG (2006)

Reviewing the literature, some key concepts and properties that need to be considered for a SoS quality requirements ontology have been identified:

(i) The ontology needs to be identified within several levels in order to be more flexible and extendable. Also, to fulfil the same objective, the ontology can be separated into several sub-ontology models.

(ii) It is a key aspect to have the definition of both the metrics for each quality requirement and the relationships and dependencies between them to enable both the trade-off between the requirements and the grouping of related requirements.

(iii) All main quality requirements need to be identified. This will drive for having more generalised and complete model.

(iv) Other quality requirements related aspects need to be identified, for instance, quality mandatory and priority. Also, any related constraints need to be identified (e.g. operational constraints). This will enhance the process of trading-off.

(v) There are no agreed quality requirements' classification in literature. (see Table 2-2).

(vi) The existing models need to be extended or adjusted in order to support the SoS arrangement. For example, because the constituent systems of the SoS arrangement may be connected to other systems or SoS(s), the interoperability requirement needs to be connected with some new concepts (e.g. External System, Constituent System (CS), Non-Constituent System, SoS level).

In this research, we are planning to identify the main ontology concepts of the QR depending on the definitions made in the aforementioned works in general and in particular the models of Tran et.al. (2009), Al Balushi et. al. (2013) and Dobson et. al. (2005). Also, the ontology model will be extended to involve the capability of supporting the SoS- related aspects. This done by reviewing the literature that is related to SoS. By doing this, the result will be an integrated QR ontology which will combine the strengths of existing quality requirements ontologies and extends the capability of supporting constituent systems and SoS related aspects.

2.9. Policies Models

A Policy is "a set of rules used to guide and determine decisions" (Susanti and Sembiring, 2011). Also, as in (Mansouri et al., 2009) policy provides quality assurance guidelines for the system's activities and processes. So, policy needs to be aligned with quality goals.

There are several policy-related activities and artefacts that are needed to be governed. For example, a vital element to be governed is identifying the roles and responsibilities of the different types of the stakeholders. The roles and responsibilities of individuals and groups are outlined by the policies of the organisation (Musa et al., 2014).

Controlling and management activities of organisations are usually facilitated by a set of policies and internal controls (Van Arnum, 2004). In the case of SoS, many inconsistencies may arise among the policies of monolithic systems. For example, the inconsistencies between policies' conditions, under which a conflict in the policy specifications may occur; specific rules can be defined to detect conflicts (Flegkas et al., 2005). Also, policies are identified and described by several stakeholders with different backgrounds from different organizations and at different times. Thus, several types of policy related conflicts exist, such as authority conflicts and conflicts of resources (Hall-May and Kelly, 2006).

Reviewing the literature, in the field of policy modelling, only few SoS policy models are available. However, in order to develop an ontology of a SoS policy model, several existing policy models were studied to be reused or extended. The purpose was to select the most suitable characteristics and concepts presented by the available models in literature.

A semantically enriched policy developed by Garcia and Toledo (Garcia and Toledo, 2008). They developed an ontology-based policy framework for the purpose of supporting the management of web service business processes. However, their work is specific to security policy. Also, the SoS related concepts are missing. Many of the concepts that are related to Garcia and Toledo's policy model can be used to build the SoS policies ontology model.

Authors in (Phan et al., 2008) developed a framework that supports the specification of quality-oriented policies at the business level and their refinement into policies at the system/service level. They focused on the security aspects as a quality objective along with the corresponding security-related functions. They did not take into consideration the heterogeneity-related issues and SoS-related concepts. Also, they did not use ontologies to develop their model. However, they have linked their model with the security quality requirement which can be used to build the SoS policies model in this research.

Also, Snir et al. (Snir et al., 2003) has developed a quality of service policy model that is not an ontology-based model. The main goal of their work was to model policies that control quality of service behaviour in a way that as closely as possible reflects the way human administrators tend to think about policy and to control quality of service resources. They dealt with conflict aspects between several devices and they focus on the role and condition aspects of the policy. Although their goal is not matching with the current research goal, the aspects of roles and conditions can be adapted in this research.

2.10. Standards Models

A standard is "something made and agreed upon as a guidance" (Susanti and Sembiring, 2011). In addition to the identification of the specifications to ensure that quality requirements are satisfied (INCOSE, 2015), standards should encapsulate a good practice for development (Sommerville, 2010). In the SoS arrangement, there are many standards which may be belong to the SoS level or to one or more CSs.

Sommerville (2010, p658) identifies two main types of standards that may be used in software quality management and need to be governed, process standards and processes standards. The products standards apply to the software product being developed. One example is document standards, such as the structure of requirements documents. The process standards can be followed during software development. Process standards may include definitions of specification, design and validation processes, and a description of the documents that should be written during these processes.

Standards are important facilitators for achieving interoperability. A standard is a technical specification approved by a recognised standardisation body, which is designed to be used consistently, as a rule, a guideline, or a definition across particular communities of interest (Dahmen-Lhuissier, 2019). The aim of a standard is to provide unambiguous specifications for error-free exchange of documents and information for achieving mutual benefits.

The governance board needs to realise which standards are needed to be used to achieve the required quality requirements for governance. Some examples of standards are:

- ISO 9001 ("ISO 9001," 2015) is an example of quality management standard.
- ISO 27001 ("ISO/IEC 27001," 2013) is an example of an information security standard.
- ANSI/GEIA-STD-0009-2008 ("STD-0009," 2008) , which supports a system life cycle approach to reliability engineering.

There are many standards ontology models in the literature. However, it seems that there is a lack of the standards models that are specific to SoS arrangements. They will be reviewed to elicit the main components of them in order to construct a general model that contains the main concepts needed for interoperability purposes in the SoS arrangements without specific details regarding one particular standard. Most of the models in the literature were built to represent a specific standard. So, many models and standards in the literature were reviewed in order to identify the common concepts of the SoS standard model.

The standard needs many resources to be applied, e.g. personal resources to adopt the different roles (Castillo-Barrera et al., 2013; Howarth and Watson, 2011), equipment resources, financial resources to cover the needed cost (Alexander, 2005; *INCOSE*, 2015; "ISO 9001," 2015). To support the deployment of a standard, software tools (Sommerville, 2011, p. 660),or hand tools (*INCOSE*, 2015) may needed. Also, each standard has one or more goals to be achieved ("ISO/IEC 21827," 2008) and one or more conditions need to be satisfied before implementing a standard (Castillo-Barrera et al., 2013; Howarth and Watson, 2011).

2.11. Processes Models

Processes are “collection of interrelated tasks that are enacted to accomplish the purpose of the governance” (Bannerman, 2009). They have to be defined to monitor the use of the standards and check that they have been followed (Sommerville, 2010).

Effective processes and structures that are recognized and understood by quality governance board members will help to successfully govern for quality. This will help with identifying opportunities for quality improvement and identifying potential risks to quality (NHS, 2011).

A process is “a coherent set of actions carried out by a collaborating set of roles to achieve a goal” (Ould, 2005). There are many ways of modelling processes, e.g. Business Processes Modelling (BPM) (Curtis et al., 1992). Business Process Modelling Notation (BPMN) is a recent standard notation proposed by OMG[1] to design business processes. Another way of process modelling is the Role Activity Diagrams (RAD) (Ould, 1995, 2005). Both ways have been used by many researchers to model business processes, e.g. Yousef et al. (Yousef et al., 2009b). However, BPMN is an emerging business process modelling language that appears to be more expressive than RAD and thus extra features in relation to the same process concept in RAD (Yousef et al., 2009b). Also, BPMN is a rich process modelling notation that can be effectively used to model business processes that are understandable by all stakeholders at all levels (Pant and Juric, 2008). BPMN contributes to reducing the gap between business processes and systems (Odeh et al., 2018).

Thus, the main concepts used in this research to model business processes are adapted from the BPMN notations. However, the concepts that are specific to SoS and CSs will be added to the BPMN concepts. Also, it is worth mentioning that the BPMN of the processes of the selected case study are already developed and evaluated in the work of Odeh et al. (Odeh et al., 2018). Also, the process of transforming BPMN concepts to ontological concepts is identified and illustrated by Fan et al. (Fan et al., 2016).

2.12. Observations and Research Gap Analysis

After reviewing the literature, several observations and gaps related to quality governance in SoS in existing research are identified and presented below.

Firstly, the literature showed that there are several SoS quality related issues that still need to be governed. This has been shown by several past and recent studies (Cadavid

[1] Object Management Group: www.omg.org

et al., 2020; “SEBok,” 2015; *DoD*, 2008; Jamshidi, 2008; Maier, 1998; Sledge, 2010). This shows that some gaps are still existing and the research related to SoS quality is an attractive research topic to be investigated.

Secondly, all the aforementioned SoS governance definitions either deal with general governance or define quality governance within a specific context. Hence, it is vital to investigate a more general quality governance definition in the SoS context.

Thirdly, the literature showed that the conflicts resulted from interoperability and heterogeneity are the main causes of SoS issues. According to (Dahmann, 2014; INCOSE, 2015; Cadavid et al., 2020), this gap is still existing and this initiates the need for semantic approach to have consensus on concepts between the CSs in the SoS arrangement. The authors in (Benammar et al., 2015) suggested that this gap can be handled by using ontologies. Ontologies are the most used approach to represent semantic knowledge between heterogeneous domains, and thereby provide robust mechanisms to detect and resolve semantic conflicts (Benammar et al., 2015).

Fourthly, it was found that the most of the current models that are related to SoS quality governance (i.e. quality requirements, policies, processes and standards models) lack the SoS related concepts and lack the process to make them attain to SoS quality governance.

Fifthly, it was recommended in the literature (INCOSE, 2015; DoD, 2008) to use incremental and iterative development methods in the case of developing a SoS artefacts. However, it was observed that several developed governance frameworks in literature e.g. (NHS, 2011) didn’t reflect this recommendation. This initiates the need of using an incremental and iterative development method to be applied when developing a SoS related artefacts.

Sixthly, the literature showed that the many of the current governance frameworks are theoretical frameworks that neither evaluated nor applied to a case study. This gap initiates the need for selecting a research method that allows using a SoS case study to implement and evaluate the SoS quality governance framework.

Seventhly, it was found that in order to adhere to governance and to detect conflicts, there are many necessary interactions between quality governance related models (i.e. policies, processes and standards models) (Benammar et al., 2015; Flegkas et al., 2005; *INCOSE*, 2015; *NHS*, 2011; *DoD*, 2008; Keating, 2015; Van Arnum, 2004). However, the literature showed a lack of a detailed linking process between them. This gap initiates the need to investigate a governance process to check on such interaction to inform the adherence to quality.

Eighthly, the literature focused on the importance of identifying stakeholder's quality requirements at SoS level and monolithic systems level (INCOSE, 2015; DoD, 2008). This initiates the need to govern the quality requirements related aspects at both SoS and CSs levels.

Lastly, there is a lack of SoS research in the health care domain (Cadavid et al., 2020). This initiates the need to select a case study from the health care domain. One possible health care domain is the SoS cancer care (Gorod et al., 2018).

2.13. Summary and Conclusion

The need for SoS has been driven by the complex nature of today's organizations and hence accomplishing their objectives entails the need to develop SoS arrangement based on existing monolithic systems of different domains but delivering new capabilities.

As discussed in this chapter, the heterogeneous SoS arrangement introduces many challenges that are related to quality governance. These challenges can be categorised into several areas. As discussed in the former sections, some of these areas are related to (i) quality requirements, (ii) heterogeneity and interoperability (iii) policies (iv) processes and (v) standards.

After reviewing the literature, several research gaps and limitations related to SoS quality governance still need to be further investigated. The reviewed studies and the existing gaps were assessed based on several aspects, e.g. the semantic representation of the models, the existence of the SoS related concepts, the interactions between the different concepts within the same model and with other models, the exist of detailed information regarding the design and implementation of the models, the application industry of the models and the evaluation methods followed.

Reviewing the literature, several observations and gaps can be summarised as follow:

(i) The literature showed that there are several SoS quality related issues that still need to be governed and several related gaps need to be further investigated.

(ii) A more general quality governance definition in the SoS context is required.

(iii) The interoperability issue of the SoS arrangement initiates the need for semantic representation modelling of the variant governance elements in order to have a universal understanding among the constituent systems. However, ontologies are the most used approach to represent semantic knowledge between heterogeneous domains, and thereby provide robust mechanisms to detect and resolve semantic conflicts (Benammar et al., 2015).

(iv) There is a need for using an incremental and iterative development method to be applied when developing a SoS related artefacts.

(v) The selected research method should allow using a case study to implement and evaluate the developed artefacts.

(vi) The current models of the quality requirements, policies, processes and standards lack the SoS related concepts and lack the process to make them attain to governance.

(vii) The need to investigate a governance process to check on the linking between quality requirements, policies, processes and standards to inform the adherence to quality.

(viii) The need to deploy a case study that is related to health care domain. One possible health care domain is the SoS cancer care domain.

As discussed in chapter 2, the main SoS quality challenges that need to be governed (in Section 2.5) can be categorised into four areas: (i) policies, (ii) quality requirements, (iii) processes and related tasks and (iv) standards.

As concluded from the resulting SoS quality governance areas that need to be governed and from the above observations, in order to have an effective governance, these areas need to interact with each other. Thus, the SoS quality governance can be defined as "the interaction of standards, policies, processes, and quality requirements for software-based systems of systems, using purpose-developed models to inform the adherence to quality in SoS context".

In order to respond to the aforementioned gaps and observations, this research aimed at introducing an ontology-based quality governance framework for SoS, namely the OntoSoS.QM.Gov framework. This framework provides an interaction process between policies, quality requirements, processes and standards models. It is anticipated to help in resolving semantic heterogeneities by providing a consensus on the concepts used and identifying the main conflicts that are expected to emerge from the constituent systems quality governance in relation to monolithic systems policies, standards, quality requirements and processes.

By reaching the end of Chapter 2, the following results have been reported:

(i) Research gap analysis.

(ii) Research hypothesis and RQs.

(iii) Framework proposal that consists of four Ontology models in relation to the areas related to the SoS quality governance (i.e. Policies, Processes, QRs and Standards).

(iv) Initial Framework Design.

(v) Proposed initial interactions between the four models.

By reaching to the aforementioned results, research question 1 has been answered and the gaps that are related to SoS quality governance areas have been identified. Also, initial proposal of the relations between these areas has been reported. Table 2-3 provides an overview of the status towards answering the research questions. The tick symbol (✓) shows that the research question/sub-question has been answered.

Table 2-3: Status towards Answering the Research Questions

RQ	Main RQ and RQs-Concerns (sub-questions)	Status	Notes	Chapter
RQ1	*What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*	✓		2
1.1	Can research gap analysis, by surveying the literature, identify these challenges ?	✓		2
RQ2	*How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*			2, 4,5,6 & 7
2.1	What are the components of the quality governance framework, and what are the initial specifications of each component ?	✓		2
2.2	Is ontology suitable to define all related aspects to quality governance ?			7
2.3	What are the elements/concepts of the ontology? Are there any ontologies that we can reuse ?			4,5,6 & 7
RQ3	*How will the semantically-enriched models of policies, processes, standards and quality requirements interact in the systems of systems context to identify and resolve semantic heterogeneities?*			5,6 & 7
3.1	Can we develop a process to detail the interaction between *policies, processes, standards and quality requirements to identifying and resolve semantic heterogeneities* ?			5,6 & 7
3.2	Are there any limitations of using an ontology-based approach that restrict the interactions between *policies, processes, standards and quality requirements* ?			5,6 & 7
RQ4	*How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*			4,5,6 & 7
4.1	How the OntoSoS.QM.Gov ontology will be assessed ?			7
4.2	Can we validate each ontology model (i.e. each component of the quality governance framework) and then to validate the whole quality governance ontology framework ?			4,5,6 & 7

The research methodology utilised to incrementally develop and evaluate the OntoSoS.QM.Gov framework is presented in Chapter 3.

3. Chapter Three: Research Design

3.1. Introduction

Having reviewed the literature in the fields that are related to semantics and SoS quality governance and its related models, several observations and limitations have been identified. In order to respond to research gaps as discussed in Chapter 2, a rigorous research design is needed that can tackle these gaps incrementally and iteratively. In addition, a real case study is needed where proposed solutions can be applied and evaluated.

This chapter demonstrates the research methods used to develop the resulting research framework. As the nature of the research problem can be classified under the Information Systems (IS) field. The chapter starts with introducing research paradigms within the IS discipline (Section 3.2). The process-based Design Science Research Methodology (DSRM) (Peffers et al., 2007) which has been adopted as a research methodology is introduced and discussed in Sections 3.3 and 3.5. The DSRM process is combined with a case study to understand and reflect on the issues pertaining design, evaluation and implementation of the main research artefacts.

One of the key research artefacts of this study is the OntoSoS.QM.Gov: Ontological Systems of Systems Quality governance framework that is semantically-enriched and incrementally developed, introduced in Section 3.6. Then, an overview of the research evaluation framework is introduced in Section 3.7, followed by a summary and conclusion in Section 3.8.

3.2. Dominant Paradigms in the Information Systems Domain

In the Information Systems (IS) field, there are two dominant research paradigms i.e. design science and natural science (March and Smith, 1995). One basic difference between them is identified by Simon (Simon, 1996, p. 55) who indicated that natural sciences attempt to realise reality, while design sciences create things that assist human to achieve their objectives. The natural science paradigm constructs and justifies theories that predict or explain individuals or organizational behaviour (March & Smith, 1995). Meanwhile, Hevner et al. (2004) explained that the paradigm of design science exceeds the limits of individuals and organizational capabilities and behaviour by creating innovative artefacts. Hevner et al. (2004) introduced an approach to conduct IS research in both paradigms.

In their conceptual framework, Hevner et al. (2004) combined behavioural science and design science paradigms in IS research. This is clarified by the concentration on three research cycles. Firstly, the relevance cycle, which joins the contextual environment of the research with the design science actions. Secondly, the rigor cycle, which joins the design science actions with the knowledge base of scientific foundations and methodologies that feed the research. Thirdly, the design cycle which relates to the actions of building and evaluating the resulting artefacts of the research.

Some artefacts were defined by March & Smith (1995) as the output of the design science to be classified into four artefact types i.e. methods, models, constructs and instantiations. The constructs constitute conceptualisation that describe problems and identify their solutions in a specific domain. The relationships between the constructs expressed using the models. A Method is a set of steps expressed by using algorithms or guidelines to perform a task. The methods are based on a set of constructs and models. When the constructs, models and methods are operated in its environment, they called instantiations. The current research produces several artefacts, e.g. abstract models and processes (March & Smith, 1995).

Hevner et al. (2004) clarified that the rigor of research attained by matching the proper foundations to develop (and/or build) and to adopt the correct methodologies to justify and/or evaluate the product of the research. The selection of research methods is significant as it will lead to the achievement of the research objectives. Thus, the reasoning to choose the research methods detailed by Hevner et al. (2004) and adopted in this research is explained in the next section.

3.3. The Adopted Research Methods

3.3.1. Justification for the Selection of Research Methods

The main reasoning behind the chosen methods in this research, firstly, stem from International Council on Systems Engineering (INCOSE) and Department of Defense (DoD) (*INCOSE*, 2015; *DoD*, 2008) recommendations to use an incremental and iterative development method in the case of developing an SoS artefact. Secondly, the nature of the OntoSoS.QM.Gov framework being developed, as it consists of several interacting models initiates the need for incremental and iterative development of the research artefact to produce several versions till a mature version is developed. Thirdly, as indicated in Chapter 2, the need to use a case study to implement and evaluate the developed SoS related artefacts in attempting to ease the evaluation of the SoS quality governance framework.

Fourthly, the nature of the resulting artefact of this research, e.g. developing an SoS quality governance framework (i.e. an IS artefact) initiates the need to select methods that are used in the IS field. Fifthly, the chosen method needs to allow further development to the resulting artefact for scalability purposes. The scalability of the resulting framework can be done by adding new constituent models (if needed for new case studies) or it can be for improving the design of each component model. The Design Science Research Methodology (DSRM) (Hevner et al., 2004; Hevner, 2007; March & Smith, 1995) satisfies the aforementioned points.

The definition of DSRM was firstly introduced by March & Smith (1995), which indicated that it is a methodology used to generate things that support fulfilling human aims. Also, Hevner et al. (2004) indicated that the purpose of DSRM is to improve the state of current practice in the IS field. This, supports achieving the aim of this research, which is to create an IS artefact, e.g. quality governance framework and its related processes. After the development process of the artefacts of this research, an evaluation process was carried out to assess the effectiveness of each of these artefacts.

Thus, stemming from the above and the pragmatic nature of the design science in the continuous interaction between the actual application domain environment and design science, the DSRM methodology (Hevner et al., 2004; Hevner, 2007) has been adopted, as one of its capabilities to tackle issues concerning the development of an IS artefact. Finally, DSRM is also applicable to deploy in combination with other research methodos such as case study (Nunamaker et al., 1990). The following sections elaborate more on the use of DSRM and case study approach.

3.3.2. Design Science Research Methodology

Simon (1996) identified the need for various design sciences with the perspective of design as a problem-solving activity primarily to create an innovative technological product. There are many Design Science Research Methodology (DSRM) processes that exist in the literature (Hevner, 2007; Hevner et al., 2004; March and Smith, 1995; March and Storey, 2008; Nunamaker and Chen, 1990; Peffers et al., 2007, 2006; Vaishnavi and Kuechler, 2015, 2007). All the former processes agreed that DSRM is a problem-solving process, and that both building and evaluating artefacts are the vital parts of the DSRM framework.

In this research, the DSRM process steps during the design science are adopted from (Peffers et al., 2007). (This model consists of six phases as illustrated in Figure 3-1).

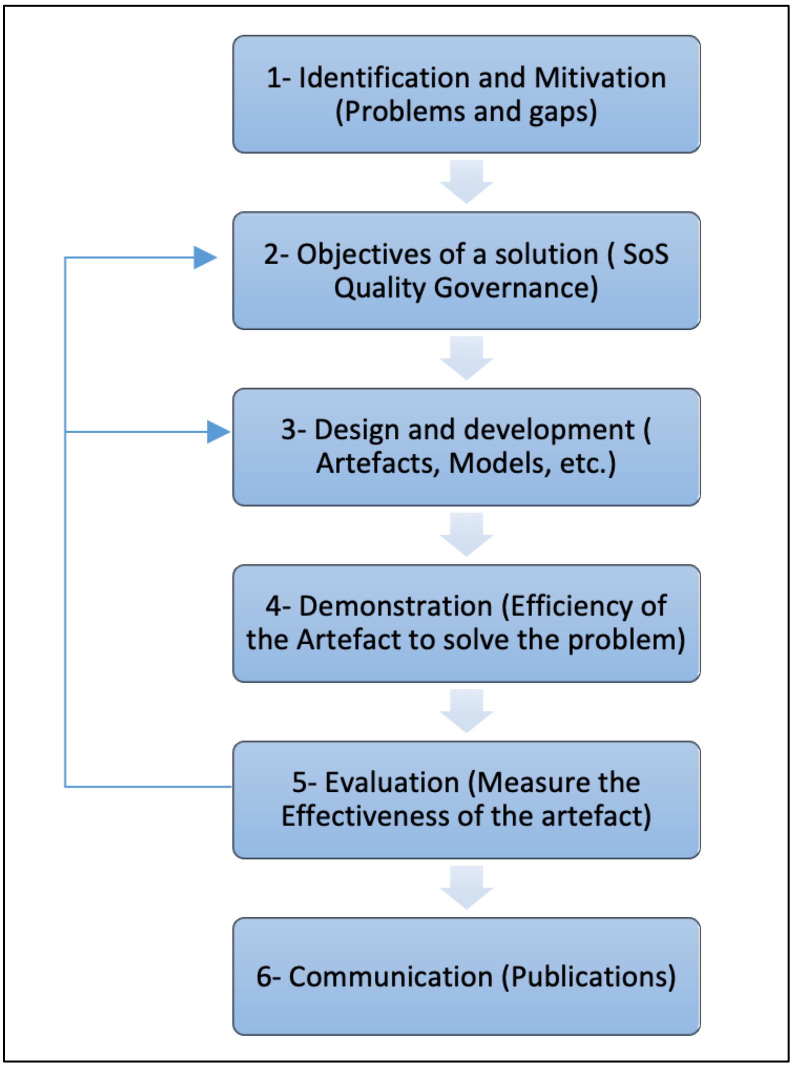

1- Identification and Mitivation (Problems and gaps)
2- Objectives of a solution (SoS Quality Governance)
3- Design and development (Artefacts, Models, etc.)
4- Demonstration (Efficiency of the Artefact to solve the problem)
5- Evaluation (Measure the Effectiveness of the artefact)
6- Communication (Publications)

Figure 3-1: Design Science Research Methodology (DSRM) phases (Adapted from [Hevner, 2007; Hevner et al., 2004; Peffers et al., 2006])

Peffers et al. (2006) introduced a simple DSRM framework (as seen in Figure 3-1) that consists of six phases (1) problem identification and motivation, (2) solution suggestions, (3) design and development, (4) demonstration, (5) evaluation and (6) communication. Phases 2- 6 can be reiterated based on the feedback of the evaluation of the artefact. The brief descriptions of each phase are as follows (Peffers et al., 2007):

1. Problem identification and motivation – this phase defines the research problem and the motivation derived from reviewing the literature;
2. Objectives of a solution - this phase identifies some possible solutions and determines the aims and criteria of the identified solution;
3. Design - the design of the solution will be built in this step, this can involve methods, constructs, models, and instantiations;
4. Demonstrate – the application of the solution to prove that the design solution meets its objectives. This phase can be conducted using a case-study;

5. Evaluate - the effectiveness of the artefact is checked in this step. Here, the results from demonstration phase will be evaluated using an evaluation framework with the identification of metrics to assess the effectiveness of the solution; and
6. Communicate - the findings of the research can be communicated to the audience via publications, thesis, etc.

DSRM states that there are two key features of design science artefacts: relevance and novelty. First, to be relevant, an artefact needs to solve a problem. Second, design science research should address either an unsolved problem in an innovative way or a solved problem in a more efficient or effective method (Hevner et al., 2004). The DSRM research activities showed in Figure 3-1 have been conducted in this research and introduced in Section 3.5.

3.3.3. The Mixed Research Approach

Nunamaker et al. (1990) stated that the DSRM approach can be combined with other research approaches e.g. case study and survey. Lately, a number of IS research has adopted the DSRM that is mixed with other methodologies. For example, the authors of (Ahmad, 2015; Aljawawdeh, 2019) deployed the DSRM with a case study to develop and evaluate effective research frameworks. The integration of the research methods was concluded from the research gap analysis conducted in Chapter 2 and was justified to achieve the current research objectives.

It was concluded in Chapter 2 that it is essential for this research to adopt a case study considering that there are two phenomena that require an engagement of a case study. These are (i) the refinement of the design specifications for SoS quality governance ontological models, (ii) the effectiveness of the quality governance framework. This will help to answer the research questions RQ 2- RQ 4 (see research questions in Table 1-1). Thus, the case study helped the researcher to accomplish the objectives of this research, e.g. to demonstrate and evaluate the ontology-based models and the process used to identify and to support resolving semantic heterogeneities. This is linked with another objective, which is to assess the effectiveness of the framework. The recommendations and the feedback from the domain experts/ interviewees in the case study were used as guidelines to evolve better versions of the developed research framework. Also, the documents (i.e. policies, plans and strategic documents) provided by the case study supported the design and development process.

Another research method that has been used in this research was the literature review method, which has been used in several phases of DSRM, e.g. the problem identification and design phases. Firstly, the literature review was used to identify the main quality governance challenges, which led to answer the first research question (RQ1). Also, literature review was used to articulate the research gap analysis which led to formulating the definition of the SoS quality governance, beside a set of research questions and hence forms the research hypothesis. Also, literature review informed the design of the ontological governance models.

Based on the above description and considering the need for conducting a literature review and to use a case study approach beside the iterative nature of developing the governance framework (as indicated in Chapter 2), the research design has been justified driven by a DSRM process with a sufficient and representative case study to inform a quality governance approach based on QRs, policies, processes and standards, semantically enriched model in SoS context.

3.4. The Research Case Study

3.4.1. Justification for the Selection of the Case Study and Selection Criteria

In information systems, a case study tests a phenomenon in its natural setting, using various means of collecting information from individuals, groups, or organizations (Benbasat et al., 1987). Another benefit of using case studies is to enable the researcher to intensely study the systems in the actual environment of the study object (Valverde and Toleman, 2007). There are many sources of evidence in case studies such as documents, interviews, observation, and physical artefacts (Yin, 2003).

Hevner et al. (2004) recommended using a case study as a strategy to evaluate IT artefacts. Also, according to Hevner et al (2004), if a case study that satisfies the requirements for testing all components of the developed research artefact is available, then the case study approach is the most effective way of demonstration and assessing the effectiveness of the resulting artefact. In addition, one useful approach to test a research hypothesis is by using a case study to help answering the research questions (National Academies of Sciences, 2009). So, in this research, a case study is an appropriate approach that is used to drive the DSRM research design process.

A case study needs to be suitable and sufficient. Thus, in order to be chosen, the following criteria need to be available in the case study:

1. The case study needs to be sufficient in order to support the design, the demonstration and the evaluation phases of the main research artefact. Hence, as mentioned above, it needs to satisfy the requirements for testing all components of the developed research artefacts (Hevner et al., 2004).
2. As the developed artefacts are in the SoS context, then the SoS arrangement context need to be available in the case study. The case study needs to satisfy at least, the two main principal distinguishing characteristics for applying the term SoS, which are that the component systems can operate independently and can be managed independently (Maier, 1998). So, each system is capable of achieving its own goals in the absence of the other systems. Also, the case study needs to be from a SoS sector that is proven to have issues in the SoS quality management, e.g. health-related sector. This will facilitate assessing the effectiveness of the resulting artefacts. Also, this will help the researchers to trace any shortcomings or any unexpected behaviour of the research artefacts. So, any positive impact of this research on the case study can be noticed.
3. As recommended by (Gobo, 2004; Tsang, 2014), the case study needs to be a representative and a comprehensive case study in its domain as a step towards generalization.
4. As a step towards generalization of the outcomes of the current research, interviews need to be conducted with experts in the domain (Flick, 2018). So, in order to facilitate the evaluation process, it would be a vital value for this research if domain experts are available in the case study.
5. As recommended by (Flick, 2018; National Academies of Sciences, 2009), the case study needs to support proving the research hypothesis and support answering the research questions of the current research (will be detailed in sections 3.4.2 and 3.7). This means, it needs to be information-rich instances that are relevant to the research question (Flick, 2018; Patton, 2015). Also, the selected case study needs be able to provide an appropriate-sized case study from the chosen domain.
6. To enable the quality governance framework to be fully represented, implemented and tested, the case study needs to provide the required information to facilitate the implementation of all governance-related areas (i.e. standards, policies, processes and quality requirements). The case study needs to be able to provide different kinds of documents that support the demonstration process of the component models of the research framework. For example: policy documents, standards chapters/ manuals, quality manuals, etc.

7. It would be a preference if the institution of the case study supports the research and researchers.

As it fulfils the former requirements, an SoS cancer care case study at King Hussein Cancer Centre (KHCC) in Jordan is chosen in this research. Next section provides an overview of the selected case study from cancer case domain.

3.4.2. The Selected King Hussein Cancer Centre Case Study

The integrated health care related systems is an example of an SoS arrangement (Gorod et al., 2014) that has quality management related issues (Bianchi et al., 2015). One example of a health care integrated systems is cancer care. In general, cancer care is a complex process (Palmieri et al., 2013). To satisfy the need for a high-quality care for cancer patients, many constituent systems are connected to form a cancer care SoS arrangement. One of the main benefits of SoS is delivering unique capabilities that cannot be totally fulfilled by any individual monolithic system (Chin et al., 2013). So, in order to accomplish cancer care-related services, there is a need to develop a SoS arrangement to enable cooperation between existing monolithic cancer care systems (Qaddoumi et al., 2018b).

Cancer care treatment related systems is an example of a SoS arrangement (Gorod et al., 2018). The KHCC case study from the cancer care domain is an SoS arrangement as it consists of several systems that are connected to form an SoS cancer care arrangement. This provides the SoS research context which is a basic requirement that is needed to demonstrate and evaluate the quality governance framework. Furthermore, "KHCC is the only healthcare institution in the Arab world and the sixth in the world to receive disease-specific accreditation from the Joint Commission International (JCI) for its oncology program. The centre is also recognized by a spectrum of international, regional, and national accreditation bodies including the College of American Pathologists (CAP), the Healthcare Accreditation Council (HCAC) and others" (Mansour, 2016). This means that the quality governance framework resulting from this research and demonstrated through using the policies and quality- related documents provided by the KHCC centre are paving the ground for the resulting governance framework to suggest generating it to other cancer care institutions in a similar context to KHCC.

Also, KHCC management governance relies on interacted quality standards, processes and clinical practical procedures. Thus, KHCC provides a suitable representative case study in relation to the four research governance related models (models will be introduced in Section 3.5.1 and Section 3.6) and sufficient to inform the governance interaction process within the OntoSoS.QM.Gov framework. A case study from KHCC is anticipated to enrich the modelling of both the main quality governance framework components and the interactions between them through several DSRM iterations (the iterations will be detailed in the next section and next chapters). Also, this comprehensive representation will enable the researcher to trace any limitations or unexpected behaviour of the OntoSoS.QM.Gov framework which answers the RQs along with the test cases for the four related governance areas (i.e. QRs, standards, processes and policies) (as discussed in Section 3.7).

For the scope of this research, the selected KHCC case study needs to deploy only test cases from a SoS arrangement that includes related monolithic systems. Thus, in this research, cancer care processes, standards and policies that are related to achieving the quality requirements that linked to cancer care services in providing cellular therapeutic and genomics-based therapy to cancer patients. This has an interaction between several monolithic systems that are related to KHCC Admission, Discharge and Transfer (ADT), and Cell Therapy and Applied Genomics (CTAG) -related monolithic systems.

These monolithic systems are: Flow Cytometry (FC), Molecular Diagnostics Immunogenetics (MDI), Blood and Marrow Transplant (BMT), Cytogenetics, Pharmacy System, Laboratory System, Patient Management System, Surgical Management Systems, Treatment Systems (radiotherapy treatment and chemotherapy treatment), Financial Systems, Medical Recording System, Information Exchange Systems, etc. All these autonomous systems are involved in conducting cancer care tasks. These constituent systems relate to stakeholders with different cancer care quality perspectives. Each system has its policies, standards, processes and quality requirements to be satisfied (Qaddoumi et al., 2017). However, for the scope of this research, only few monolithic systems have been identified for the CTAG process operation.

ADT and CTAG-related systems follow several standards. Examples of these standards are the (HCAC; 2012, JCI; 2014, JCI; 2017, CAP; 2016). Their main aim is to ensure adherence to safety and quality of cancer care services in SoS context. Also, they have policies that control the overall delivery of cancer care, as well as the policies that are dedicated to certain services for certain constituent systems. With regards to the several activities, written policies, quality requirements and the operational constraints for both cancer care level (i.e. SoS level) and the constituent systems level, KHCC provided the research team with various documents that consist of the detailed policies, plans and strategic documents.

For the CTAG and ADT functional area, domain experts for quality assurance and management, operational processes and governance have been available to support the design and evaluation- related activities in order to validate the research governance models and their interaction process.

It may worth mentioning that KHCC was selected to provide case studies for several earlier research projects (Aburub et al., 2007; Aljawawdeh, 2019; Yousef et al., 2009b). More details regarding the sufficiency of the selected test cases are provided in section 3.7.

3.5. The Research Process

3.5.1. The Incremental and Iterative Development and Evaluation of the Research Framework

The main research artefact (i.e. OntoSoS.QM.Gov framework) has been developed incrementally through iterations over DSRM process phases 2 - 5. "The framework is developed as a series of versions (increments), with each version adding functionality to the previous version" (Sommerville, 2010).

Figure 3-2, illustrates the iteration of DSRM phases of specification, development, and evaluation. In this research, the four main models of quality governance framework are: (1) Policy (OntoSoS.Policy), (2) Quality Requirements (OntoSoS.QR), (3) Processes (OntoSoS.Process), and (4) Standards (OntoSoS.Std). Each of these models is handled incrementally in one single iteration of the DSRM method. This means, the first iteration is to design and develop the QR model with the evaluation of this model and its improvement to inform the second iteration, where the policy model is developed, evaluated, improved and integrated with the previous developed model. This, in turn, feeds into the third iteration where the processes model is developed, evaluated, improved and integrated with the previous developed models. And hence, by the end of the third iteration, the QR, policy and processes improvements will have taken place to inform the fourth iteration where the standards model is developed, evaluated, improved and integrated with the previous developed models. Thus, when the fourth iteration is completed, the interaction between the four governance models in this research framework will have been developed, assessed, and improved reflecting an incremental design of the OntoSoS.QM.Gov framework, following the adopted DSRM process. The next section will provide an overview of the activities in each phase.

As the intended framework is defined to contain four models, at least, four main DSRM iterations are needed. Firstly, as the main aspect of this research is quality, the QR model was developed in the first iteration. Secondly, as the governance aspects require controlling and management activities of organisations, which are usually facilitated by a set of policies and internal controls (Van Arnum, 2004), the policy model was developed in the second DSRM iteration. Also, as in (Mansouri et al., 2009) policy provides quality assurance guidelines for the system's activities and processes. So, policy needs to be aligned with quality goals. Thirdly, two options are available here, either to develop the processes model or to develop the standard model. However, as the processes have to be defined to monitor the use of the standards and check that they have been followed (Sommerville, 2010) and they need to be conformed with the procedures identified in the policies, the processes model was developed in the third iteration. Fourthly, the standard model was developed in the fourth DSRM iteration.

In order to confirm the order chosen for developing each model in each DSRM iteration, a semi-structured interview was conducted with some domain experts to check if they agree or not to the chosen track. This interview ended with supporting the research team choices.

Table 3-1 shows the details regarding the four increments developed until the OntoSoS.QM.Gov was incrementally evolved. In each DSRM iteration, a new model was combined to the predefined DSRM increment in order to add new capabilities to the previously developed increment. After the validation of each increment, a new version of the increment was produced and baselined. The resulting increment version consisted of an updated version of all components. This was so, because the new interaction may require some new concepts or relations to be added or deleted. However, in most cases only minor amendments were applied. More details are described in the following sections and chapters.

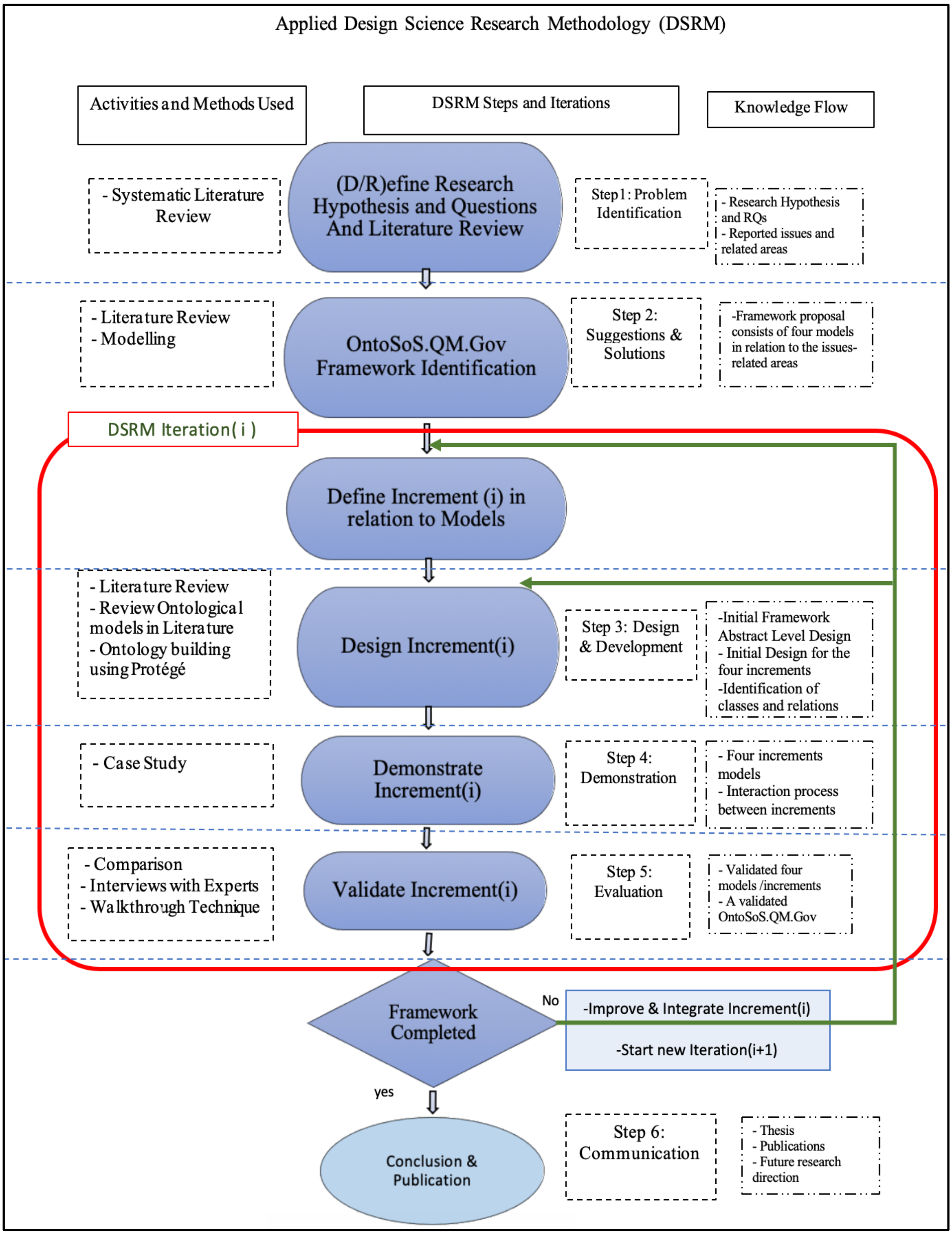

Applied Design Science Research Methodology (DSRM)
Activities and Methods Used
DSRM Steps and Iterations
Knowledge Flow
- Systematic Literature Review
(D/R)efine Research Hypothesis and Questions And Literature Review
Step1: Problem Identification
- Research Hypothesis and RQs
- Reported issues and related areas
- Literature Review
- Modelling
OntoSoS.QM.Gov Framework Identification
Step 2: Suggestions & Solutions
-Framework proposal consists of four models in relation to the issues-related areas
DSRM Iteration(i)
Define Increment (i) in relation to Models
- Literature Review
- Review Ontological models in Literature
- Ontology building using Protégé
Design Increment(i)
Step 3: Design & Development
-Initial Framework Abstract Level Design
- Initial Design for the four increments
-Identification of classes and relations
- Case Study
Demonstrate Increment(i)
Step 4: Demonstration
- Four increments models
- Interaction process between increments
- Comparison
- Interviews with Experts
- Walkthrough Technique
Validate Increment(i)
Step 5: Evaluation
- Validated four models /increments
- A validated OntoSoS.QM.Gov
Framework Completed
No
-Improve & Integrate Increment(i)
-Start new Iteration(i+1)
yes
Conclusion & Publication
Step 6: Communication
- Thesis
- Publications
- Future research direction

Figure 3-2: The Research Process and Associated Methods

Table 3-1: Design Science Research Methodology Iterations

DSRM Iteration	Input	Output	Notes
	(similar colours represent similar increments as the output increment of each iteration will be considered as an input to the next iteration)		
First Iteration	Version 1(V1) of Increment 1 = V1 of Quality Requirements (QR) model	V2 of Increment 1 = V2 of QR model	The details are provided in Chapter 4
Second Iteration	V1 of Increment 2 = V1 of Policy model + V2 of Increment 1	V2 of Increment 2 = V2 of Policy model + V3 of QR model	The details are provided in Chapter 5
Third Iteration	V1 of Increment 3 = V1 of Processes model + V2 of Increment 2	V2 of Increment 3 = V2 of Processes model + V3 of Policy model + V4 of QR model	The details are provided in Chapter 6
Fourth iteration	V1 of Increment 4 = V1 of Standard model + V2 of Increment 3	V2 of Increment 4 = V2 of Standard model + V3 of Processes model + V4 of Policy model + V5 of QR model	Increment 4= OntoSoS.QM.Gov Framework. The details are provided in Chapter 7.

3.5.2. DSRM Process Phases and Iterations

DSRM consists of six main phases. An overview of the main activities conducted in each phase are briefly described in the following sub-sections.

3.5.2.1. Phase 1: Problem Definition

The first phase of this research begins with identifying the problems that affect the quality governance for an SoS arrangement. At this stage, literature review was the main activity that has been carried out (see Figure 3-2). Here, a literature survey to the areas related to this study has been conducted in this phase. The aim of this phase was to identify the main gaps and challenges in the literature. This also resulted in affirming the definition of quality governance, research objectives, the research questions and the formulation of the research hypothesis. (The details are provided in Chapters 1 and 2).

3.5.2.2. Phase 2: Objectives of a Solution

As in Figure 3-2, an initial OntoSoS.QM.Gov framework was identified in the second phase (i.e. suggestions and solutions) of this research. This identification involves an initial identification of four ontological models, (1) policy model (OntoSoS.Policy), (2) quality requirement model (OntoSoS.QR), (3) processes model (OntoSoS.Process) and (4) standards model (OntoSoS.Std).

Chapter 2 provides details on the relevance (Hevner et al., 2004) to this research. This is shown by identifying the needs of and the gaps in the literature related to SoS quality governance. After conducting the research gap analysis in the first phase, the problem has been identified and the approach of how to solve this problem has been suggested (phase 2). Here, the use of ontologies for semantic modelling was suggested. This semantic modelling captures the key features of the four related areas (i.e. standards, policies, processes and quality requirements). Also, while conducting the research gap analysis, several frameworks in the literature have been reviewed to identify the state-of –art of governance and related areas frameworks. This shows the *rigour* (Hevner et al., 2004) for the design phase of the research artefact. Next, the general and initial objectives that need to be available in the developed solution, identified after phase 2 of DSRM, are summarised.

The following are the initial objectives and guidelines for the OntoSoS.QM.Gov framework:

1. Adherence to its components: OntoSoS.QM.Gov needs to be created in relation to the four ontological models (i.e. OntoSoS.QR, OntoSoS.Process, OntoSoS.Std and OntoSoS.Policy). All the former models will be semantically integrated together to formulate the required quality governance. Also, OntoSoS.QM.Gov framework needs to adhere to these four areas, so it needs to capture all key features of these areas which affect the quality governance of SoS arrangements. However, more details regarding these models will be provided in the following chapters (Chapters 4, 5, 6 and 7).
2. Attempt to create a generic quality governance framework in relation to processes, standards, quality requirements and policies. However, this will be supported by selecting a suitable and sufficient case study in the selected domain (as discussed in Section 3.4.1) .
3. Semantically- enriched framework: to have a consensus on the concepts between all CSs, the main artefact of this research (i.e. OntoSoS.QM.Gov) and its related components (i.e. the four models) are developed using semantic approach. One approach is by using ontology method to conceptualise the governance aspects and their relations to provide common governance- related concepts.
4. Providing a holistic approach to SoS and the relationships among its constituent systems by proposing a holistic approach to governance framework in attempting

to link the quality governance-related areas of standards, policies, QRs and processes in SoS context.

5. Effectiveness of the artefact: the framework needs to be evaluated to determine its effectiveness incrementally as per the quality governance models in relation to the KHCC selected case study.

Next, a brief of the main four iterations are overviewed in the next sub-sections. Then a description of each iteration will be detailed in Chapters 4 to 7.

3.5.2.3. First Iteration of the Design, Development, Demonstration and Evaluation steps of the DSRM: Increment 1

The main objective of this iteration is to design, demonstrate and evaluate the quality requirements model (i.e. OntoSoS.QR). The semantic representation of this model is generated using ontology. Here, the conformance of the developed QR model to the international IEEE standard has been taken into consideration while developing the QR model. Semi-structure interviews and walkthrough technique were conducted for evaluation purposes with the domain experts from the field of quality in the case study.

The OntoSoS.QR model is the first increment of the OntoSoS.QM.Gov (will be detailed in Chapter 4). Any required amendments are applied, and a second version of Increment 1 was generated and considered as an input to the second iteration.

3.5.2.4. Second Iteration of the Design, Development, Demonstration and Evaluation steps of the DSRM: Increment 2

Here, the resulting increment of the previous iteration is combined with policy model to form the first version of increment 2 (as discussed in Chapter 5). The main objectives of this iteration are to design, demonstrate and evaluate the policy model (i.e. OntoSoS.Policy), determine possible interactions between the QRs and the policy models, generate the semantic representation of the combined models, and hence forming the second increment of the OntoSoS.QM.Gov. Domain experts from the case study validated the second increment in two phases. First, validating the policy model and then validating the interaction between policy model and the previous developed QR model in the first DSRM process increment. Any required amendments that resulted from the interaction between both models are reviewed and applied. Semi-structure interviews and walkthrough techniques were conducted for evaluation purposes. Then, a second version of Increment 2 was generated and considered as an input to the third iteration.

3.5.2.5. Third Iteration of the Design, Development, Demonstration and Evaluation steps of the DSRM: Increment 3

The resulting increment of the previous iteration is combined with process model to form the first version of Increment 3 (as discussed in Chapter 6). The main objectives of this iteration are to design, demonstrate, evaluate the processes model (i.e. OntoSoS.Process), determine possible interactions between the process model and both QR and policy models, and to generate the semantic representation of the combined models to form the third increment of the OntoSoS.QM.Gov framework.

Then, some domain experts from the case study validated the third increment. Two phases of evaluation were conducted at this stage: validating the process model and validating the interaction between process model and the previous developed increments (i.e. the QR model and policies model). Any required amendments that resulting from the interaction between the three models are reviewed and applied. Semi-structure interviews and walkthrough techniques were conducted for evaluation purposes. Then, a second version of Increment 3 was generated and considered as an input to the fourth iteration.

3.5.2.6. Fourth Iteration of the Design, Development, Demonstration and Evaluation steps of the DSRM: Increment 4

Here, the resulting increment of the previous iteration is combined with the standards model to form the first version of Increment 4 (as discussed in Chapter 7). The main objectives of this iteration are to design, demonstrate and evaluate the standard model (i.e. OntoSoS.Std), determine and validate possible interactions between the standards model, processes model, QR model and policies model, and to generate the semantic representation of the combined models to form the fourth increment of the OntoSoS.QM.Gov.

Then, domain experts from the case study validated the fourth increment. Two phases of evaluation were conducted at this stage: to validate the standard model and to validate the interaction between standard model and the previous developed increment (i.e. the models of QRs, processes and policy). Any required amendments resulting from the interaction between the four models are reviewed and applied. Semi-structure interviews and walkthrough techniques were conducted for evaluation purposes. Then, a second version of increment 4 was generated. At the end of the fourth iteration an intermediate version of the OntoSoS.QM.Gov was developed.

3.5.2.7. Phase 6: Communication

The evaluated artefacts of this research study were communicated to the practicing professionals and research communities (i.e. scientific publications and seminars). A final version of the thesis has been written. Also, the publications resulting from this study are listed in the thesis. Also, future work driven to enhance the resulting OntoSoS.QM.Gov framework of this study are communicated too (as discussed in Chapter 8).

3.6. Introducing the OntoSoS.QM.Gov Framework

3.6.1. The Interactions between Models to Formulate Quality Governance

This research is an attempt to bridge the gaps in SoS quality governance components, and the OntoSoS.QM.Gov is the artefact resulting of this gap bridges. This research attempts to develop a generalic framework for quality governance in the SoS context. This framework consists of four main models: standards, policy, quality requirements and processes. These models are linked to each other to formulate the quality governance framework in order to satisfy the needed quality requirements at both the SoS level and the monolithic systems level. At SoS level, the proposed framework will serve as a Global Quality Governance (GQGov) for the arrangement of the SoS with a holistic approach to the SoS arrangement and the relationships among its constituent systems with emphasis on the linking of Local Quality Governance (LQGov) for the constituent systems. The LQGov of each constituent system consists of the former four models (i.e. OntoSoS.Std, OntoSoS.Policy, OntoSoS.QR and OntoSoS.Process).

Thus, at the SoS level, the SoS quality governance and its formalised ontological components are briefly specified in Equation (1) where i= 1 to n, and n is the number of monolithic constituent systems (Qaddoumi et al., 2018b).

$$\boxed{GQGov = \bigcup_{i=1}^{n} (LQGov_i)} \qquad (1)$$

GQGov = LQGov$_{(1)}$+ LQGov$_{(2)}$+ …+ LQGov$_{(n)}$, where *n* = number of the constituent systems in the SoS arrangement. The holistic view of the GQGov ontology of SoS comprised of *n* number of constituent systems, is depicted in Figure 3-3.

For each constituent system, the LQGov consists of all policies, standards, processes and QRs in each CS as in Equation (2).

$$LQGov = \sum_{i=1}^{i=po} Policies + \sum_{i=1}^{i=s} Standards + \sum_{i=1}^{i=pr} Processes + \sum_{i=1}^{i=qr} Quality\ Requirements \qquad (2)$$

In Equation (2), LQGov = (policy$_{(1)}$+ policy$_{(2)}$+…+ policy$_{(po)}$)+ (standard$_{(1)}$ + standard$_{(2)}$+…+ standard$_{(s)}$)+ (process$_{(1)}$+ process$_{(2)}$+…+ Process$_{(pr)}$)+ (QR$_{(1)}$+ QR$_{(2)}$+…+ QR$_{(qr)}$), where *po* = number of policies in the constituent system, *s* = number of standards in the constituent system, *pr* = number of processes in the constituent system, *qr* = number of quality requirements in the constituent system.

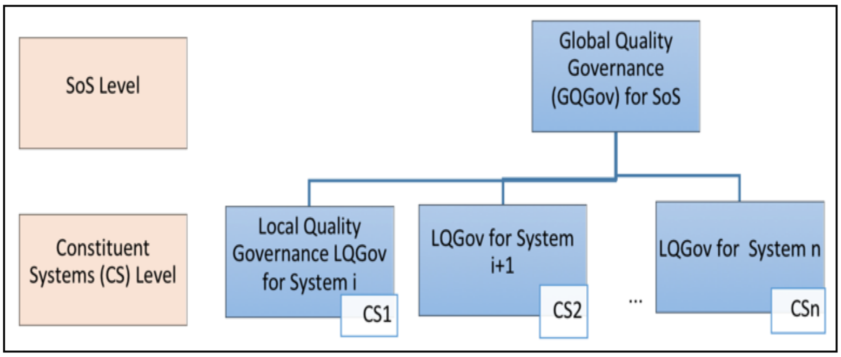

SoS Level
Global Quality Governance (GQGov) for SoS
Constituent Systems (CS) Level
Local Quality Governance LQGov for System i
CS1
LQGov for System i+1
CS2
...
LQGov for System n
CSn

Figure 3-3: Global Quality Governance for SoS with a Holistic View

The incremental development approach was used in order to incrementally implement and illustrate the applicability of the OntoSoS.QM.Gov framework using a linking process to link the four ontological models forming the research quality governance.

Identifying possible heterogeneities that are related to a specific QR is proposed in Algorithm 3-1 as an example to demonstration. One benefit resulting from this interaction, as in Algorithm 3-1, if we want to check if a QR is satisfied. Firstly, for each CS in the SoS arrangement, all policies that have the purpose of satisfying the entered QR will be listed. Then, several comparisons between each policy, and all related processes and standards are conducted to identify any conflicts between roles and values (if any). After all conflicts are listed along with all related systems that are engaged in delivering the entered QR, then human can be involved to prioritise and compromise between any alternatives.

Algorithm 3-1: General Algorithm to Identify Possible Quality Requirement related Heterogeneities

```
  Input:
      $QR_i$ = The Quality Requirement that we need to check governance for.
  Output:
  /* similar output can be seen later in Algorithms 7-1, …, 7-4*/
CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies $Po_j$ that are related to $QR_i$ in all constituent systems CS , where 0 < i <= n, n=number of CSs; 0<j<=p, p=number of all policies in each CS.
Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=c, c=number of all constraints in each CS; and  0<x<=s, c=number of all services in each CS.
CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, $CS_n$. $QR_q.Properties_r$},  where 0 < i <=n, n= number of CSs; 0<j<=q, q=number of all QRs in each CS; 0<k<=r, r=number of all Properties in each QR.
CS.Process_Set= { $CS_1.Pr_1$, $CS_1.Pr_2$, …, $CS_2.Pr_1$, …, $CS_i.Pr_j$, …, $CS_n.Pr_p$}, the set of all processes $Pr_j$ that are related  to each policy in CS.Policy_Set in all constituent systems CS , where 0 < i <= n, n=number of CSs, 0<j<=p, p=number of all processes in each CS.
Conflicts_Policy_Process_Set={$CS_1.Role_1$, $CS_1.Task_1$, $CS_2.Documentation_1$, $CS_1.Action_1$, $CS_1.DataObject_1$, …, $CS_i$. $Role_j$, $CS_i.Task_k$, $CS_i.Action_l$, $CS_i.DataObject_m$, $CS_i$. $Documentation_n$, …,$CS_n$. $Role_r$, $CS_i.Task_t$, $CS_i.Action_a$, $CS_i.DataObject_o$, $CS_i$. $Documentation_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=t, t=number of all tasks in each CS; 0<l<=a, a=number of all actions in each CS; 0<m<=o, o=number of all data objects in each CS; 0<n<=d, d=number of all documents in each CS;
Standards set St_Set = {$CS_1.St_1$, $CS_1.St_2$, …, $CS_i.St_j$, …, $CS_n.St_s$), where 0 < i <= n, n=number of CSs; 0<j<=s, s=number of all standards in each CS.
Conflicts_Standard_Process_Set= {$CS_1.Role_1$, $CS_1.Documentation_1$, $CS_2.Condition_1$, $CS_1.Tool_1$,…, $CS_i$. $Role_j$, $CS_i$. $Documentation_k$, $CS_i$. $Condition_l$, $CS_i.Tool_m$ , …,$CS_i$. $Role_r$, $CS_i$. $Documentation_d$, $CS_i$. $Condition_c$, $CS_i.Tool_t$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts with their values, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=d, d=number of all documents in each CS; 0<l<=c, c=number of all conditions in each CS; 0<m<=t, t=number of all tools in each CS.
  Begin:
    For each Constituent System $CS_i$ do, (where 1<=i<=n, n= number of CSs in the SoS)
        /*Steps 1 and 2 will be detailed later in Chapters 5 and 7*/
        Step 1:
        Find the related policies that have the purpose of satisfying the related $QR_i$.
        Step 2:
        For each policy identified in Step1, identify any conflicts by comparing between the concepts of the policy model and the concepts of the related standards to check if the policy is conforming with (adhere to) those standards. Then, repeat for all CS(s).
        End for
        /*Steps 3 will be detailed later in Chapter 6*/
        Step 3:
        For each policy identified in Step 1, identify any conflicts by comparing between the concepts of the policy model and the concepts of the related processes to check if the processes are satisfying (adhere to) those policies. Then, repeat for all CS(s).
        End for
        /* Step 4 to list the output*/
        Step 4:
        List all heterogeneities/conflicts resulted from the comparisons and all constituent systems, departments and roles that are engaged in delivering $QR_i$.
    End for
  Human involvement: Prioritise the identified conflicts and compromise between alternatives.
End.
```

The abstract model of the quality governance framework is depicted in Figure 3-4. More details regarding the relations between the framework components are described in the following chapters.

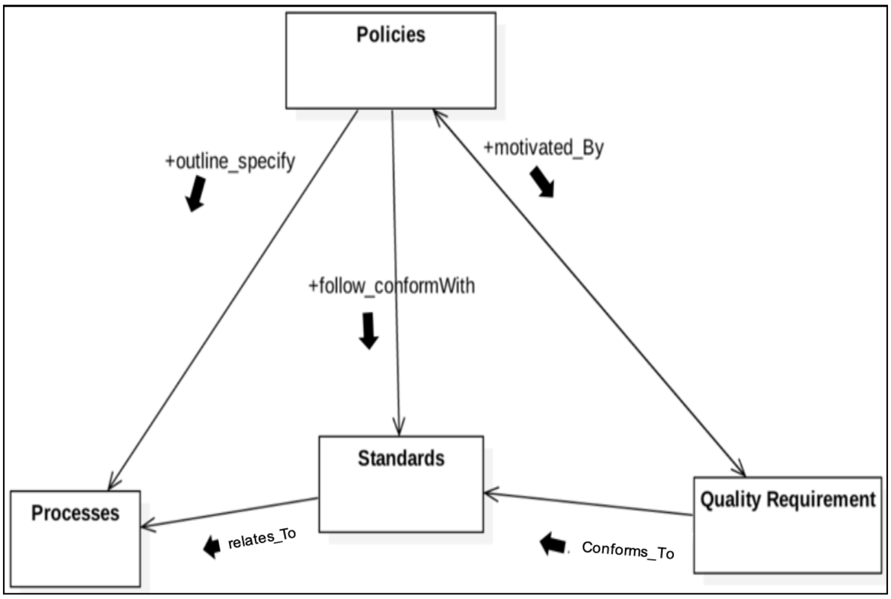


Figure 3-4: Abstract Level of Quality Governance

3.6.2. The Architectural/Layered Framework of OntoSoS.QM.Gov

The main aim of architecture is to define or improve systems of systems (Siegel, 2014). It allows both more understanding of the scope of the SoS and derive a design that satisfies various needs. Also, a layered architecture helps supporting the separation of concerns by enabling different viewpoints of a system as well as the transformation between levels of abstraction (Siegel, 2014). The OntoSoS.QM.Gov SoS architecture shows how the governance process is managed and how the roles involve in the governance process. This will support answering RQ3. (Figure 3-5 illustrates the layered architecture).

The layered architecture in this research consists of the following main layers: (i) infrastructure (development) and raw data layer, (ii) pre-processing (intelligence gathering) and instantiation layer, (iii) operating system layer and (iv) presentation and human involvement layer.

In the *infrastructure layer*, for each CS, all related concepts of processes, standards, QRs and policies are stored in each CS's Local Source System Registry (LSSR). All the former data can be accessed by a Global Source System Registry (GSSR) at the SoS level from the pre-processing layer. This will enable conducting any needed processing and resolving mechanisms at the SoS level in the GSSR pre-processing layer.

Data stored at each LSSR, in each **constituent system**, are as follows:

1) Set of Business Processes= {Pr_0, Pr_1, …Pr_i,…, Pr_r}, Pr: Processes at each CS where 0<= i<=r from Process 0 (Pr_0) to Process r (Pr_r) and associated sub-activities and all other concepts of the process model, e.g. roles titles conducting activities, resources needed for each activity, etc.
2) Set of Standards= {St_0, St_1, …St_i, …, St_t}, St: Standards at each CS where 0<= i<=t from Standard 0 (St_0) to Standard t (St_t) and all other concepts of the standard model, e.g. roles titles needed, resources needed, tools, conditions, etc..
3) Set of QR= {QR_0, QR_1, …QR_i, …, QR_q}, QR: Quality Requirements where 0<= i<=q from QR_0 to QR_q and all other concepts of the QRs model, e.g. quantitative or qualitative metric definitions, constraints, resources, etc.
4) Set of Policies= {Po_0, Po_1, …Po_i, …, Po_p}, Po: Policies at each CS where 0<= i<=p from Policy 0 (Po_0) to Policy p (Po_p) and all other concepts of the policy model, e.g. roles, conditions, etc.

In the *pre-processing layer*, GSSR can access data in LSSR which are stored at each CS's register. The retrieved data from LSSR to GSSR will take the following form:

1) Set of CSs= {CS_0, CS_1, …CSi, … , CS_n}, CS: Constituent Systems where 0<= i<=n, n = number of CSs in the SoS arrangement. CS from System 0 (CS_0) to System n (CS_n).
2) For all CSs, there are set of quality requirements $CS_n.QR_q$ (e.g. $CS_0.QR_0$, $CS_1.QR_1$, …, $CS_i.QR_j$, …, $CS_1.QR_1$, etc.) where 0<= i<=n, n = number of CSs in the SoS arrangement and 0<= j<=q, q= number of QRs in each CS .
3) For all CSs, there are set of standards $CS_n.St_t$ (e.g. $CS_0.St_0$, $CS_0.St_1$, …, $CS_1.St_0$, $CS_i.St_j$, …, $CS_1.St_2$, etc.) 0<= i<=n, n = number of CSs in the SoS arrangement and 0<= j<=t, t= number of standards in each CS.
4) For all CSs, there are set of processes $CS_n.Pr_r$ (e.g. CS_0. Pr_0, CS_1. PrR_1, …, CS_i. Pr_j,…, CS_1. Pr_1, etc.) 0<= i<=n, n = number of CSs in the SoS arrangement and 0<= j<=r, r= number of processes in each CS.
5) For all CSs, there are set of policies $CS_n.Po_p$ (e.g. CS_0. Po_0, CS_0. Po_1, …, CS_1. Po_0, CS_i. Po_j, …, CS_1. Po_2, etc.) 0<= i<=n, n = number of CSs in the SoS arrangement and 0<= j<=p, q= number of policies in each CS.

In the pre-processing layer, the ontological GQGov will be captured and represented using ontology.

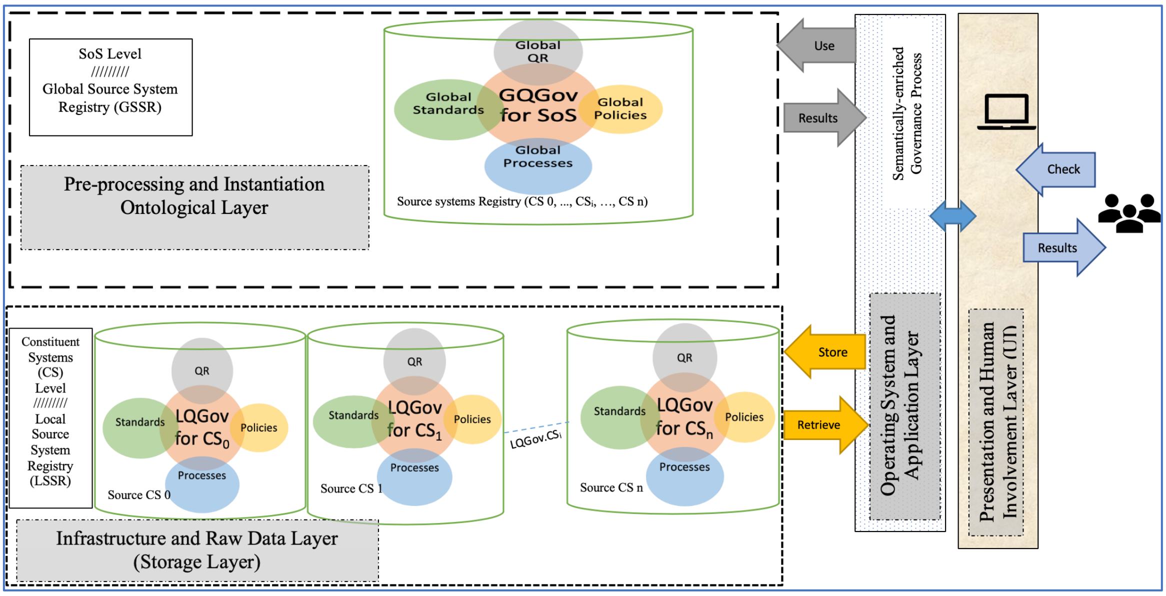

SoS Level
/////////
Global Source System Registry (GSSR)
Pre-processing and Instantiation Ontological Layer
Global QR
Global Standards
GQGov for SoS
Global Policies
Global Processes
Source systems Registry (CS 0, ..., CS$_i$, ..., CS n)
Use
Results
Semantically-enriched Governance Process
Check
Results
Operating System and Application Layer
Presentation and Human Involvement Layer (UI)
Constituent Systems (CS) Level
/////////
Local Source System Registry (LSSR)
QR
Standards
LQGov for CS$_0$
Policies
Processes
Source CS 0
QR
Standards
LQGov for CS$_1$
Policies
Processes
Source CS 1
LQGov.CS$_i$
QR
Standards
LQGov for CS$_n$
Policies
Processes
Source CS n
Store
Retrieve
Infrastructure and Raw Data Layer (Storage Layer)

Figure 3-5: Layered Decomposition of the Quality Governance Framework

In the *presentation and human involvement* layer, end user will interact with the system via computer graphical user interface to check the results of adherence to a QR, process, policy or standard. Human involvement will be at this layer.

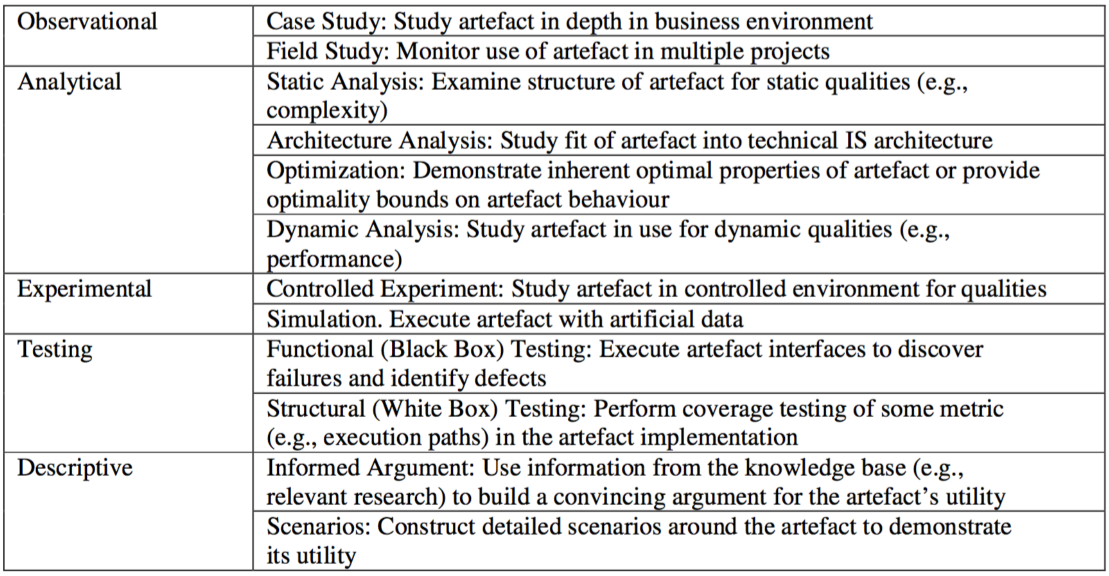
In the *operating system and application* layer, a software system coding is needed to perform queries and to reflect the algorithms used to detail the interactions between policies, QRs, processes and standards in order to check the adherence of them to quality governance.

3.7. Introducing the Research Evaluation Framework

After developing an artefact, it needs to be evaluated to assess how well an artefact works (Hevner et al., 2004). Thus, the evaluation phase of the DSRM provides an important feedback to the preceding research activities. DSRM allows the use of different evaluation methods in each iteration.

After a succession of iterations and improvements, the output of evaluation phase produces the ideal artefacts (Vaishnavi and Kuechler, 2007). There are several approaches to evaluate the artefact as suggested by Hevner et al. (2004) e.g. observational, analytical, experimental, testing, and descriptive techniques. The following are the descriptions of these approaches (see *Table 3-2*).

Table 3-2:Design Science Research Methodology evaluation methods (Hevner et al. (2004))



Observational	Case Study: Study artefact in depth in business environment
	Field Study: Monitor use of artefact in multiple projects
Analytical	Static Analysis: Examine structure of artefact for static qualities (e.g., complexity)
	Architecture Analysis: Study fit of artefact into technical IS architecture
	Optimization: Demonstrate inherent optimal properties of artefact or provide optimality bounds on artefact behaviour
	Dynamic Analysis: Study artefact in use for dynamic qualities (e.g., performance)
Experimental	Controlled Experiment: Study artefact in controlled environment for qualities
	Simulation. Execute artefact with artificial data
Testing	Functional (Black Box) Testing: Execute artefact interfaces to discover failures and identify defects
	Structural (White Box) Testing: Perform coverage testing of some metric (e.g., execution paths) in the artefact implementation
Descriptive	Informed Argument: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artefact's utility
	Scenarios: Construct detailed scenarios around the artefact to demonstrate its utility

3.7.1. A Roadmap for the Development of an Evaluation Framework

The evaluation process of the main artefact of this research study is introduced in this section. Four iterations of evaluations were conducted following the DSRM process. In each DSRM iteration, an evaluation is conducted to validate each developed increment. In each increment. Firstly, an evaluation was applied to assess the structural correctness of the ontology model. Secondly, an evaluation was carried out to validate the interaction relations between the added model and all other components of that increment. Following the evaluation feedback in each DSRM iteration, the feedback was identified to lead into revisiting the design of the components of that increment.

For evaluation, mainly static and dynamic validation have been conducted. The research hypothesis is tested driven by RQs bottom-up. Both Verification (i.e. examines product structure to check that the artefacts being built correctly) and Validation (i.e. examines product semantics to check that the artefact satisfied its purpose) (V&V) methods (Juristo and Morant, 1998) are used to evaluate the main artefacts of the current research (i.e. OntoSoS.QM.Gov and its component models).

The evaluation process followed in this research was adapted from the process illustrated in (Gómez-Pérez, 2001; Juristo and Morant, 1998) with taking into considerations the sub-questions to be evaluated (Bergin and Stokes, 2006) and derived from the main RQs. Table 3-3 provides a breakdown of the research evaluation phases of the OntoSoS.QM.Gov framework.

One of the most commonly used static analysis technique is walkthrough (John and Packer, 1995), where the design is checked by a group of people Another technique is by using design modelling tool, e.g. to check for anomalies in the design (Sommerville, 2000). the evaluation in this research was carried out by conducting interviews (Bergin and Stokes, 2006) and walkthroughs (John and Packer, 1995) with domain experts from the case study. This has been conducted in each increment through DSRM iterations (as shown in Table 3-3). Each increment was evaluated to examine the correctness of its semantic representation of the concepts and their properties and relations for both abstract level which is a domain independent level and the representation of models that represent the domain of the cancer care. Here, the Correctness, Consistency and Completeness (3C check) (Gómez-Pérez, 2001; Juristo and Morant, 1998) are examined. Correctness implies that the main concepts in each model in the case study should be correctly represented using the model (Juristo and Morant, 1998). Also, it needs to completely represent the concepts at the case study with no additional or missing concepts. The consistency check provided by Protégé reasoner tool ("Protege," 2013) was conducted for each increment. This implies that the concepts in each model are consistent with no contradictions between other concepts in the same model.

Table 3-3: Research Evaluation Framework

DSRM Iteratio ns	What to Evaluate? (OntoSoS.QM.Gov Evaluation Components)	How to Evaluate? (Evaluation Type in Each Iteration)		
		Domain Independent Static Verification (Structure correctness)	**Domain dependent Static Validation**	**Domain dependent Validate the effectiveness of the OntoSoS.QM.Gov**
Iteration 1	Increment 1: OntoSoS.QR model	1) Apply consistency check by OWL reasoner tool and check the consistency of the models and their relations with domain experts. 2) Semi-structured interviews, Walkthrough and checklist with domain experts to evaluate the completeness and correctness of the increment/ added model. 3) Comparison with literature's models and ontologies.	Semi-structured interviews, Walkthrough and checklist methods to evaluate the correctness of the increment/added model in terms of satisfaction in representing the case study.	
Iteration 2	New added model of increment 2: OntoSoS.Policy model			
	Increment 2: Interaction process between OntoSoS.QR and OntoSoS.Policy models			
Iteration 3	New added model of increment 3: OntoSoS.Process model			
	Increment 3: Interaction process between OntoSoS.QR, OntoSoS.Policy and OntoSoS.Process models			
Iteration 4	New added model of increment 4: OntoSoS.Stand model			

DSRM Iterations	What to Evaluate? (OntoSoS.QM.Gov Evaluation Components)	How to Evaluate? (Evaluation Type in Each Iteration)		
		Domain Independent Static Verification (Structure correctness)	Domain dependent Static Validation	Domain dependent Validate the effectiveness of the OntoSoS.QM.Gov
	Increment 4: Interaction process between the component models in the OntoSoS.QM.Gov(i.e. interaction between OntoSoS.QR, OntoSoS.Policy, OntoSoS.Process and OntoSoS.Stand models	4) Check the conformance of the conceptual ontological model to standards and principles.		Semi-Structured Interviews and Walkthrough methods with domain experts to compare against quality governance methods used in the case study

3.8. The General Instantiation Method for the Quality Governance Framework Components

3.8.1. Introduction

This section details the general instantiation of the four component models of the OntoSoS.QM.Gov (i.e. OntoSoS.QR, OntoSoS.Policy, OntoSoS.Process and OntoSoS.Stand). The four ontological models were build using ontology. As discussed earlier in Chapter 2, the iterative method of Noy and McGuiness (2001) is adopted in this research to build each model. This method is consistent with the iterative nature of DSRM process adopted in this research (see Figure 3-6). However, more details will be provided in the next four chapters to provide more details regarding each model.

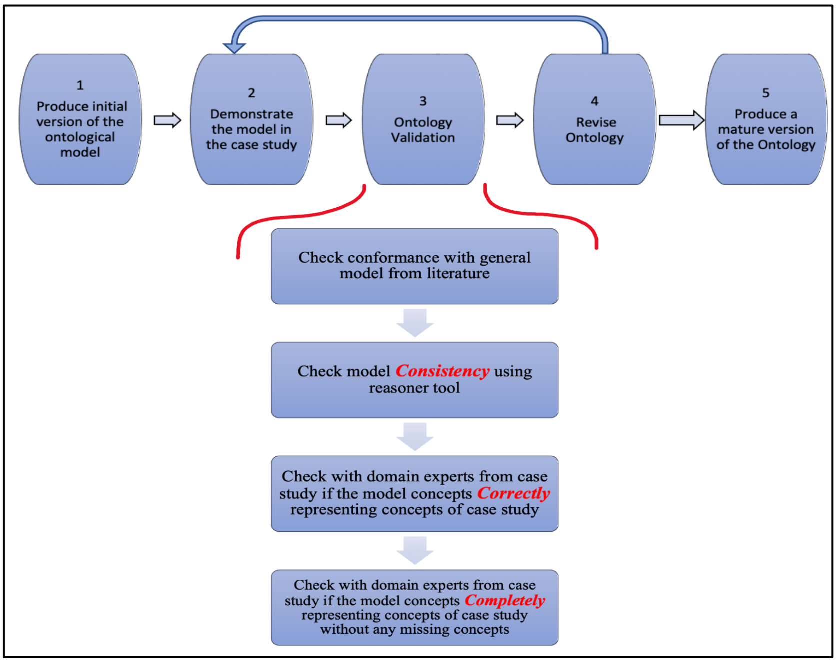

1
Produce initial version of the ontological model
2
Demonstrate the model in the case study
3
Ontology Validation
4
Revise Ontology
5
Produce a mature version of the Ontology
Check conformance with general model from literature
Check model Consistency using reasoner tool
Check with domain experts from case study if the model concepts Correctly representing concepts of case study
Check with domain experts from case study if the model concepts Completely representing concepts of case study without any missing concepts

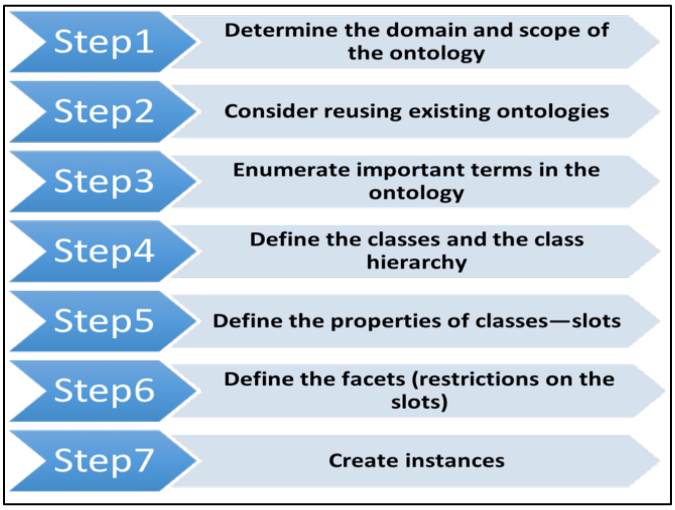
Figure 3-6 :The Iterative Process Used for Constructing Ontology Models (the main steps adopted from [Noy and McGuinness, 2001]

The first step of Noy and McGuiness process is to design an initial version of the ontological model (will be detailed in Section 3.8.2). Then, the resulting ontology will be demonstrated using data from the case study (will be detailed in Section 3.8.3), and then to be validated by domain experts (will be detailed in Section 3.8.4). The feedback of the experts will be used to revisit the syntax of the ontology.

The following sub-sections will overview the general design, demonstration and evaluation steps adopted to create the aforementioned models.

3.8.2. The General Design Method Followed

In order to design and develop each model, seven main steps were followed. These steps defined by Noy and McGuiness (2001) and depicted in Figure 3-7.



Step1
Determine the domain and scope of the ontology
Step2
Consider reusing existing ontologies
Step3
Enumerate important terms in the ontology
Step4
Define the classes and the class hierarchy
Step5
Define the properties of classes—slots
Step6
Define the facets (restrictions on the slots)
Step7
Create instances

Figure 3-7: The Steps Used for Building the Initial Version of Each Ontology Model (adopted from [Noy and McGuinness, 2001])

The steps are as follows:

i) Determine the domain and scope of the ontology: both the domain that the ontology will cover and the purpose the ontology is used for. The ontology domain in this research is related to knowledge representation of quality requirements for the OntoSoS.QR model, policies for the OntoSoS.Policy model, processes for the OntoSoS.Process model, and standards for the OntoSoS.Std model. Each ontology is a part of local quality governance model (LQGov(s)) at monolithic system level. Then, the LQGov(s) that have been built at monolithic system level will be semantically linked to develop the GQGov (global quality governance) at the SoS level.

ii) Consider reusing existing ontologies: reuse exiting ontologies for standards, processes, policies and quality requirements either fully or partially.

iii) List the key terms in the ontology: after studying existing ontologies, the main concepts and their relationships are either extracted from the existing ontologies or from literature to form the basic concepts in each ontology. If an existing ontology has been selected, then, it can be extended with any new concepts. In this step, documents related to QR definitions, policies, standards and processes from the case study will be reviewed in order to enrich the set of terms identified.

iv) Define class hierarchy: the main concepts of each ontology will be defined as parent classes and their sub-classes. In this step, all identified terms will be categorised and linked together into groups to define the hierarchy between them.

v) Define the properties (slots) of classes: two types of properties are to be defined; object properties which represent the relations between classes and data properties, which represent data types of each class. In this step, the documents made available by the case study can be reviewed to elicit the properties of each class.

vi) Define the facets (restrictions) of the slots: cardinality of each relation, types of each value, and domain and range of each property need to be identified.

vii) Create instances: data from the case study will be used to create individual instances of classes. However, at this stage, only test instances may need to be created. Next, real data from the case study will be used (see Section 3.8.3).

viii) An additional step needs to be conducted, which is to decide which model to be developed and linked with other developed models. Then, an interaction process needs to be developed to check the conformance between the models.

All abstract models will be conceptualised using Unified Modelling Languages (UML) diagrams. Here, any UML tool can be used to draw the diagrams, e.g. StarUML, Visual Paradigm, etc. Also, the capability of generating diagrams embedded in the Protégé tool can be used as well. These diagrams will help in the demonstration and evaluation steps later.

The aforementioned seven steps were utilised in the development of the four research framework ontologies and will be further detailed in the design section of each of the chapters (4-7).

3.8.3. The General Demonstration Method Followed

After developing each ontological model using Protégé tool, data from the case study will be used in order to create real instances. Many resources here can be used to create the instances, e.g. documents, flowcharts, standards chapters, interviews, etc. Creating instances will facilitate the understanding of the demonstration of each model. Then, some of these data can be represented by creating data diagrams in order to provide more clarity to the connected instances, which will facilitate the validation of the models by domain experts.

3.8.4. The General Evaluation Method Followed

In order to evaluate each model, several methods were adopted (as detailed earlier in Section 3.7.1 and Table 3-3). The general evaluation steps conducted for each model as follows:

i) Firstly, a comparison needs to be conducted between the ontological model and other models in the literature.

ii) Secondly, to verify the structural correctness and consistency of the ontology models and the semantic representation of the concepts and their properties and relations, semi-structured interviews (Bergin and Stokes, 2006), Walkthrough technique (John and Packer, 1995) and checklist need to be conducted with domain experts to evaluate the structural correctness of the model in terms of satisfaction in representing the concepts related to the case study. These models were developed using the generalised design method. The experts need to check the definitions of the ontology concepts. Then, to check the conceptualisation of each model.

iii) Thirdly, several implemented test cases will be selected to be reviewed by domain experts from the case study to validate the models by assessing the satisfaction of the complete representation of the various elements and concepts in the case study. Here, semi-structured interviews and Walkthrough methods will be conducted with domain experts.

iv) Fourthly, the aforementioned evaluation steps will be repeated for each model. This will produce a second version of each model. Then, another evaluation step will be carried out to evaluate the effectiveness of the overall SoS quality governance interaction process between processes, policies and standards to inform the adherence to quality in the SoS context. Here, semi-structured interviews will be conducted with domain experts from KHCC to inform the satisfaction of the results of the whole interaction process. More details will be provided in Section 7.6.

The domain experts (i.e. interviewees) selected in order to validate the resulting artefacts include staff from quality assurance offices, IT experts from IT department and mangers of KHCC Admission, Discharge and Transfer (ADT), and Cell Therapy and Applied Genomics (CTAG) departments. The number of interviewees was 3-5 as the type of the roles of the participants were from middle and top management only. This number was not as significant as the selecting suitable interviewees who are able to provide the appropriate knowledge and familiarity in the areas of quality requirements and modelling. Patton (1990, p. 184) indicated that, in qualitative research, there are no specific rules that can be applied for sample size. Moreover, Patton continued to outline that the number of interviewees depends on their reliability, the objectives of the interviews, and time and resources availability. Also, he emphasised as well that, "the validity, meaningfulness, and insights generated from qualitative inquiry have more to do with the information-richness of the cases selected than with sample size" (Patton, 1990, p. 185).

In this research, selection criteria proposed included competencies that mainly related to the position, professional experience and the willingness of participants to be part of the evaluation process of the governance framework. For example, (i) knowledge and experience in one or more of the following areas: governance activities in KHCC, quality requirements, policies, standards, (and/or) understanding the processes and procedures inside KHCC, (ii) the domain experts selected have enough time to participate in the evaluation process, (iii) other preferred criteria (optional), e.g. general knowledge and understanding of modelling and semantic representation, experienced enough with English language, participants represent middle or top-level managerial roles, and general knowledge and understanding of the evaluation process. Satisfying the aforementioned criteria will provide reliable information and sufficient knowledge from experts that can be used in this research to report the effectiveness of the models and to reflect the satisfaction of the experts from the KHCC.

3.9. Chapter Summary

The research framework design has been introduced in this chapter using the Design Science Research Methodology (DSRM) process along with a case study to facilitate the design, implementation and evaluation of quality governance framework in the SoS context.

The main research artefact namely the OntoSoS.QM.Gov framework has been designed, implemented and evaluated incrementally through four iterations following the DSRM process.

The KHCC cancer care case study has been utilised and its selection has been justified due to its representative of the domain and its sufficiency to conduct the research governance process employing quality requirements, processes, standards and policies models.

4. Chapter Four: The Quality Requirements Ontology Model (DSRM Process Increment I)

4.1. Introduction

This chapter details the first iteration of the DSRM process followed in this research. Firstly, the design of the first increment of the quality governance framework which is the quality requirements model (i.e. OntoSoS.QR model) is detailed. Then, the demonstration of this model using a case study from the cancer care domain is illustrated, followed by an evaluation process with the resulting feedback utilised to revisit the design of the quality requirements model, following the DSRM process.

4.2. The Design of the OntoSoS.QR Ontology Model: Increment 1

Following the general steps of ontology design process illustrated in Section 3.8.2, the steps are conducted as follows: Firstly, the ontology domain is related to knowledge representation of quality requirements for the OntoSoS.QR model. Secondly, in order to reconsider/ extend existing ontologies, several QR ontology models in literature have been reviewed and presented in Chapter 2, Section 2.8. Table 4-1 presents the most relevant models adopted in this research to define the QR model. Thirdly, the QR ontology terms were adopted from existing models and new concepts were added. For example, to involve the capability of related aspects to SoS, new concepts need to be considered in the construction of the OntoSoS.QR ontology. Another example, because the constituent systems of the SoS arrangement may be connected to other systems or SoS(s), some new concepts need to be created (e.g. External System, Constituent System (CS), Non-Constituent System, SoS level). Also, other QR- related aspects need to be identified. For instance, roles and priority (*INCOSE*, 2015). Furthermore, to enable the trade-off between requirements in the case of the existence of conflicts between them, it is better to have some concepts that support prioritising and selecting which QR is the best to adhere to, for example the concepts of metrics and operational constraints for each quality requirement and the relationships and dependencies between them. Fourthly, in order to define class hierarchy, the quality Requirements taxonomies need to be identified. The taxonomies of Van Lamsweerde (2009) and Odeh & Odeh (2011) has been adopted and introduced in Chapter 2. Then, in order to have more flexible and extendable ontology, the ontology can be separated into several sub-ontology models/views (i.e. roles view, quality attributes view, quality property view, metric view and constraint view). As shown in Table 4.1, some parts of these sub-models have adopted from literature and then extended to include new concepts. The last two steps are to identify the main classes and their hierarchy. All the concepts of OntoSoS.QR (i.e. main classes and their properties) are shown in Table 4-2. The Figures Figure *4-1*, Figure *4-2*, Figure *4-3* and Figure *4-4* show the conceptual representation of the variant views of the OntoSoS.QR model. Also, the whole OntoSoS.QR is conceptualised in Figure 4-5.

Table 4-1: The traceability of the parts/views of the OntoSoS.QR ontology to Literature Sources

Part /view name	Traceability to literature
Roles and Service	Adopted from (Tran et. al., 2009). The concept of the QoS Group extended to include SoS related group.
Quality main attributes	Adopted from (Tran et. al., 2009).
Quality property	Adopted from (Van Lamsweerde,2009) and (Odeh & Odeh, 2011).
Metric view	Adopted from (Dobson et.al., 2005), (Tondello & Siqueira, 2008), (Maximilien & Singh, 2004).
Constraint view	Adopted from (Tondello & Siqueira, 2008) and (Odeh & Odeh, 2011).
SoS	Extended for the purposes of this research.

Table 4-2: OntoSoS.QR Main Concepts and Traceability to Literature Sources

SN	Concepts	Description and Resources	Properties and their types
1	**OntoSoS.QR model**	This class/ concept describes the ontology of the SoS quality requirements model in order to have a central repository for this model.	N/A
2	Quality Requirements (QR)	Requirements that are not directly concerned with the specific services delivered by a system to its users (Sommerville, 2011). Describe all QR that are related to quality governance as a product for SoS. Classification adopted from Van Lamsweerde, (2009 and Odeh & Odeh, (2011).	**name**: string. title of the QR. **description**: string. description of the QR. **version**: string. version number (if any). **has_Relationship:** has Relationship to another QR **relatedToGroup:** related to one of QR Groups: user, system, or/and SoS. **appliedToService.** QR could be applied to service. **hasMandatory.** has mandatory relation to QoS mandatory concept. **hasWeight.** has weight relation to QoS weight concept. **hasTendency.** has tendency relation to QoS tendency concept. **hasValidPeriod.** has valid period relation to period concept. **hasMetric.** has metric relation to QoS metric concept.
3	Usability	According to (ISO/IEC 25010, 2011), it is the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use - Definition : " Usability requirements are concerned with specifying the user interface	Same as 2.

		and end-user interactions with the system" (Kotonya & Sommerville, 1998). - usability requirements have four types: Entry Requirements, Learning Requirements, Handling Requirements and Likeability Requirements (Kotonya & Sommerville, 1998).	
4	Learnability	It is the degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use (ISO/IEC 25010, 2011).	Same as 2.
5	Operability	It is the degree to which a product or system has attributes that make it easy to operate and control (ISO/IEC 25010, 2011).	Same as 2.
6	User Interface Aesthetics	It is the degree to which a user interface enables pleasing and satisfying interaction for the user (ISO/IEC 25010, 2011).	Same as 2.
7	Understandability/ Appropriateness Recognizability	It is the degree to which users can recognize whether a product or system is appropriate for their needs (ISO/IEC 25010, 2011).	Same as 2.
8	Accessibility	It is the degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use (ISO/IEC 25010, 2011).	Same as 2.
9	Accuracy	It is the degree to which a product or system provides the correct results with the needed degree of precision (ISO/IEC 25010, 2011).	Same as 2.
10	Reliability	"The degree to which a system, product or component performs specified functions under specified conditions for a specified period of time" (ISO/IEC 25010, 2011).	Same as 2.
11	Maturity	It is the degree to which a system, product or component meets needs for reliability under normal operation (ISO/IEC 25010, 2011).	Same as 2.
12	Availability	It is the degree to which a system, product or component is operational and accessible when required for use (ISO/IEC 25010, 2011).	Same as 2.
13	Fault Tolerance	It is the degree to which a system, product or component operates as intended despite the presence of hardware or software faults (ISO/IEC 25010, 2011).	Same as 2.
14	Recoverability	It is the degree to which , in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system (ISO/IEC 25010, 2011).	Same as 2.
15	Security	"The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization" (ISO/IEC 25010, 2011).	Same as 2.

16	Integrity	“ The degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data” (ISO/IEC 25010, 2011).	Same as 2.
17	Confidentiality	“ The degree to which a product or system ensures that data are accessible only to those authorized to have access” (ISO/IEC 25010, 2011).	Same as 2.
18	Safety	Safety-critical systems are systems where it is essential that system operation is always safe; that is, the system should never damage people or the system’s environment even if the system fails (Sommerville, 2011). Definition: " How well the system protects against injury or damage" (Wiegers & Beatty, 2013, p 263). Definition: " Safety requirements deal with the need to prevent a system from doing any injury to people or damage to property. " (Wiegers & Beatty, 2013, p 276 cited from (Leveson 1995; Hardy 2011)).	Same as 2.
19	Economic Safety	It is the degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use (ISO/IEC 25010, 2011).	Same as 2.
20	Health and safety Risk Mitigation	It is the degree to which a product or system mitigates the potential risk to people in the intended contexts of use (ISO/IEC 25010, 2011).	Same as 2.
21	Environmental Safety	It is the degree to which a product or system mitigates the potential risk to property or the environment in the intended contexts of use (ISO/IEC 25010, 2011).	Same as 2.
22	Performance / Efficiency	“It is the amount of resources used under stated conditions” (ISO/IEC 25010, 2011).	Same as 2.
23	Time Behaviour related (Response Time, Throughput)	It is the degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements (ISO/IEC 25010, 2011).	Same as 2.
24	Resource Utilization	It is the degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements (ISO/IEC 25010, 2011).	Same as 2.
25	Capacity/ Space related	It is the degree to which the maximum limits of a product or system parameter meet requirements (ISO/IEC 25010, 2011).	Same as 2.
26	Condition	The QR has a condition(s) to happen (i.e. describes the conditions under which the QR is constrainedd) (e.g. activity or task as a prerequisite for satisfying the QR, any additional information or circumstances, etc.)	
27	System	Any system that is an external system/CS/ SoS is interoperable with. In the case of SoS, quality requirements at the SoS level may conflicts with satisfying the quality requirements of one or more of the constituent systems. So, these conflicts need to be detected and with which systems (INCOSE, 2015).	**name**: string. title of the system.
28	Constituent System (CS)	System that is part of the SoS arrangement (INCOSE, 2015).	**name**: string. title of the system.

29	Non-Constituent System	Any External System that is not part of the SoS arrangement.	**name**: string. title of the system.
30	Constraint	- "An external factor that prevents an organization from pursuing particular approach to meet its goals" (TOGAF, 2011). - Constraints can be either positive (always do this) or negative (never do this) (ISO/IEC 21827, 2008). - May be imposed by a (client) to restrict the implementation of the system or development process. (Sommerville, 2011). There are five types of constraints: contract, operator, schedule, cost and architectural constraints. - Maybe imposed by CS/SoS/External system	**name**: string. title of the constraint.
31	Assessment Measure/ QoS Metric	A metric to measure the QR satisfaction "An indicator or factor that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives and goals" (TOGAF, 2011).	**name**: string. title of the metric.
32	Qualitative Measure	an Assessment Measure that is a qualitative. Satisfying Status: string list: Fully satisfied, satisfied, partially satisfied, unsatisfied	**satisfyingStatus: string list.** 'satisfied, partially satisfied, unsatisfied'
33	Quantitative Measure	an Assessment Measure that is a quantitative. It is a Double value.	**quantitativeValue: double**
34	Role	- Role: "The usual or expected function of an actor or event. An actor may have a number of roles" (TOGAF, 2011). -Each role needs to be performed by an actor (person) or department in order to govern the application of the QR. Each role needs to be performed by an actor (person) or department in order to govern the application of the QR.	**name**: string. title of the role.
35	QoS Mandatory	(Tran et al., 2009) This requirement allows a service requester to specify which QoS properties are strongly required whilst others may be optional. To give a value to indicate the level of mandatory of a QR. Mandatory value list is strongly required, required, optional.	**MandatoryValue: string list.** 'strongly required, required, optional'
36	QoS Tendency	(Tran et al., 2009) This property is an important factor for evaluating QoS metrics and their values. A QoS property can have one of these impact directions: negative, positive, close, and exact. -exact: the property's value must be exactly the same as the requested value. For example, authorization method can be specified as an exact QoS property. -close: the property's value is as close with the requested value as better than the others. -positive: the higher value is better than the lower one. For example, throughput, availability, and accessibility are positive QoS properties	**tendencyValue: string list.** 'negative, positive, close, and exact'
38	QoS Group	QoS ontology should allow for grouping QoS properties which share similar characteristics or impacts in order to facilitate a process of computing service ranking value. The QoS gropsg (system and user groups) idea from Tran et al., (2009). Also, brainstorming to the SoS group.	**name**: string. title of the group.

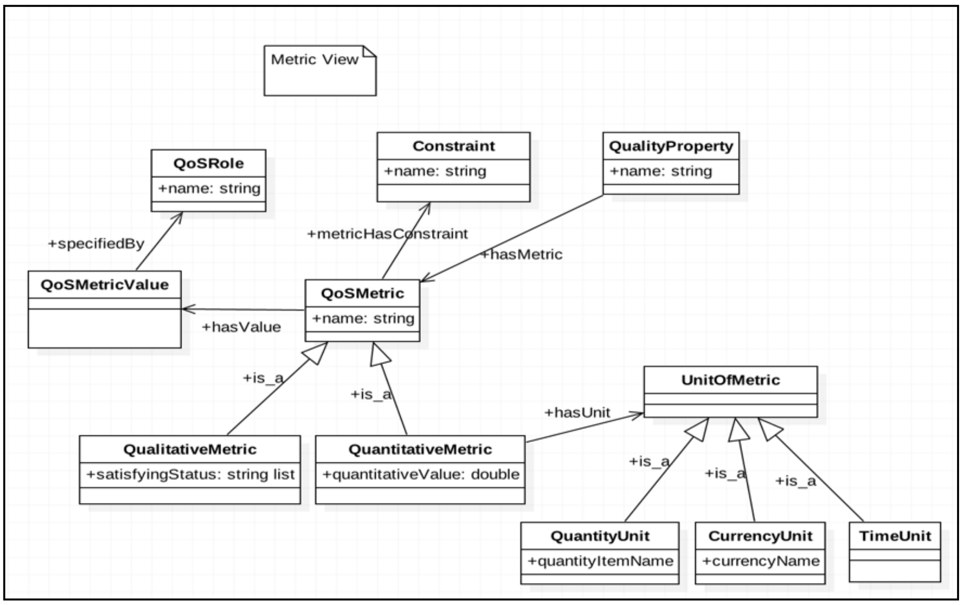


Figure 4-1:The First Version of the Metric View of OntoSoS.QR

Figure 4-1 shows that each quality requirement has a metric which has a value specified by a role. This value can be continuously tracked or tested to check the adherence to a certain level of quality. Depending on the quality requirement, the measure value may be either qualitative or quantitative. An example of a qualitative value is the level of security to be satisfied, partially satisfied or unsatisfied. In the case of quantitative value, it needs to have a specific value, e.g. the speed of delivering a blood sample should be within two hours. It is important to know the values of each metric in order to check if values of a quality requirement in one constituent system is similar to the value of the same quality requirement in another constituent system.

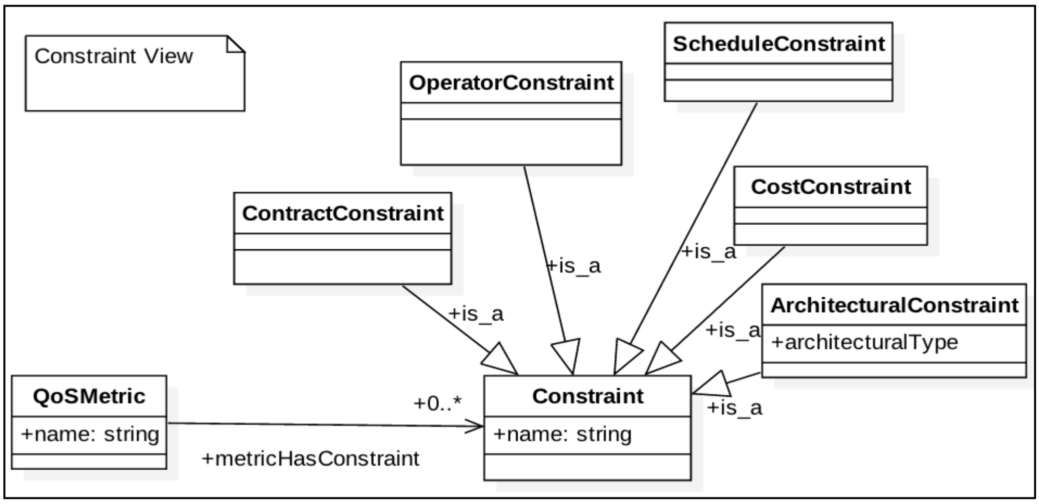


Figure 4-2: The First Version of the Constraint view of OntoSoS.QR

Figure 4-2 shows that the metric value may be restricted with one or more constraints. This constraint represents an external factor that can affect the measure value and hence, it is important to know any factors that prevent a measure to reach its goal. For example, the quality requirement of 'ensure financial coverage' has 'Safety: Economic risk mitigation' constraint with regard to 'cost constraint'.

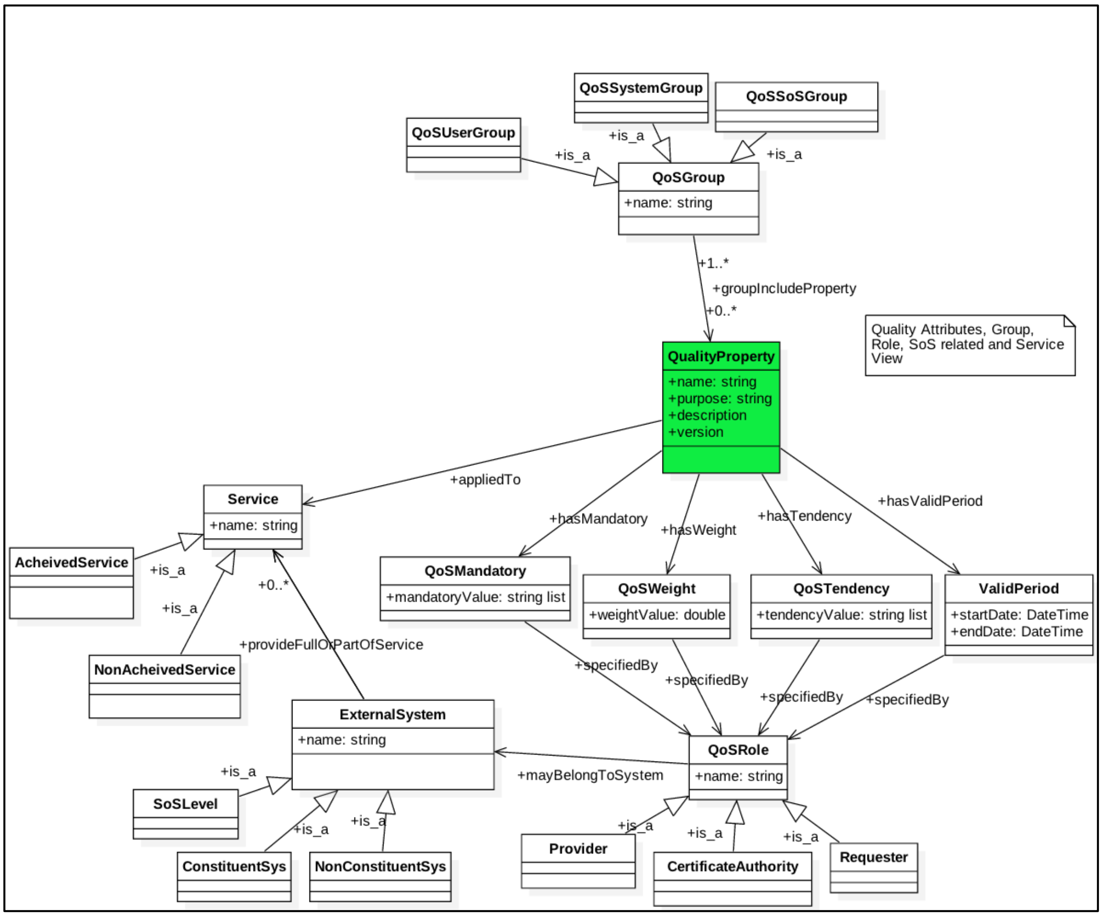


Figure 4-3:The First Version of the Quality Attributes, Groups, SoS related and Service Views of OntoSoS.QR

Figure 4-3 shows that a quality requirement has several characteristics which are mandatory, weight, tendency and valid period. Tran et.al. (2009) indicated that the 'weight' attribute represents the priority level of a QR over the others whilst the 'mandatory' attribute indicates that if a QR is required or it is optional.

The 'valid period' of a QR can also be specified in a certain time and then can be changed. 'Tendency' is an important factor for evaluating QR metrics and their values. A QR can have one of these tendency impact directions: 'negative', 'positive', 'close', and 'exact'. The impact named 'Exact' is the QR's value must be exactly the same as the requested value. For example, authorization method can be specified as an exact QR. The impact named 'Close' value is as close with the requested value as better than the others. The impact 'Positive' indicates that the higher value is better than the lower one. For example, the 24 hours ‘availability’ is better than 12 hours availability, while the ‘negative’ impact of a QR is when the lower value is the better, e.g. response time.

Another aspect is related to the role who specify the value of a QR’s characteristics. Also, to specify which system the role is part of, e.g. a constituent system in the SoS arrangement or a system that is out of the SoS arrangement or it could be specified as a requirement to be fulfilled related to the main objectives of the SoS.

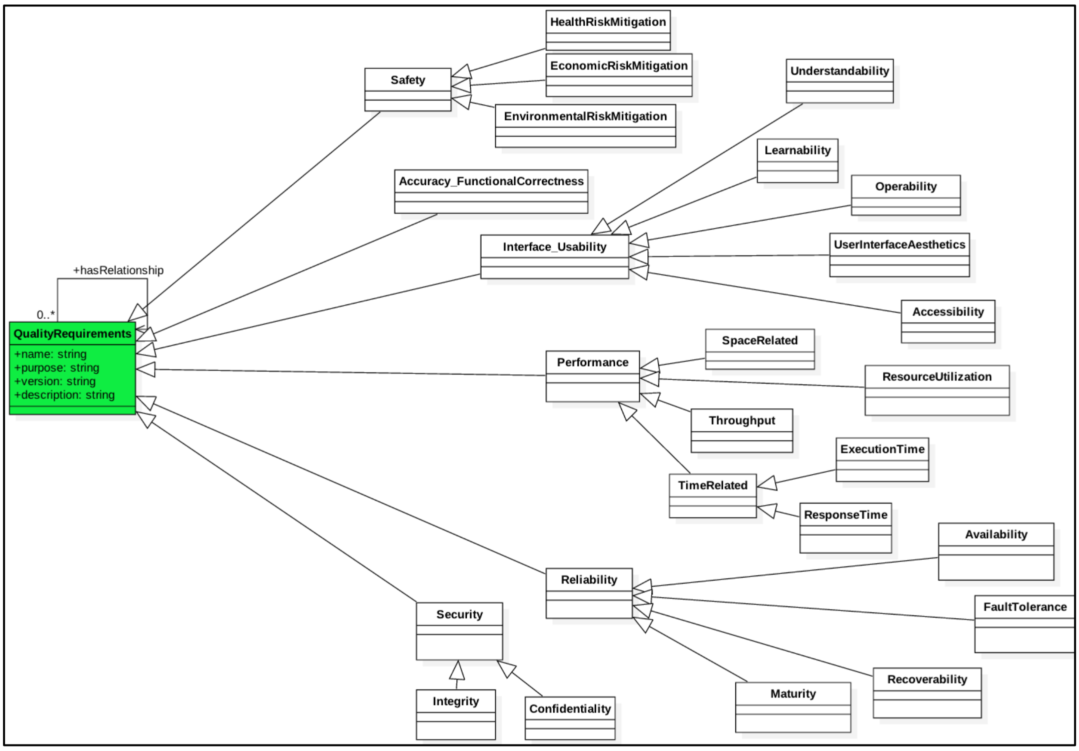


Figure 4-4: First Version of the Quality Requirements Hierarchy View of OntoSoS.QR

Figure 4-4 shows the hierarchy of the quality requirements. Also, it shows that some QRs have relations with other QRs. It is important to relate the QRs to each other to be taken into consideration in the case of conflict.

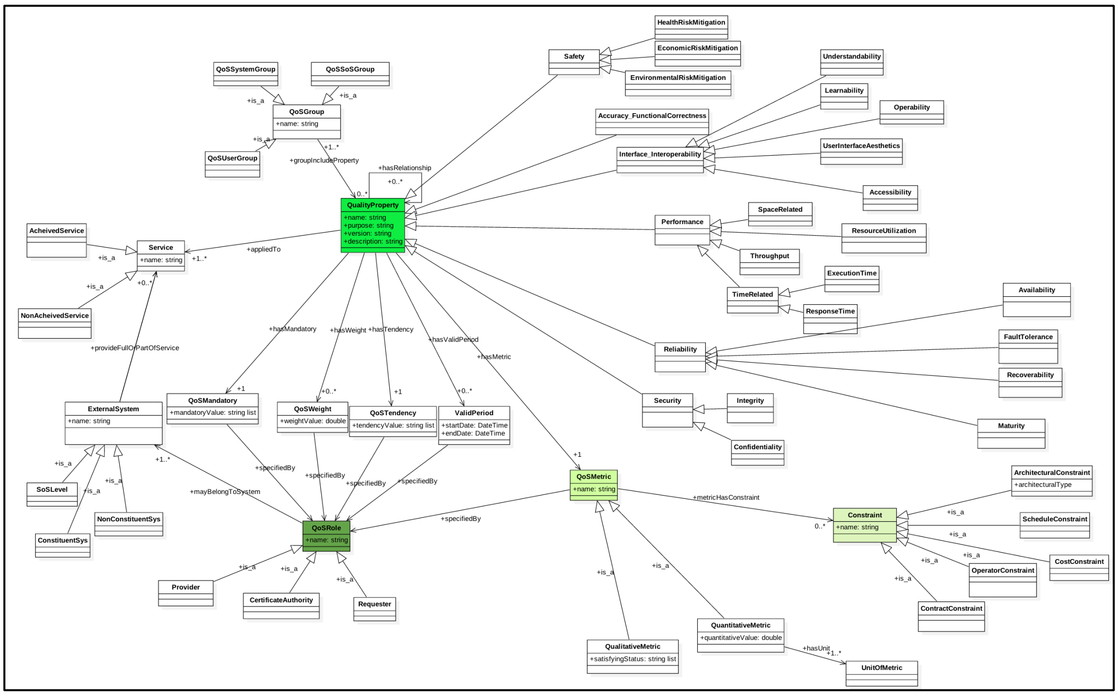


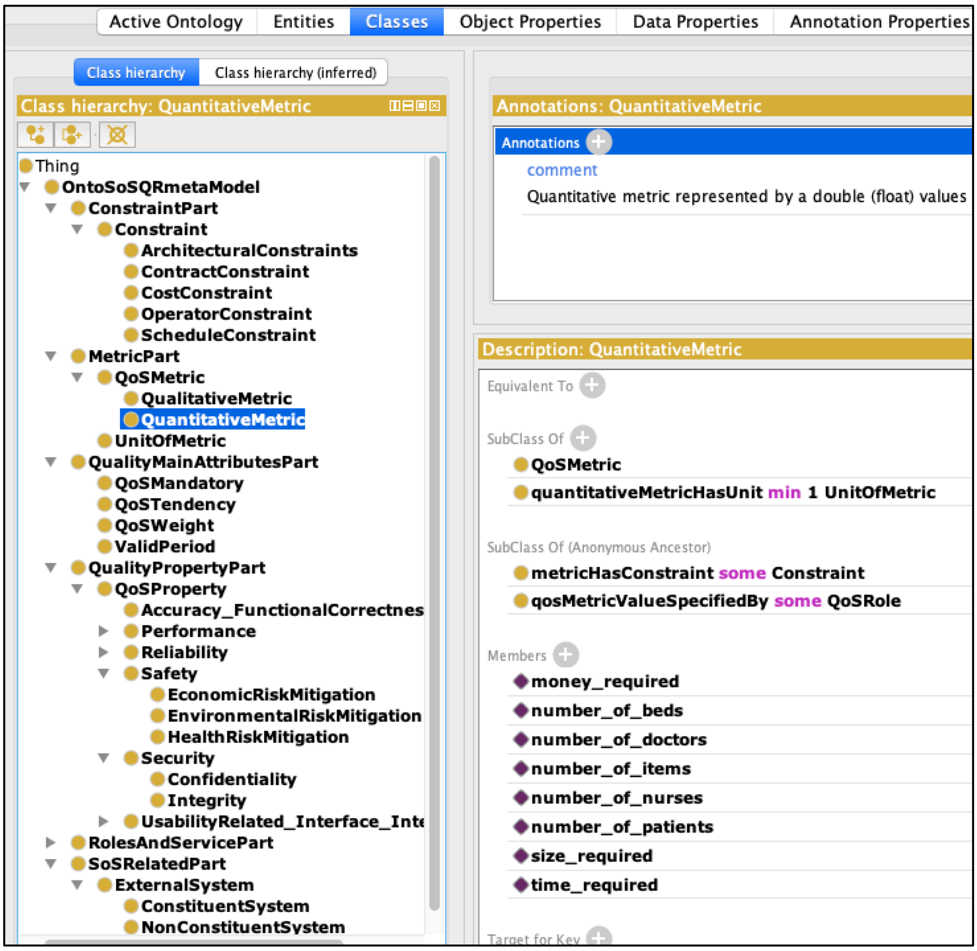
Figure 4-5: The Conceptual Representation of Version 1 of OntoSoS.QR

After having presented and discussed the OntoSoS.QR ontology conceptualisation model, the second step is to demonstrate this model using Protégé tool with the research case study. Figures Figure *4-6* and Figure *4-7* show snapshots of the QR ontology representation and some examples of instances from the case study that were created using the Protégé ontology editor.

4.3. The Demonstration of Increment 1

The next two figures (i.e. Figure *4-6* and Figure *4-7*) show some examples of instances from the case study that were created using Protégé tool. However, in order to facilitate the understanding of the demonstration of the QR model, conceptualisation with data from the case study. Figure 4-8 is an example regarding a quality requirement data model.

To demonstrate the OntoSoS.QR ontology model with KHCC case study, the quality requirements concepts need to be mapped with the corresponding titles in the KHCC. For example, (as in Figure 4-6) the values of quantities can be referred to the ‘number of beds’, ‘number of patients’, ‘number of minutes’ , etc. Also, the quality requirements can be mapped to corresponding health care requirements. For instance, ‘safety’ can be expressed as ‘safe discharge’, ‘patients’ safety’, ‘tissue's doner safety’, etc. Another example is the ‘resource utilisation’ which can be expressed as ‘bed utilisation’, ‘efficient use of plasma extraction’, etc.



Class hierarchy: QuantitativeMetric

```
Thing
▼ OntoSoSQRmetaModel
  ▼ ConstraintPart
    ▼ Constraint
        ArchitecturalConstraints
        ContractConstraint
        CostConstraint
        OperatorConstraint
        ScheduleConstraint
  ▼ MetricPart
    ▼ QoSMetric
        QualitativeMetric
        QuantitativeMetric
      UnitOfMetric
  ▼ QualityMainAttributesPart
      QoSMandatory
      QoSTendency
      QoSWeight
      ValidPeriod
  ▼ QualityPropertyPart
    ▼ QoSProperty
        Accuracy_FunctionalCorrectnes
      ► Performance
      ► Reliability
      ▼ Safety
          EconomicRiskMitigation
          EnvironmentalRiskMitigation
          HealthRiskMitigation
      ▼ Security
          Confidentiality
          Integrity
      ► UsabilityRelated_Interface_Inte
  ► RolesAndServicePart
  ▼ SoSRelatedPart
    ▼ ExternalSystem
        ConstituentSystem
        NonConstituentSystem
```

Annotations: QuantitativeMetric

Annotations

comment
Quantitative metric represented by a double (float) values

Description: QuantitativeMetric

Equivalent To

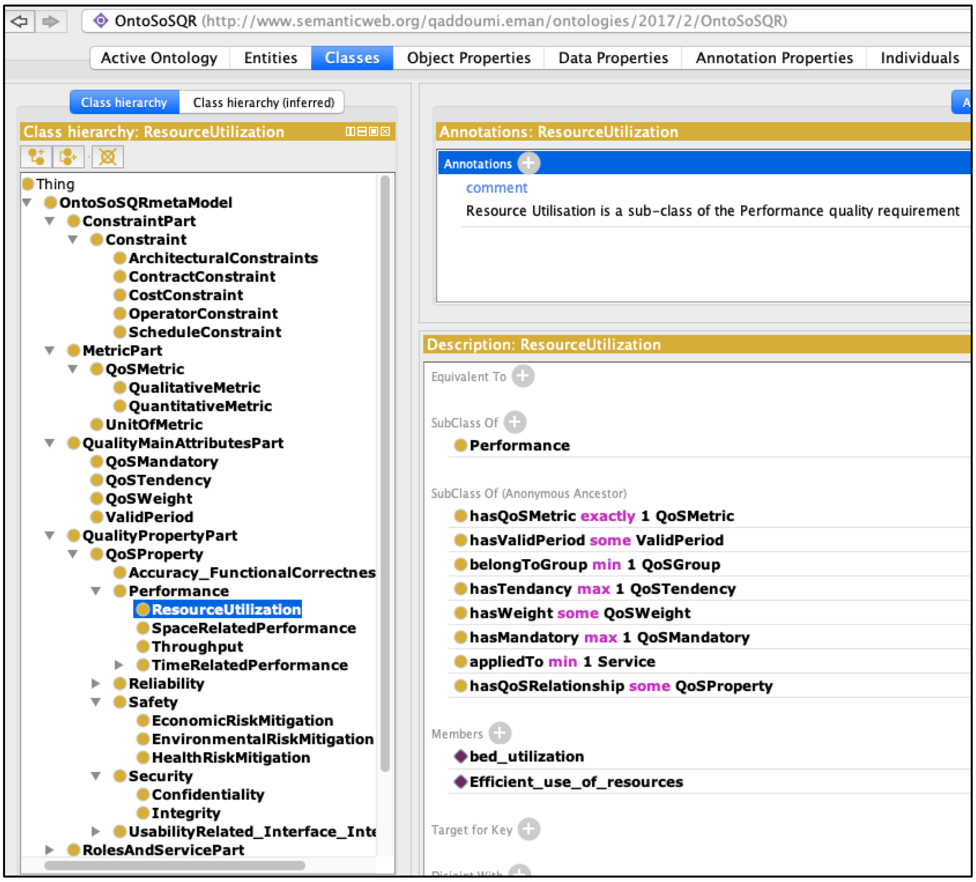
SubClass Of
- QoSMetric
- quantitativeMetricHasUnit min 1 UnitOfMetric

SubClass Of (Anonymous Ancestor)
- metricHasConstraint some Constraint
- qosMetricValueSpecifiedBy some QoSRole

Members
- money_required
- number_of_beds
- number_of_doctors
- number_of_items
- number_of_nurses
- number_of_patients
- size_required
- time_required

Target for Key

Figure 4-6: Snapshot 1 to show a part of the ontology of QR model



Class hierarchy: ResourceUtilization

```
Thing
▼ OntoSoSQRmetaModel
  ▼ ConstraintPart
    ▼ Constraint
        ArchitecturalConstraints
        ContractConstraint
        CostConstraint
        OperatorConstraint
        ScheduleConstraint
  ▼ MetricPart
    ▼ QoSMetric
        QualitativeMetric
        QuantitativeMetric
      UnitOfMetric
  ▼ QualityMainAttributesPart
      QoSMandatory
      QoSTendency
      QoSWeight
      ValidPeriod
  ▼ QualityPropertyPart
    ▼ QoSProperty
        Accuracy_FunctionalCorrectnes
      ▼ Performance
          ResourceUtilization
          SpaceRelatedPerformance
          Throughput
        ► TimeRelatedPerformance
      ► Reliability
      ▼ Safety
          EconomicRiskMitigation
          EnvironmentalRiskMitigation
          HealthRiskMitigation
      ▼ Security
          Confidentiality
          Integrity
      ► UsabilityRelated_Interface_Inte
  ► RolesAndServicePart
```

Annotations: ResourceUtilization

Annotations

comment
Resource Utilisation is a sub-class of the Performance quality requirement

Description: ResourceUtilization

Equivalent To

SubClass Of
- Performance

SubClass Of (Anonymous Ancestor)
- hasQoSMetric exactly 1 QoSMetric
- hasValidPeriod some ValidPeriod
- belongToGroup min 1 QoSGroup
- hasTendancy max 1 QoSTendency
- hasWeight some QoSWeight
- hasMandatory max 1 QoSMandatory
- appliedTo min 1 Service
- hasQoSRelationship some QoSProperty

Members
- bed_utilization
- Efficient_use_of_resources

Target for Key

Figure 4-7: Snapshot 2 to show a part of the ontology of QR model

Figure 4-6 shows the main classes of the model. One concept that is detailed by this snapshot is the concept of ‘quantitative metric’. It indicates some of the rules that applied on this concept. One rule is specific to this class, i.e. it has at least one metric unit. Other rules are common with all ‘metric’ classes, e.g. the ‘role’ who may specify the value of the ‘unit’ and any ‘constraint’ that may be applicable to this metric. Also, it shows some instances of this class e.g. the quantity of items (number of beds, size required, etc...). Another example is presented in Figure 4-7, which shows a ‘resource utilisation’ as a ‘performance’ indicator. It shows some rules/constraints that need to be applied on this class and other QRs, e.g. it may have a relationship to another quality requirement, and it has exactly one metric value.

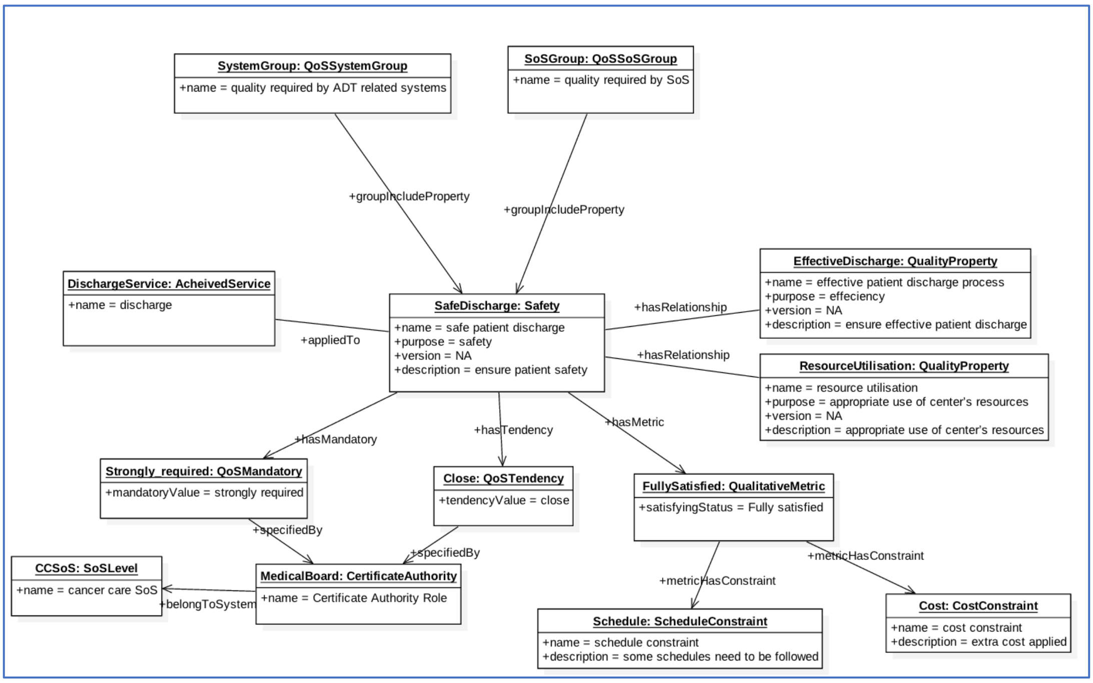

SystemGroup: QoSSystemGroup
+name = quality required by ADT related systems
SoSGroup: QoSSoSGroup
+name = quality required by SoS
+groupIncludeProperty
+groupIncludeProperty
DischargeService: AcheivedService
+name = discharge
+appliedTo
SafeDischarge: Safety
+name = safe patient discharge
+purpose = safety
+version = NA
+description = ensure patient safety
+hasRelationship
EffectiveDischarge: QualityProperty
+name = effective patient discharge process
+purpose = effeciency
+version = NA
+description = ensure effective patient discharge
+hasRelationship
ResourceUtilisation: QualityProperty
+name = resource utilisation
+purpose = appropriate use of center's resources
+version = NA
+description = appropriate use of center's resources
+hasMandatory
+hasTendency
+hasMetric
Strongly_required: QoSMandatory
+mandatoryValue = strongly required
Close: QoSTendency
+tendencyValue = close
FullySatisfied: QualitativeMetric
+satisfyingStatus = Fully satisfied
+specifiedBy
+specifiedBy
CCSoS: SoSLevel
+name = cancer care SoS
+belongToSystem
MedicalBoard: CertificateAuthority
+name = Certificate Authority Role
+metricHasConstraint
+metricHasConstraint
Schedule: ScheduleConstraint
+name = schedule constraint
+description = some schedules need to be followed
Cost: CostConstraint
+name = cost constraint
+description = extra cost applied

Figure 4-8: Quality Requirement Object Model- “Safe Discharge” Quality Requirements

4.4. The Evaluation of Increment 1

The first step of evaluation was conducting a comparison between the OntoSoS.QR and other QR models (Qaddoumi et al., 2018a). The results of this comparison are presented in Table 4-3. The comparison criteria are the elements of the key concepts that need to be considered for developing a quality requirement model, presented in Section 2.8.

Table 4-3:Comparison between some Quality Requirements Models in Literature

	Wide support for QR properties	Support metrics-related aspects	Wide support for Relations among concepts	Degree of flexibility / extendibility	Wide support for Quality attributes	Support Roles	Support SoS-related concepts	Ontology-based model
Dobson et.al. (G. Dobson et al., 2005; Glen Dobson et al., 2005)	Partially	Yes	Partially	Medium degree of extendibility	No	No	No	Yes
Tran et. al. (Tran et al., 2009)	Mostly	Yes	Yes	High degree of extendibility	Yes	Yes	No	Yes
Zhou et.al. (Zhou et al., 2005)	Yes	Partially	Partially	Low degree of extendibility	No	No	No	No
Maximilien and Singh (Maximilien and Singh, 2004)	Partially	Mostly	Yes	Low degree of extendibility	Partially	Agent role only	No	Yes
Kassab et. al. (Kassab et al., 2009)	Mostly	Yes	Mostly	Medium degree of extendibility	No	No	No	Yes
ElicitO model (Al Balushi et al., 2013, 2007)	Partially	Partially	Yes	Low degree of extendibility	No	No	No	Yes
OntoSoS.QR	Yes	Yes	Yes	High degree of extendibility	Yes	Yes	Yes	Yes

Table 4-3 shows a comparison between the OntoSoS.QR ontology model and some of the models discussed in Chapter 2, Section 2.8. After reviewing the literature, it was concluded (in Section 2.8) that some key concepts and properties need to be considered for developing a quality requirement model. These concepts represent the criteria followed in the above comparison. Table 4-3 shows that both the OntoSoS.QR model and Zhou model (Zhou et al., 2005) provide a support for a large set of QRs. However, Zhou model does not provide concepts that are related to roles, SoS and QR main attributes.

Furthermore, both the OntoSoS.QR model and the Tran model (Tran et al., 2009) are the most comprehensive models. However, Tran model does not support the concepts related to SoS and conflicts. For example, it does not include any relations to connecting QR to all related QRs to detect any conflicts. Also, no concepts regarding any other systems involved in the SoS arrangement (i.e. constituent system of the SoS).

Also, the classification and the hierarchy of the classes of the QRs are checked for conformed with the ISO/IEC 25010 standards. The same hierarchy of the quality requirements and their sub- quality requirements were apparent in the OntoSoS.QR.

The second step of evaluation was to check the consistency of the concepts of the QR model. The reasoner tool HermiT 1.3.8 (“HermiT 1.3.8,” 2013), which is free and compatible with Protégé 4.3 was applied to the ontology model to report the consistency between the concepts. After this, semi-structured interviews (Bergin and Stokes, 2006) and walkthroughs techniques (John and Packer, 1995) were used to verify the structural correctness and consistency between the concepts of the OntoSoS.QR ontology model.

Table *4-4* reflects the main questions and the check list conducted with domain experts. This table was designed to enable domain experts to walkthrough all the concepts of the OntoSoS.QR model and their relations to reflect the satisfaction of domain experts regarding the degree of the structural correctness and consistency of the model. Domain experts checked the definitions provided in Table 4-2 to clarify any ambiguity. Then, they validated the OntoSoS.QR model in Figures Figure *4-1*, Figure *4-2*, Figure *4-3*, and Figure *4-4*.

Table 4-4:Checklist to Validate the Quality Requirements Ontology Model Using Walkthrough Approach

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
1	Validate the concept "Quality Requirements" and its Data Properties (DP) and Object Properties (OP).	Concept: QualityRequirements and its sub-concepts (i.e. concepts 2- 25 in Table 4-2.	*		
		DP: name	*		
		DP: purpose	*		
		DP: version		*	No need
		DP: description	*		
		OP: has_Relationship	*		
		OP: relatedToGroup	*		
		OP: appliedToService	*		
		OP: hasMandatory	*		
		OP: hasWeight		*	Not applicable to all QRs (related to concept 10)
		OP: hasTendency		*	Not applicable to all QRs (related to concept 11)
		OP: hasValidPeriod		*	Not applicable to all QRs

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
					(related to concept 12)
		OP: hasMetric	*		
2	Validate the concept “External System” and its properties	Concept: ExternalSystem	*		
		DP: name	*		
		OP: provide Full Of/Part Of	*		Connected to service
3	Validate the concept “SoS Level” and its properties	Concept: SoS Level	*		
4	Validate the concept “Constituent System” and its properties	Concept: Constituent System	*		
5	Validate the concept “Non-Constituent System” and its properties	Concept: NonConstituent System	*		
6	Validate the concept “Role” and its properties	Concept: QoSRole and its sub-concepts.	*		
		DP: RoleName	*		
		OP: specify Mandatory/Weight/Tendency/Valid period	*		
		OP: belong to system	*		
7	Validate the concept “Provider” and its Properties		*		
8	Validate the concept “Authority” and its Properties		*		
9	Validate the concept “QoS Mandatory” and its Properties	DP: mandatory Value	*		
10	Validate the concept “QoS Weight” and its Properties	DP: weight Value	*		Check concept 1
11	Validate the concept “QoS Tendency” and its Properties	DP: tendency Value	*		Check concept 1
12	Validate the concept “QoS Valid Period” and its Properties	DP: start Date	*		Check concept 1
		DP: end Date	*		
13	Validate the concept “QoS Group” and its Properties	DP: name	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
		DP: Description	*		
		OP: is_A	*		
14	Validate the concept “QoS User Group” and its Properties		*		
15	Validate the concept “QoS System Group” and its Properties		*		
16	Validate the concept “QoS SoS Group” and its Properties		*		
17	Validate the concept “Service” and its Properties	DP: name	*		
		DP: description	*		
18	Validate the concept “Non-Achieved Service” and its Properties	DP: name	*		Rarely
		DP: description	*		
19	Validate the concept “Achieved Service” and its Properties	DP: name	*		
		DP: description	*		
20	Validate the concept “Constraint” and its Properties	DP: name	*		
		DP: description	*		
		OP: is_A	*		
21	Validate the concept “Architectural Constraint” and its Properties	DP: architectural type	*		
22	Validate the concept “Cost Constraint” and its Properties		*		Not always a constraint
23	Validate the concept “Operator Constraint” and its Properties		*		
24	Validate the concept “Contract Constraint” and its Properties		*		
25	Validate the concept “Schedule Constraint” and its Properties		*		
26	Validate the concept “QoS Metric” and its Properties	DP: name	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
		DP: description	*		
		OP: is_A	*		
		OP: has Value	*		With QoS Metric Value
		OP: metric has Constraint	*		With Constraint
27	Validate the concept “QoS Metric Value” and its Properties	DP: metric Value	*		
		DP: metric description	*		
28	Validate the concept “Qualitative Value” and its Properties	DP: satisfying Status	*		
29	Validate the concept “Quantitative Value” and its Properties	DP: quantitative Value	*		
		OP: has Unit	*		
30	Validate the concept “Unit of Metric” and its Properties	DP: name	*		
		DP: description	*		
		OP: is_A	*		
31	Validate the concept “Quantity Unit” and its Properties		*		
32	Validate the concept “Currency Unit” and its Properties		*		JD (Jordanian Dinar)
33	Validate the concept “Time Unit” and its Properties		*		
	Please list any new suggested concepts here.				

In order to carry out the third evaluation step, the satisfaction of experts regarding the correctness and completeness of the representation of the QR model in the case study context, they validated the representation of this model in Figures Figure *4-6* and Figure *4-7*. To report a sufficient and complete representation of the QR model, several test cases were required to cover all the concepts of the model. One test case of a specific QR will not be able to cover all main concepts and relations of a model. For example, the safety QR (e.g. safe discharge of patient from hospital) can only have a qualitative metric as a satisfying status. So, another QR is required to cover the quantitative metric of the QR. For example, the response time QR (e.g. deliver medical records to care providers in a timely manner) has a quantitative value which differs from system to another depending on the task provided. Table 4-5 provides the details of the test cases coverage. It shows that a subset of the testing cases covers a wide range of the OntoSoS.QR concepts.

Table 4-5: Testing Cases Coverage Matrix

Concepts	Quality Requirement	System Group Related	SoS Group Related	User Group Related	Service Applied	Valid Period	Tendency	Weight	Mandatory	Role	Quantitative Metric	Qualitative Metric	Constraint	Exist of other related QR
Test 1	Safe Discharge	√	√	-	√	√	√	-	√	√	√	-	√	√
Test 2 (More details in Appendix II)	Response Time (deliver medical record)	-	√	√	√	√	√	√	√	√	-	√	√	√

The key questions and outputs of the interviews conducted with experts from the case study illustrated in Table 4-6.

Table 4-6: The Outcomes of the Main Questions of the Interviews Conducted with Domain Experts to Validate SoS Quality Requirements Model - Increment 1

	Questions	**Notes**
1	Do you agree that the model concepts and associated properties **completely** represent the KHCC's quality requirements- related concepts ?	√
2	Are there any missing concepts or properties?	No
3	Are there any other properties that could enhance the model?	A suggestion to include the currency of the JD "Jordanian Dinner" as a currency unit value (concept 32 from Table 4-5) .
4	Are there any extra concepts or properties -that would be better if they been removed from the model?	A suggestion to remove the DP "version" of concept 1 from Table 4-5.
5	Do you agree that the concepts and associated properties are **correctly** represent the quality requirements-related concepts?	√
6	Are the data types that used to describe the Data Properties, correct?	√
7	Are the relations (Object Properties) that used to connect the concepts of the quality requirements model, correct?	√

	Questions	Notes
8	Are the domain and the range for the relations b/w the concepts correct?	✓
9	Are the cardinality constraints of the relations correct?	Some suggestions provided regarding the cardinalities of three relations between concept 1 and concepts 10, 11 and 12 from Table 4-5.
10	Do you agree that the above concepts and associated properties are **consistent (free of contradiction with other components in the ontology)** with each other?	✓
11	Do you agree with definitions and the classification of the quality requirements following the ISO/IEC 25010 standards? As in Table 4-2 and Figure 4-6.	✓
12	Is it possible to check the adherence to quality requirements governance in the SoS arrangement using the OntoSoS.QR?	The constraints affecting the quality values need to be enforced by policies specific to the services connected to each QR.

As can be observed in Table 4-6, the main concepts of the OntoSoS.QR model were satisfied by experts. However, only few relations and cardinalities were not fully satisfied which are the relations between concept 1 and concepts 10, 11, and 12. The relations are 'QR has 1 or more Tendency', 'QR has 1 or more Weight' and 'QR has 1 or more valid period'. Interviewees said that these relations are not mandatory and no need to be applied to all QRs. For example, in the case of the tendency relation, it is specified that tendency values of the QR in relation to its measure value is member of the list (see Table 4-2) {exact, close, negative (lower value is better), positive (higher value is better)}. Interviewees agreed that it is not a must to have a specific value from the list. Also, experts indicated that no need for the property 'version for a QR'.

Based on the evaluation conducted for the first DSRM process increment, the following results were concluded:

i) The evaluation criteria (i.e. verifying the structural correctness, concepts consistency and validate the completeness of case study representation) have been mostly satisfied. The walkthrough and check list of the OntoSoS.QR ontology model showed that the demonstration of quality requirements from case study was extensively complete and correctly captured by the ontology. Also, all concepts are structurally and logically consistent (i.e. free of contradiction with other components in the ontology), which was also confirmed by applying consistency check using reasoner tool installed with Protégé editor. (The results of the evaluation support answering RQ4).

ii) Reusing and extending some existing ontologies in the literature benefited the building of the OntoSoS.QR ontology model. A comparison was conducted (Qaddoumi et al., 2018a) between OntoSoS.QR and other QR models from literature, which showed that OntoSoS.QR model is comprehensive and combine the strengths of some existing ontologies (e.g. examples of ontologies adapted from Table 4-3). This reflects that using ontology is suitable to define all related aspects in the quality requirements model. (This supports answering RQ2).

iii) It was shown in Table 4-6, point 12, that the constraints affecting the quality values to deliver high quality services need to be enforced by policies specific to the services connected to each QR. However, these concepts are part of the policy model which has been chosen to be linked to the OntoSoS.QR model and developed in the second DSRM iteration.

iv) Some minor amendments were suggested by domain experts to enhance the OntoSoS.QR model. Section 4.5 reflects these suggestions.

4.5. Revisiting Increment 1

After the validation of the OntoSoS.QR model, only minor amendments were conducted to produce the second version of this model (depicted and highlighted in Figure 4-9).

Some object properties (i.e. relation) are slightly changed. For instance, the relation ‘QR has 1 or more Tendency’ has been changed to ‘QR has 0 or 1 Tendency’. According to Table 4-4, domain experts agreed that it is not necessarily to have always a QR that has a tendency value. So, the cardinality rule is changed from mandatory to optional to have a minimum value of 0 instead of 1. The same amendment is applied to another two relations (i.e. QR has a weight and has a valid period). Another example is omitting the data property ‘version’ of ‘Quality Requirement’ class.

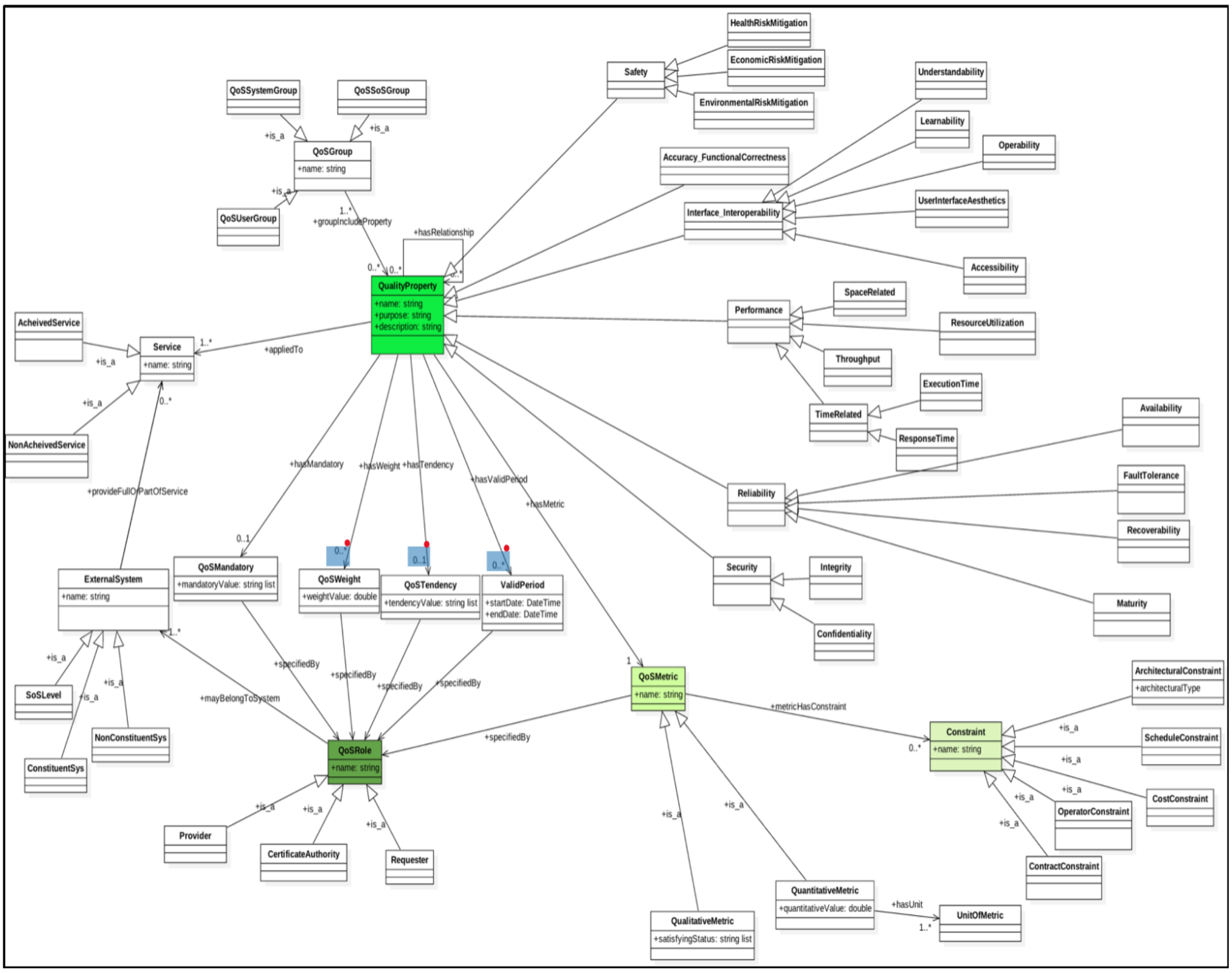


Figure 4-9: The Conceptual Representation of Version 2 of the OntoSoS.QR model

By the end of DSRM iteration 1, Table 4-7 provides an overview of the status towards answering the research questions. The tick symbol (✓) shows that the research question/sub-question has been answered, and the hourglass with flowing sand symbol (⏳) shows that the status of answering the research question/sub-question still in progress and the question is partially answered.

Table 4-7:Status towards Answering the Research Questions

RQ	Main RQ and RQs-Concerns (sub-questions)	Status	Notes	Chapter
RQ1	*What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*	✓		2
1.1	Can research gap analysis, by surveying the literature, identify these challenges ?	✓		2
RQ2	*How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*	⏳		2, 4,5,6 & 7
2.1	What are the components of the quality governance framework, and what are the initial specifications of each component ?	✓		2
2.2	Is ontology suitable to define all related aspects to quality governance ?			7
2.3	What are the elements/concepts of the ontology? Are there any ontologies that we can reuse ?	⏳	See Tables Table 4-1, Table 4-2.	4,5,6 & 7
RQ3	*How will the semantically-enriched models of policies, processes, standards and quality requirements interact in the systems of systems context to identify and resolve semantic heterogeneities?*			5,6 & 7
3.1	Can we develop a process to detail the interaction between *policies, processes, standards and quality requirements to identifying and resolve semantic heterogeneities* ?			5,6 & 7
3.2	Are there any limitations of using an ontology-based approach that restrict the interactions between *policies, processes, standards and quality requirements* ?			5,6 & 7
RQ4	*How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*	⏳		4,5,6 & 7
4.1	How the OntoSoS.QM.Gov ontology will be assessed ?		General evaluation process presented in Chapter 3 only	7
4.2	Can we validate each ontology model (i.e. each component of the quality governance framework) and then to validate the whole quality governance ontology framework ?	⏳	See Section 4.4.	4,5,6 & 7

4.6. Summary and Conclusion

Reviewing the literature, many QR models exist. However, many of them were reused and combined to build the OntoSoS.QR ontology model in the SoS context. The quality requirements taxonomy adopted in the QR ontology model is consistent with the ISO/IEC 25010 standard.

In order to identify both limitations and strengths of the available QR models in literature, a research gap analysis was conducted (see Section 2.8) which resulted in identifying a list of key concepts and properties that need to be considered while constructing the QR ontology model and to contrast other quality requirements models. The QR ontology model was evaluated with by related domain experts at KHCC, with their feedback applied. However, only minor amendments were required to be reflected on the model.

This chapter reflected the first DSRM iteration. It partially answered research questions RQ2 and RQ4. The main concepts of the first part of OntoSoS.QM.Gov were identified and a semantically -enriched model was created using ontology. It adopted several concepts from existing ontologies in the literature and brainstorming new concepts to extend the functionality of deploying the OntoSoS.QR model in the SoS context. This ontology was evaluated by comparing it to existing ontologies. Then it was evaluated using data from the case study to facilitate testing the semantics and the structure of the ontology. The correctness, completeness and consistency of the ontology were assessed which appeared satisfied by the domain experts.

After revisiting the model, the first DSRM iteration was ended and the updated version of the model is used to interact with the policy model to act as an input to the second DSRM iteration which will be discussed in chapter 5.

In order to fully address RQ2 and RQ4, other DSRM iterations will be conducted to address other OntoSoS.QM.Gov framework components.

In order to check the adherence to quality governance, it is appeared that the concepts of the QR model are not sufficient to check such adherence (as noted in Table 4-6, point 12), this is due to lack of the concepts related to the constraints affecting the quality values and the services related to the quality requirements. However, these concepts are part of the policy model which has been chosen to be linked to the OntoSoS.QR model and developed in the second DSRM iteration.

5. Chapter Five: - The Policies Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment II)

5.1. Introduction

This chapter details the second increment of the DSRM process that was followed in this research. Firstly, the design of the policy model (i.e. the OntoSoS.Policy model) is described in detail. Then, the interaction between the previously developed model (i.e. the OntoSoS.QR model) with the new OntoSoS.Policy model is described, followed by the demonstration of the developed components (i.e. Increment 2) using a case study from the cancer care domain. After this, an evaluation process has been conducted, and the resulting feedback from the evaluation process was used to revisit the design of Increment 2.

It has been discussed earlier in Chapters 2 (Section 2.7) and 3 (Section 3.8), the ontology construction process adopted to design the ontology models of the constituent four models of the OntoSoS.QM.Gov framework, i.e. the quality requirements, policy, processes and standards models respectively. This method is depicted in **Error! Reference source not found.** and **Error! Reference source not found.**.

The next section details the construction process of the policy model (i.e. OntoSoS.Policy) and the linkage between it and the OntoSoS.QR model. Then, the demonstration of Increment 2 using data from the research case study is described in Section 5.3. After this, the validation process followed is described in Section 5.4. Finally, Section 5.4 details the feedback reported by the domain experts, which was used to revisit the policy model of Increment 2.

5.2. The Design of the OntoSoS.Policies Ontology Model Linked to Increment 1: Increment 2

To develop the OntoSoS.Policy model, the domain and the scope needed to be determined. In general, business policies employ functional or non-functional requirements (i.e. system's qualities) (Phan et al., 2008). In this research, the focus is on the policies that can be related to one or more quality requirements. This needs to be taken into consideration during the demonstration process as well.

Secondly, as illustrated in Chapter 2, Section 2.9, many of the ontologies in the literature could be extended to include further concepts making them fit for the present research. So, several existing policy models were studied with the purpose of reusing or extending some of them to elicit the best characteristics and concepts presented by each one to be used for the semantic representation of the policies for SoS.

Thirdly, the key terms (i.e. concepts) for a policy need to be specified. Garcia and Toledo (2008) stated that policy is “a collection of alternatives and each alternative is a collection of assertions”. Assertions specify characteristics that are critical to service selection and use, for instance, security characteristics. Also, they suggested some concepts that each policy needs to contain: policy reference, service, policy alternatives and policy assertions (Garcia and Toledo, 2008).

The authors in (Lee et al., 2002) indicated that a policy model needs to include the following aspects: (1) organizational aspects, e.g. ‘Role’, (2) process aspects, e.g. task hierarchy structure such as parent task, and (3) product aspects, e.g. artefacts. Also, a list of desired general and service-specific properties for a policy-based frameworks are introduced in (Schneider et al., 2008), include the following directives: (1) policy specification needs to include the actions used to execute the policy; (2) policy needs to be easily extensible to support many types of policies; (3) the same domain policies needs to be grouped together; (4) distributed policy enforcement needs to be considered to increase the scalability; (5) rules and constraints (e.g. timing constraints) need to be considered; (6) one should stay at a reasonable level of abstraction so that systematic policy requirements can be specified independently of the implementation details; (7) one should connect to the processes of the service or system lifecycle to detect a change; (8) dynamic policy updates should be supported; and (9) the ability to derive individual services’ policies from the policy of a service orchestration and vice versa should be supported to automatically codify policies. In this research, the main policy-related concepts of the ontology, depending on the definitions proposed in literature (Damianou et al., 2001; Garcia and Toledo, 2008; Payne and Metzler, 2005; Phan et al., 2008; Schneider et al., 2008; Snir et al., 2003).

Also, the ontology policy model was extended to involve the capability of supporting SoS related aspects e.g. the concepts of SoS top level, constituent system CS, etc. By doing this, the policy ontology model combines the strengths of existing policy ontologies and extends the capability of supporting SoS- related concepts. The details of the employed concepts in the policy ontology model and their traceability to the literature are provided in Table 5-1.

Fourthly, after determining the main terms, they need to be organized to form a class hierarchy to have further sub-classes. For instance, the main concept ‘Role’ will have three sub-classes: Staff, Manager and Customer/Patient. Figure 5-1 shows the conceptual representation of the OntoSoS.Policy model.

The next two steps are related to identifying the data properties and object properties (i.e. slots) of the classes as well as identifying any restrictions on these slots. Garcia and Toledo (2008) suggested some data attributes that each policy needs to contain, e.g. name and ID. Also, as illustrated in (Noy and McGuinness, 2001), the restrictions to be identified in relation to the types of the slots values (e.g. string, number, Boolean, etc.), slot cardinality to show how many values a slot and a domain can have. Some objects properties (i.e. relations) were suggested, e.g. 'constraint_ By' as it is a recursive relation to the 'policy concept'. For instance, the admission-related policies are constraint by the registration policy.

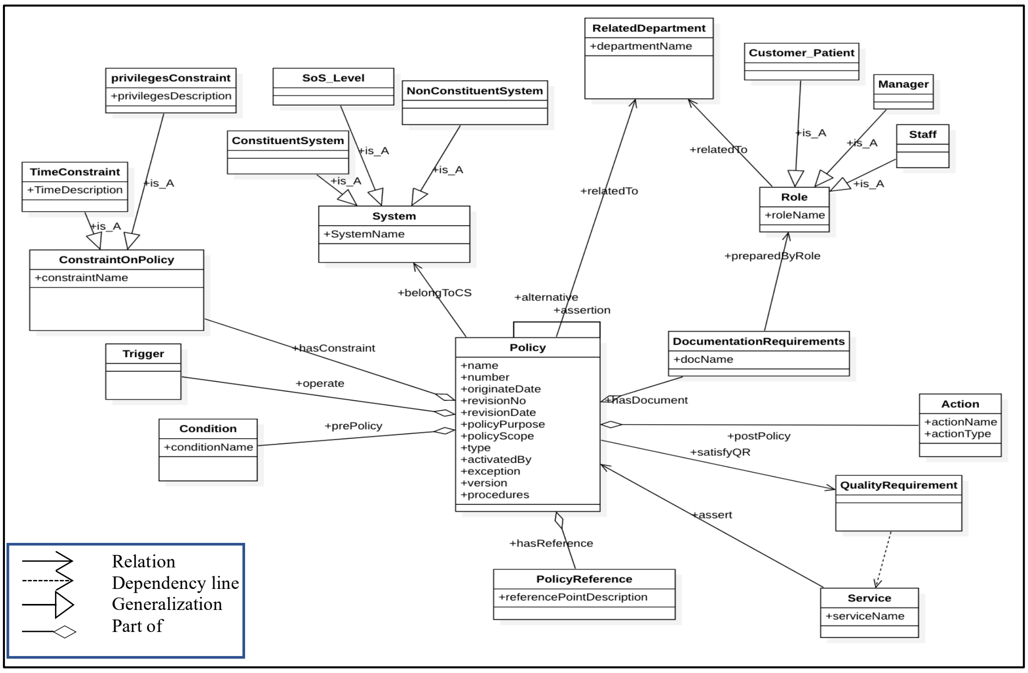


Figure 5-1: The Conceptual Representation of Version 1 of the OntoSoS.Policy Model

Table 5-1: OntoSoS.Policy Main Concepts and Traceability to Literature Sources

	Concept	Main Sources from Literature and Description	Properties and their types
1	Policy	The root element that indicates a policy (Garcia and Toledo, 2008)	-**name**: string. The name of the policy
			-**number**: string. The unique number of the policy
			-**originateDate**: Date. The originate date of the policy
			-**revisionNumber**: number. The revision number of a policy
			-**revisionDate**: Date. The date on which the policy is revised
			-**policyPurpose**: string. The main aim of the policy
			-**policyScope**: string. The scope of the policy: the related departments
			-**policyType**: string. The type of the policy
			-**exception**: string. The exception/situation that the policy is not followed

	Concept	Main Sources from Literature and Description	Properties and their types
			-**version**: string. The version of the policy
			- **description**: string. The description of a policy
			-**has_alternative** of type Policy. Relation b/w two policies: The policy that is alternate another policy
			-**constraintBy** of type Policy. Relation b/w two policies: The policy that constraint deploying another one
			-**assert/interactWith** of type Policy. Relation b/w two policies: The policy that interact with or assert to the policy
			-**hasDocument** of type Documentation-Requirement. Relation b/w a policy and document: The documents/forms that are used and need to be filled according to a policy
			-**satisfyQR** of type Quality-Requirement. Relation between a policy and QR: The quality requirements that is supposed to be fulfilled -if any- when deploying a policy
			-**hasReference** of type Policy Reference. Relation between a policy and a Policy Reference: The policy that has a reference point that refer to another policy
			-**belongToSystem** of type System. Relation b/w a policy and a constituent system of the cancer SoS arrangement
			-**postPolicy** of type Action. Relation b/w a policy and an action.
			-**approvedBy** of type Approving-Committee. Relation between a policy and an Approving- Committee
			-**responsibilityOf** of type Related-Department. Relation between a policy and department. The main department that responsible of deploying the policy
2	Policy- Reference	This element may be used to include the content of a policy into another policy (Garcia and Toledo, 2008)	-**name**: string. The name of the policy reference
			-**assert** of type Policy. *see (-**hasReference**) above in line1
3	Service	This element to describe details of the service implementation for which the policy has been specified (Garcia and Toledo, 2008) (Schneider et al., 2008)	-**name**: string. The title of the service
			-**description**: string. The description of the service
			-**performedBy** of type Role. Relation between a service and a role: the main Role(s) that have the responsibility of performing the service
4	Constraint- On-Policy	This element to describe the constraints on the policies such as timing constraints and prioritization constraints (Schneider et al., 2008)	-**name**: string. The title of the constraint
			-**hasConstraint** of type Policy. Relation between a constraint and a policy
5	Condition	A policy is in the form of *if conditions then actions* rules in which *conditions* are a set of expressions. Policy can be defined as an abstract class which contains a set of abstract *Condition* classes and a set of abstract *Action* classes. (Snir et al., 2003); (Schneider et al., 2008); (Uszok et al., 2004); (Payne and Metzler, 2005))	-**name**: string. The name of the condition
			-**prePolicy** of type Policy. Relation between a condition and a policy

	Concept	Main Sources from Literature and Description	Properties and their types
6	Action	*Actions* are a set of actions that must be executed eventually (regardless of order) (Snir et al., 2003); (Schneider et al., 2008); (Uszok et al., 2004); (Payne and Metzler, 2005)	**-description**: string. The description of the action **-postPolicy** of type Policy. Relation between an action and a policy
7	Trigger	Action trigger points that call the policy to be implemented (Payne and Metzler, 2005)	**-name**: string. The name of a trigger **-operate** of type of Policy. Relation between a trigger and a policy
8	Related-Department	All departments in the system that are to be affected by deploying a policy. Suggested by domain experts from KHCC.	**-name**: string. The name of the department
9	Approving-Committee	The committees that provide a formal approval of a policy. Suggested by domain experts from KHCC.	**-name**: string. The name of the committee
10	Documentation-Requirement	The documents that need to be filled by a role according to the specification of a policy.	**-name:** string. The title of the document **-preparedBy** of type Role. Relation b/w a document and a role
11	Quality-Requirement *(Refer to QR model)	The related quality requirements that need to be satisfied by deploying a policy. (Garcia and Toledo, 2008); (Phan et al., 2008))	**-name**: string. The title of the QR *see (-**satisfyQR**) above in line 1
12	Role	The roles in a system that are involving with deploying a policy. (Payne and Metzler, 2005)	**-name**: string. The title of the role *see (-**preparedBy**) above in line10 **-relatedTo** of type RelatedDepartment. Relation b/w a department and a role
13	System	A policy can belong to the SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement. System: (Schneider et al., 2008)), Domain experts and brainstorming to elicit SoS and CS concepts.	**-name**: string. The name of the constituent system
14	SoS	Brainstorming of concepts: A policy can belong to a higher level i.e. SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement.	**-name**: string.
15	CS	Brainstorming of concepts: A policy can belong to a higher level i.e. SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement.	**-name**: string.

As seen from both Figure 4-9 and Figure 5-1, both models are connected together using ontology. Software quality requirements need to be consistent with their related policies (Robinson, 2006). A policy needs to satisfy one or more quality requirements (see Figure 5-2).

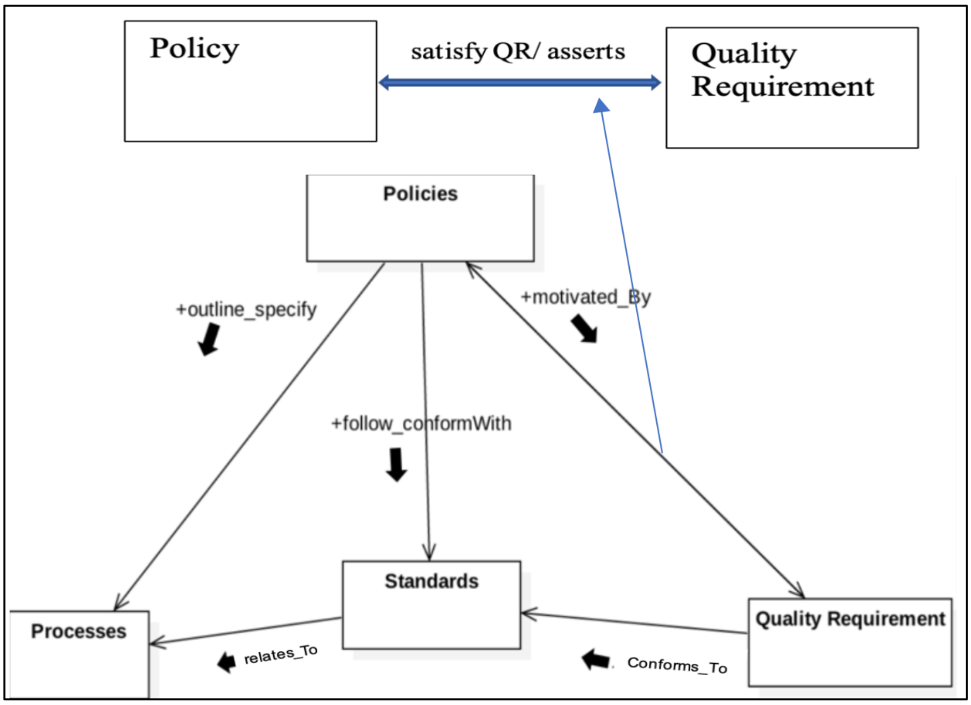


Figure 5-2: Abstract Level of the Proposed Quality Governance Model with Focus on the Link between the Quality Requirements model and the Policy model

The two ontology components (i.e. policy and QR models) represent the second increment of the OntoSoS.QM.Gov. If a policy has the purpose of satisfying a certain quality requirement, then the policy specifications conform with the specification of the quality requirement CSs. Figure 5-3 shows the linking relations between both of them (coloured in blue). For instance, the conformance between time constraints, properties such as mandatory and tendency, roles and services (see the concepts' definitions in Chapter 4, Section 4.2).

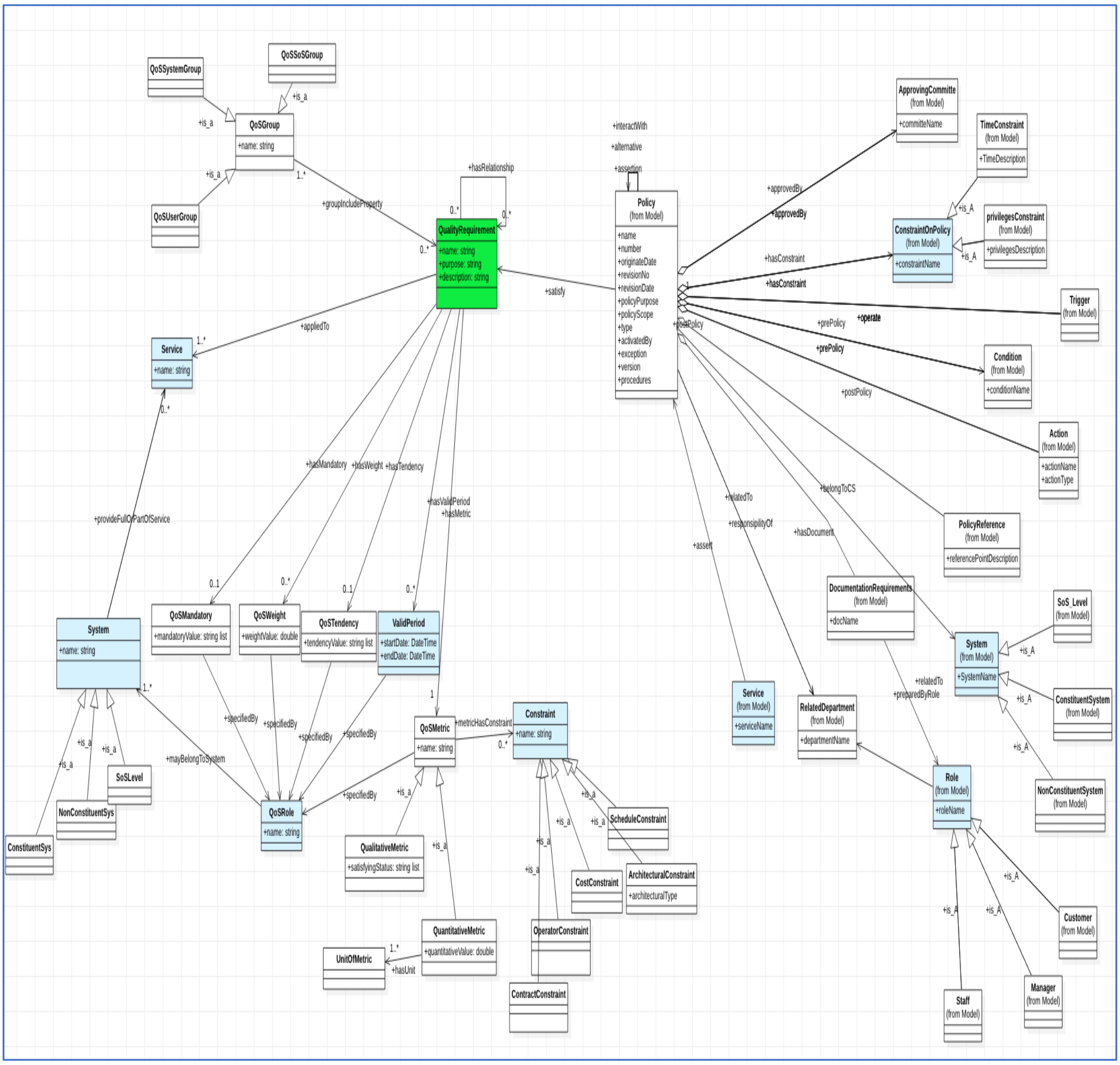


Figure 5-3: Conceptual View of Increment 2

As depicted in Figure 5-3, both models QR and policy are linked via the relation 'satisfies', in order to check the adherence to quality requirements as well as for policies. The process to indicate the interaction between both models is described in both Algorithms Algorithm 5-1 and *Algorithm 5-2*.

Algorithm 5-1: Interaction between the SoS Policies and Quality Requirements Models (Alternative 1:QRs-Policies)

```
Input:
$QR_i$ = The quality requirement that we need to check governance for.
Output:
CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies $Po_j$ that are related to $QR_i$ in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number of all policies in each CS.
Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each CS; and $0<x<=s$, c=number of all services in each CS.
CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, $CS_n.QR_q.Properties_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$, r=number of all Properties in each QR.
Begin
  /*Step 1: Find all policies $Po_j$ that are related to $QR_i$*/

  For each constituent system $CS_i$ in all CS do
      For all policies $Po_j$ in each CSi do
          Find all policies $CS_i.Po_j$, where $CS_i.Po_j$ <<has>> $Po_j$. 'policy purpose' = $QR_i$ or $QR_i$ that <<have Relation with>>$QR_m$
          Add Both $Po_j$ from previous point and the interacted Policy $CS_i.Po_j$ <<interact with>> $CS.Po_i$ to the policy set CS.Policy_Set = {$CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, $CS_2.Po_2$, …, $CS_i.Po_j$};
      End for
  End for

  /*Step 2: Check the conformance of QR to policies*/
  For each policy $Po_j$ in CS.Policy_Set that resulted from Step 1 do
      {
      if QR.Service: 'service_name': Service  exists in Policy. Service: 'service_name',
        then {
              /*Compare*/
              {
              if (QR. Constraint. 'constraint_name') does not match ($Po_j$. Constraint.'constraint_name')
          OR
              if  (QR. Valid period. 'start date' and 'end date') does not match ($Po_j$. Time constraint.'time')
          OR
              if (QR. QoS.Role.'Role name') does not match ($Po_j$. Role.'name')
              }
              then
              add conflicts results to Conflicts_Poicy_QR_Set.
              add additional information {QR. Properties (mandatory, valid period, tendency, weight)}
              to CS_QR.Properties_Set.
          }
      }
  End for
End
```

Algorithm 5-2: Interaction between the SoS Policies and Quality Requirements Models (Alternative 2: Policies-QRs)

```
Input
$Po_i$ = The Policy that we need to check governance for.
Output:
QR_Set= {$CS_1.QR_1$, $CS_1.QR_2$, …, $CS_i.QR_j$, …, $CS_n.QR_q$), where $0 < i <= n$, n=number of CSs, $0<j<=q$,
    q=number of all QR in each CS.
CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies $Po_j$ that
    are related to $QR_i$ in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number
    of all policies in each CS.
Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … ,
    $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles,
    constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$,
    n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each
    CS; and $0<x<=s$, c=number of all services in each CS.
CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, $CS_n.QR_q.Properties_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$,
    r=number of all Properties in each QR.
Begin
/*Step 1: Find all interacted policies $Po_j$ to the entered policy (if any) and that are related to QR*/

  For each constituent system $CS_i$ in all CS do
      For all policies $Po_p$ in each CSi do
      For all QR  QR j in each $CS_i$ do
              Find all QRs $CS_i.QR_j$, where $CS_i.QR_j$  that <<haveRelation with>> $CS_i.QR_n$
              add Both $CS_i.QR_j$  and $CS_i.QR_n$ from previous step to the QR set QR_Set=
              {$CS_1.QR_1$, $CS_1.QR_2$, …, $CS_2.QR_1$, $CS_2.QR_2$, …, $CS_i.QR_j$).
      End for
  End for

/*Step 2: Check the conformance of QR to policies*/
 For each QR $QR_j$ in QR_Set that resulted from Step 1 do
     Execute Algorithm 5-1.
End for
End
```

Algorithm 5-1 embodies the process that informs if all policies in the SoS's constituent systems conform with a particular QR, while Algorithm 5-2 represents the process to detect if all QRs in the SoS's constituent systems conform with a particular policy. These two algorithms represent the two possible alternatives as the relation between both models is bidirectional one. So, the relation 'satisfies' between QR and policy and vice versa is checked for satisfaction. If any policy or QR satisfies this relation, then they need to be selected to be in the policy set 'Po_Set' or QR set 'QR_Set'. Then, the values of the main similar aspects, e.g. time constraints, roles titles, services related to the policies or QRs in the set will be compared with the entered values of the QR and policy. Also, because there are iterative relations 'QR has relation with another QR' and 'Policy has relation with another Policy' (see Figure 5-3), more QRs and policies will be included in the process.

The output of executing the algorithms will be a list of conflicts along with the related CSs and related aspects, e.g. the properties of the QR to allow for further prioritisation and compromising between QRs or policies (if required). Examples demonstrating the application of Algorithms 5-1 and 5-2 are illustrated in section 5.3.

5.3. The Demonstration of Increment 2

In order to demonstrate increment 2, an example from the research case study was enacted for this demonstration. Figure 5-4 shows the connection between the policy ‘Emergent Admission of patients to the centre’ and the ‘performance QR: resource utilisation’ QR.

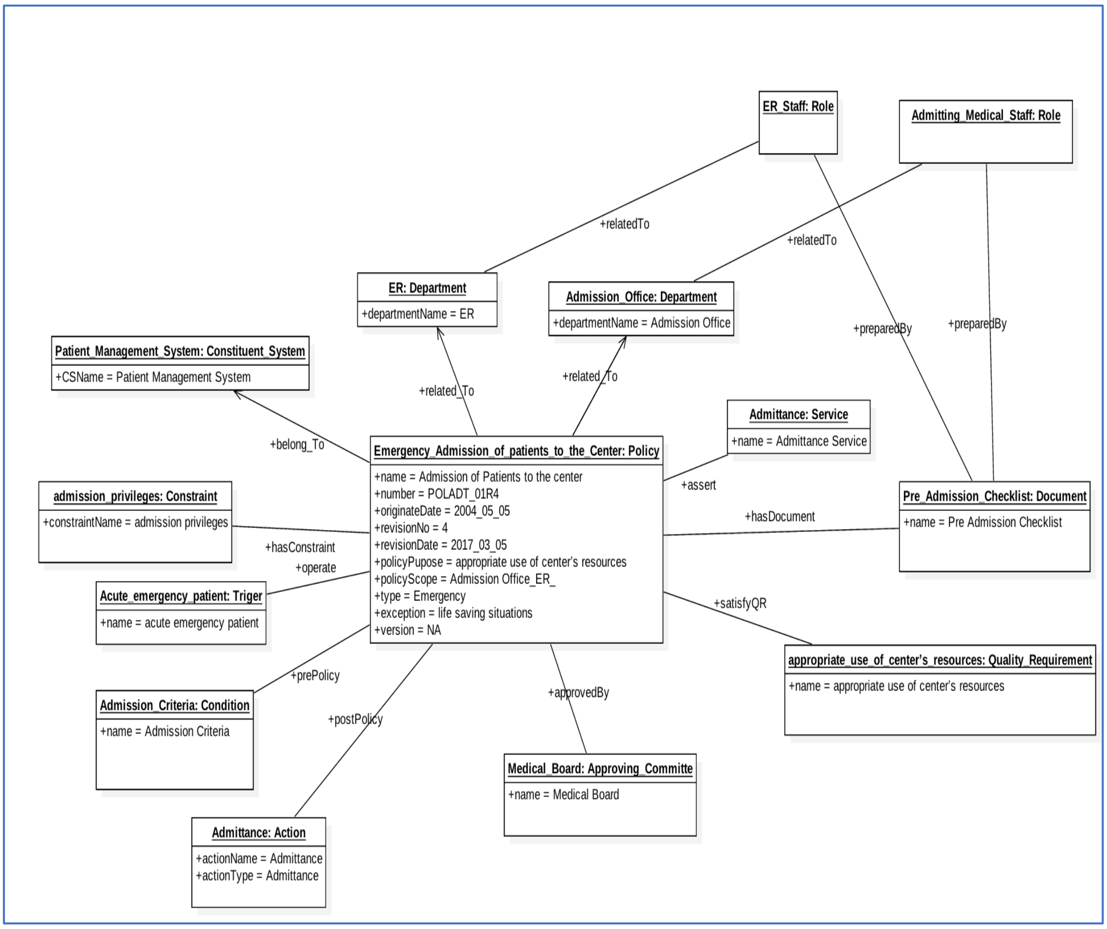
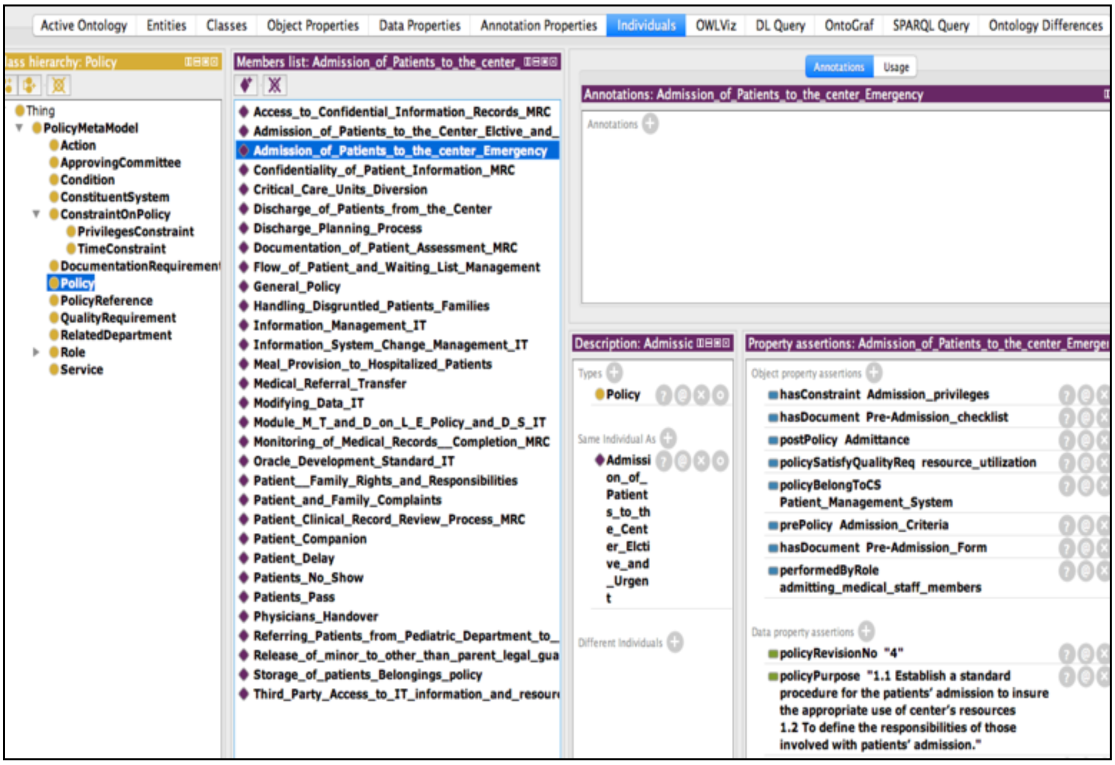

ER_Staff: Role
Admitting_Medical_Staff: Role
+relatedTo
+relatedTo
ER: Department
+departmentName = ER
Admission_Office: Department
+departmentName = Admission Office
+preparedBy
+preparedBy
Patient_Management_System: Constituent_System
+CSName = Patient Management System
+related_To
+related_To
Admittance: Service
+name = Admittance Service
+belong_To
Emergency_Admission_of_patients_to_the_Center: Policy
+name = Admission of Patients to the center
+number = POLADT_01R4
+originateDate = 2004_05_05
+revisionNo = 4
+revisionDate = 2017_03_05
+policyPupose = appropriate use of center's resources
+policyScope = Admission Office_ER_
+type = Emergency
+exception = life saving situations
+version = NA
admission_privileges: Constraint
+constraintName = admission privileges
+assert
+hasDocument
Pre_Admission_Checklist: Document
+name = Pre Admission Checklist
+hasConstraint
+operate
Acute_emergency_patient: Triger
+name = acute emergency patient
+satisfyQR
appropriate_use_of_center's_resources: Quality_Requirement
+name = appropriate use of center's resources
+prePolicy
+approvedBy
Admission_Criteria: Condition
+name = Admission Criteria
+postPolicy
Medical_Board: Approving_Committe
+name = Medical Board
Admittance: Action
+actionName = Admittance
+actionType = Admittance

Figure 5-4: Policy Object Model - Emergency Admission of Patients in KHCC Policy Model



lass hierarchy: Policy

- Thing
  - PolicyMetaModel
    - Action
    - ApprovingCommittee
    - Condition
    - ConstituentSystem
    - ConstraintOnPolicy
      - PrivilegesConstraint
      - TimeConstraint
    - DocumentationRequirement
    - Policy
    - PolicyReference
    - QualityRequirement
    - RelatedDepartment
    - Role
    - Service

Members list: Admission_of_Patients_to_the_center_

- Access_to_Confidential_Information_Records_MRC
- Admission_of_Patients_to_the_Center_Elctive_and_
- Admission_of_Patients_to_the_center_Emergency
- Confidentiality_of_Patient_Information_MRC
- Critical_Care_Units_Diversion
- Discharge_of_Patients_from_the_Center
- Discharge_Planning_Process
- Documentation_of_Patient_Assessment_MRC
- Flow_of_Patient_and_Waiting_List_Management
- General_Policy
- Handling_Disgruntled_Patients_Families
- Information_Management_IT
- Information_System_Change_Management_IT
- Meal_Provision_to_Hospitalized_Patients
- Medical_Referral_Transfer
- Modifying_Data_IT
- Module_M_T_and_D_on_L_E_Policy_and_D_S_IT
- Monitoring_of_Medical_Records__Completion_MRC
- Oracle_Development_Standard_IT
- Patient__Family_Rights_and_Responsibilities
- Patient_and_Family_Complaints
- Patient_Clinical_Record_Review_Process_MRC
- Patient_Companion
- Patient_Delay
- Patients_No_Show
- Patients_Pass
- Physicians_Handover
- Referring_Patients_from_Pediatric_Department_to_
- Release_of_minor_to_other_than_parent_legal_gua
- Storage_of_patients_Belongings_policy
- Third_Party_Access_to_IT_information_and_resour

Annotations: Admission_of_Patients_to_the_center_Emergency

Annotations

Description: Admissic

Types
Policy

Same Individual As
Admission_of_Patients_to_the_Center_Elctive_and_Urgent

Different Individuals

Property assertions: Admission_of_Patients_to_the_center_Emerge

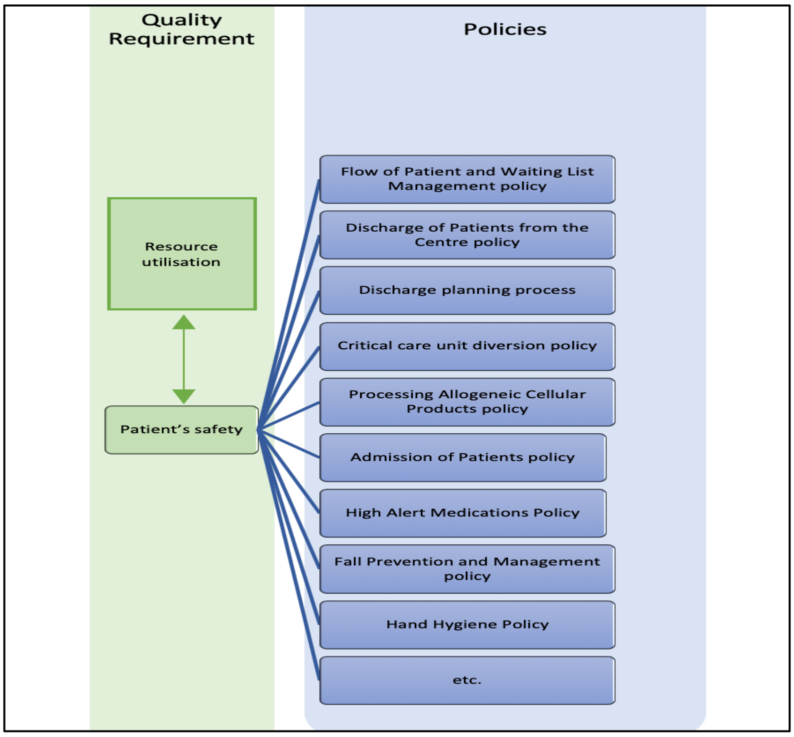
Object property assertions
hasConstraint Admission_privileges
hasDocument Pre-Admission_checklist
postPolicy Admittance
policySatisfyQualityReq resource_utilization
policyBelongToCS Patient_Management_System
prePolicy Admission_Criteria
hasDocument Pre-Admission_Form
performedByRole admitting_medical_staff_members

Data property assertions
policyRevisionNo "4"
policyPurpose "1.1 Establish a standard procedure for the patients' admission to insure the appropriate use of center's resources 1.2 To define the responsibilities of those involved with patients' admission."

Figure 5-5: Snapshot of the Policy Ontology Model

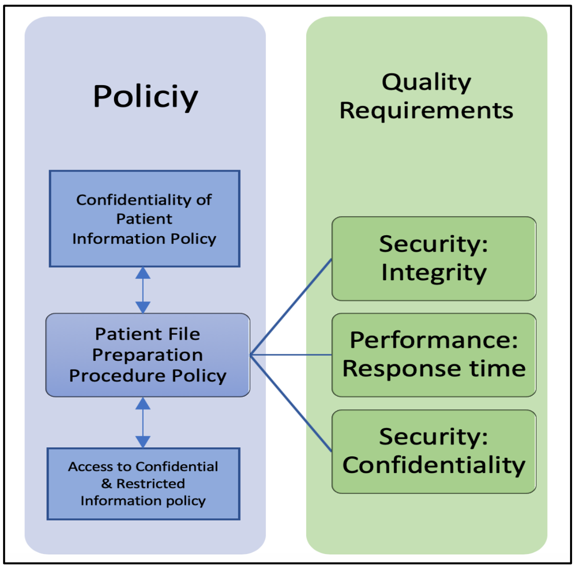
A snapshot of the policy ontology representation including some examples of instances from the case study is represented in Figure 5-5. The instances were created using the Protégé tool.

An example to demonstrate the interaction between CS's policies and QRs is depicted in Figure 5-6 which demonstrates the adherence to the quality requirement 'Patient's safety', following the process in Algorithm 5-1. Firstly, (step 1 in Algorithm 5-1) all related policies that have the purpose of satisfying 'Patient's safety' and satisfying the QRs that have relations with 'safety' QR, e.g. 'resource utilisation: efficient use of centre's resources'. The resulting policies (i.e. policy_set) will be used to perform the second step. Many policies resulting of this step, e.g. 'Flow of Patient and Waiting List Management policy', 'Discharge of Patients from the Centre policy', 'Discharge planning process', 'Critical care unit diversion policy', 'Processing Allogeneic Cellular Products policy', 'Admission of Patients policy', 'High Alert Medications Policy', 'Hand Hygiene Policy', 'Fall Prevention and Management policy', etc. All the aforementioned policies refer to different CSs, mainly related to the CSs that are related to KHCC Admission, Discharge and Transfer (ADT), and Cell Therapy and Applied Genomics (CTAG), e.g. surgical management systems, patient management system, financial system and pharmacy system. Secondly, (step 2 in Algorithm 5-1) several steps of comparisons need to be conducted in order to list any conflicts (if any) and to list the characteristics that need to be considered later by managers for any required quality-related decisions. However, for this example, the general aspects of the QRs specifications were satisfied by the policies and no major conflicts were identified and all policies conformed with the safety QR. However, some missing constraints were identified in some policies, e.g. cost-related aspects in the 'Discharge planning process'. This policy has a related policy that is called 'Discharge of Patients from the Centre policy' and covers time aspects. Another example is shown in Figure 5-7.



Quality Requirement
Policies
Resource utilisation
Patient's safety
Flow of Patient and Waiting List Management policy
Discharge of Patients from the Centre policy
Discharge planning process
Critical care unit diversion policy
Processing Allogeneic Cellular Products policy
Admission of Patients policy
High Alert Medications Policy
Fall Prevention and Management policy
Hand Hygiene Policy
etc.

Figure 5-6: Partial Example of Interactions in Increment 2 - Algorithm 5.1



Policiy
Confidentiality of Patient Information Policy
Patient File Preparation Procedure Policy
Access to Confidential & Restricted Information policy
Quality Requirements
Security: Integrity
Performance: Response time
Security: Confidentiality

Figure 5-7: Partial Example of Interactions in Increment 2- Algorithm 5.2

In order to have a sufficient representation of increment 2 evaluation, several test cases are required to cover all the concepts of both QR model and policy model. One test case may not be able to cover all main concepts and relations of the increment. For example, the policy 'Flow of patient and waiting list management policy' does not need any documentation to be completed. So, another test case was used, which is the policy 'Emergency admission of patients' policy'. Table 5-4 provides an overview of the coverage of test cases used. The common concepts between policy and QR models are written in green.

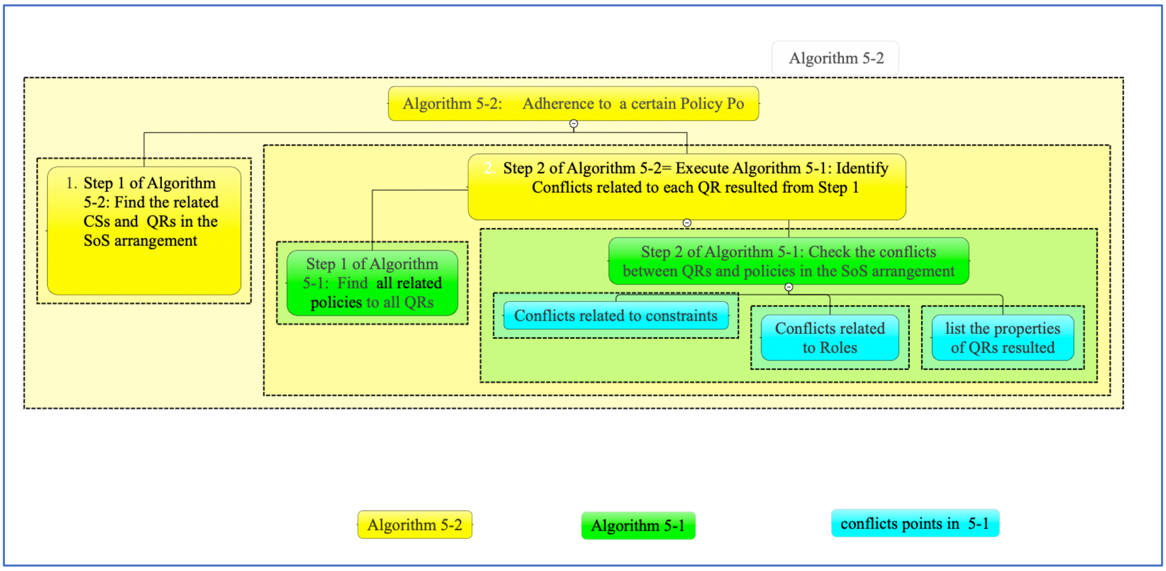

Algorithm 5-2
Algorithm 5-2: Adherence to a certain Policy Po
1. Step 1 of Algorithm 5-2: Find the related CSs and QRs in the SoS arrangement
2. Step 2 of Algorithm 5-2= Execute Algorithm 5-1: Identify Conflicts related to each QR resulted from Step 1
Step 1 of Algorithm 5-1: Find all related policies to all QRs
Step 2 of Algorithm 5-1: Check the conflicts between QRs and policies in the SoS arrangement
Conflicts related to constraints
Conflicts related to Roles
list the properties of QRs resulted
Algorithm 5-2
Algorithm 5-1
conflicts points in 5-1

Figure 5-8: Conceptualisation Process of Algorithm 5-2

To illustrate the demonstration of the interaction provided in Algorithm 5-2, the steps in Algorithm 5-2 are depicted in Figure 5-8 and can be illustrated as follows: checking the adherence of the policy 'Patient File Preparation Procedure policy IPPCTAG-BMT-27R5', which belongs to the Bone Marrow transplants CS, and its related QRs. The first step (Figure 8-5, Step 1 in green colour) is to find all CSs engaged and all policies in all CSs that satisfy the relation 'a Policy *hasRelation* with none or more policies' need to be listed together to form the Policy set. Here, the resulting policies are:

1) Patient File Preparation Procedure policy IPPCTAG-BMT-27R5 – belongs to BMT CS
2) Cord Blood Collection Procedure Policy IPPCTAG-BMT-15R4– belongs to BMT CS
3) Patient/Donor Preparation Procedure IPPCTAG-BMT-35R5– belongs to BMT CS
4) Confidentiality of Patient information Policy POLMRC -09R2– belongs to SoS/quality management.

5) Access to Confidential and Restricted Information Records POLMRC -12R4 – belongs to SoS/quality management.

6) Patient Identification Policy POLPIC-01R6– belongs to SoS/quality management.

7) Medication Administration Policy (POLNUR-11R7) – belongs to Nursing CS.

8) Blood Product Administration & Monitoring (POLNUR-10R7) – belongs to Nursing CS.

9) Specimen collection/handling instructions manual (IPPLAB-PHL01/03R5) – belongs to Pathology and Laboratory CS.

10) Patient Transportation Policy+ Ticket to ride (POLNUR-22R6) – belongs to Nursing CS.

11) Radiation Oncology - Patient Identification (POLRAD -06R5) – belongs to Radiology Oncology CS.

12) Patient movement and Care in Radiology (POLDGR -97R3) – belongs to Diagnostic Radiology CS.

13) Radiology request form Oncology (POLDGR-67R7) – belongs to Diagnostic Radiology CS.

14) DPLM Labeling Procedure IPPLAB-GEN.33R5– belongs to Pathology and Laboratory CS.

15) Handling Laboratory Verbal Requests and Results IPPLAB-GEN.24R4– belongs to Pathology and Laboratory CS.

16) Reporting of Lab Results IPPLAB-GEN.32R5– belongs to Pathology and Laboratory CS.

Also, the list of all related QRs to the polices in the policy set need to be listed in the QRs set as well. Here the QRs of ‘confidentiality’, ‘identify patients correctly’, ‘response time’ and ‘safety for patients and donors’ are the main QRs that the above policies should satisfy (Figure 8-5, Step 1 in orange colour).

The CSs’ set contains all the CSs that are involved with: SoS/quality management, Bone Marrow transplants CS (MBT), Pathology and Laboratory CS, Nursing CS, Diagnostic Radiology CS, Radiology Oncology CS.

The second step (Figure 8-5, Step 2 in green colour) is to check the conformance of policies in the policy set to each QR in the QRs set and to identify all conflicts between them. The results of conflicts are represented in the Conflicts_Poicy_QR_Set.

Few time-related constraints were identified:

-No match: Time 2 minutes sample collection time in the policy 15R4 and missing from the QR time constraints in the QR-related documents.

- No match: Timing for identification of patients should be before any diagnostic according to identify patients correctly QR and to 35R5 and 06R5 policies but it is missing from 15R4 and 01R5 policies.

A summary of the results is as follows:

- Policy set ={27R5, 15R4, 09R2, 12R4, 01R6, 11R7, 10R7, 03R5, 22R6, 06R5, 97R3, 67R3, 33R5, 24R4, 32R5, 35R5}
- CS set ={SoS, BMT, Diagnostic Radiology, Radiology Oncology, Nursing }
- QRs set ={ confidentiality, response time, integrity security, identify patients correctly, safety for patients and donors }
- Conflicts_Poicy_QR_Set ={ sample collection time, identification of patients' time, 35R5, 06R5, 15R4, 01R5, 15R4}
- QR_Properties= { Identify patients correctly-Strongly required mandatory, identify patients correctly- exact tendency, Safety for patients and donors- Strongly required mandatory, Safety for patients and donors- exact tendency}.

The next section details the evaluation process that followed.

5.4. The Evaluation of Increment 2

The first evaluation step was to conduct a comparative evaluation between the developed ontology policy model and some of the main models in literature. The result of the comparison is presented in Table 5-2. The comparison criteria are the elements of the key concepts that needed to be considered for developing the policy model, which were presented in section 5.2 and specified by (Garcia and Toledo, 2008; Lee et al., 2002).

Table 5-2: Comparison between some Policy Models in Literature

	Can be extensible	Include the concepts of Conditions and Actions	Support Constraints, e.g. Time Constraints	Support product aspects, e.g. Artefacts	Support process aspects, e.g. Task or Activity	Support Organizational aspects, e.g. Role	Linked to services	Linked to quality requirements	Support SoS-related concepts	General model or specific to a single domain	Ontology- based model
(Payne and Metzler, 2005)	Yes Specific to one domain	Yes Specific to one domain	No	Yes Specific to one domain	Yes Specific to one domain	Yes Specific to one domain	No	No	No	Specific domain	Yes
(Garcia and Toledo, 2008)	Yes	Yes Specific to one domain	Yes	Yes Specific to one domain	No	No	Yes	Partially Security aspects only	No	Specific domain	Yes
(Phan et al., 2008)	Yes	No	Yes	No	Yes	Yes	Yes	Partially	No	Services interaction domain	No

Ponder' policy model (Damianou et al., 2001)	Yes	Yes	Yes	No	Yes	Yes	Yes	Partially Security aspects	No	General	No
(Snir et al., 2003)	Yes Specific to one domain	Partially Actions only	Partially	No	Yes	Yes	Yes	Yes	No	Specific domain	No
OntoSoS. Policy	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	General	Yes

It is concluded from the comparison between OntoSoS.Policy and other policy models reviewed in Chapter 2, Section 2.9 and Table 5-2 that the OntoSoS.Policy model combines the advantages of the other models and follows the criteria defined by (Garcia and Toledo, 2008; Lee et al., 2002).

Secondly, Table Table *5-3* reflects the main interview questions and the checklists conducted with domain experts (Appendix III includes the list of documents that were used during the evaluation) to verify the structural correctness of OntoSoS.Policy model and consistency of the structure of the model. The conceptual view of the policy model (Figure 5-3) has been verified as well. Only suitable participants were selected who were able to provide the necessary knowledge and had experience in the area of policies, quality requirements and modelling.

Table 5-3: Checklist to to Validate the Policies Ontology Model Using the Walkthrough Approach

	Goal	**Concepts and properties**	**Result**		**Remarks from Domain Experts** (if not satisfied)
			Satisfied	unsatisfied	
1	Validate the concept "Policy" and its Data Properties (DP) and Object Properties (OP).	Concept: Policy	*		
		DP: policyName	*		
		DP: policyNumber	*		
		DP: originateDate	*		
		DP: revisionNumber	*		
		DP: revisionDate	*		
		DP: policyPurpose	*		
		DP: policyScope	*		
		DP: policyType	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
		DP: exception	*		Not applicable always
		DP: version	*		
		DP: description	*		
		OP: has_alternative	*		Not applicable always
		OP: constraintBy	*		
		OP: assert/interactWith	*		Not applicable always
		OP: hasDocument	*		Not applicable always
		OP: satisfyQR	*		
		OP: hasReference	*		Not applicable always
		OP: belongToSystem	*		
		OP: postPolicy	*		
		DP: approvedBy	*		
		OP: responsibilityOf	*		Can be related to one or more departments
2	Validate the concept “Policy-Reference” and its properties	Concept: PolicyReference		*	No Need
		DP: referencePointDescription		*	No Need
		OP: hasReference		*	No Need
3	Validate the concept “ApprovingCommittee” and its properties	Concept: ApprovingCommittee	*		Suggested by some Domain experts
		DP: committeeName	*		
		OP: approvedBy/approve	*		
4	Validate the concept “DocumentationRequirement” and its properties	Concept: DocumentationRequirement	*		Not always applicable
		DP: docName	*		
		OP: hasDocument	*		Not always applicable

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
5	Validate the concept "Service" and its properties	Concept: Service	*		Not always applicable
		DP: serviceName	*		
		OP: assert/assertedBy	*		
6	Validate the concept "Role" and its properties	Concept: Role	*		
		DP: RoleName	*		
		OP: perform/performedBy	*		
		OP: related To Department	*		Can be related to more than one department but to one system
		OP: preparedBy/prepare	*		
7	Validate the concept "Staff" and its Properties	Class: Staff	*		
		OP: is_A	*		
8	Validate the concept "Manager" and its Properties	Class: Manager	*		
		OP: is_A	*		
9	Validate the concept "Patient_Customer" and its Properties	Class: patient_customer	*		Carer can be included as well
		OP: is_A	*		
10	Validate the concept "Department" and its Properties	Concept: Department	*		
		DP: departmentName	*		
		OP: responsibilityOf	*		Suggested by some domain experts
		OP: role_relatedTo_department	*		
11	Validate the concept "Condition" and its Properties	Concept: Condition	*		
		DP: conditionName	*		
		OP: prePolicy	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
12	Validate the concept “ConstraintOnPolicy” and its Properties	Concept: ConstraintOnPolicy	*		
		DP: constraintName	*		
		OP: hasConstraint	*		
13	Validate the concept “TimeConstraint” and its Properties	Concept: TimeConstraint	*		
		DP: TimeDescription	*		
		OP: is_A	*		
14	Validate the concept “PrivilegesConstraint” and its Properties	Concept: PrivilegesConstraint	*		
		DP: PrivilegesDescription	*		
		OP: is_A	*		
15	Validate the concept “Triger” and its Properties	Concept: Triger	*		
		DP: name	*		
		OP: operate	*		
16	Validate the concept “System” and its Properties	Concept: System	*		
		DP: name	*		
17	Validate the concept “ConstituentSystem” and its Properties	Concept: ConstituentSystem	*		
		OP: is_A	*		
18	Validate the concept “NonConstituentSystem” and its Properties	Concept: NonConstituentSystem	*		
		OP: is_A	*		
19	Validate the concept “SoS_Level” and its Properties	Concept SoS_Level	*		
		OP: is_A	*		
20	Validate the concept “Action” and its Properties	Concept Action	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
21	Validate the relation ‘satisfies’ between the two models		*		

In order to have a sufficient representation of the policy model, several test cases are required to cover all the main concepts of the model. Table 5-4 provides the details of the test cases coverage. Then, the experts have been interviewed (see Table 5-5) to validate the completeness of policy model representation of the case study. Some examples of the validated representation are depicted in Figures Figure 5-5, Figure *5-6* and Figure *5-7*. The evaluation continued to validate the relation and the interaction between policies and QRs. The feedback resulted from this evaluation has been used to revisit increment 2. Thus, at the end of this phase, increment 2 was validated and the output was the second version of increment 2. This version consists of the second version of the policy model combined with any suggested modification to the QR model. However, no further modifications were suggested to be applied to the OntoSoS.QR model.

Table 5-4: Testing Cases Coverage Matrix

Concepts	Policy	System-Related	Service Applied	Related Department	Trigger	Action	Condition	Document-related	Role	Approving committee	Constraint	Exist of other related QR	Exist of other related Policy
Test 1 (More details in Appendix II)	Flow of patient and waiting list management policy	√	√	√	√	√	√	-	√	√	√	√ Safety, Care delay Avoidance, Recourse Utilisation (Appropriate use of centre’ s resources)	√
Test 2	Patient File Preparation Procedure policy 27R5	√	√	√	√	√	√	√	√	√	√	√	√

Table 5-5: The Outcomes of the Main Questions of the Interviews Conducted with Domain Experts to Validate *SoS Policies Model and the Relations with Other Models in Increment* 2

	Questions	Notes
1	Do you agree that the above concepts and associated properties **completely** represent the KHCC's policy related concepts?	✓
2	Are there any missing concepts or properties?	The concept Approving_Committee was suggested
3	Are there any other properties that could enhance the model?	A relation to connect between the policy and the new suggested concept Approving_Committee. Also, a connection between the concepts Role and Action need to be available, e.g. 'Action is performed By a Role
4	Are there any extra concepts or properties that would be better removed from the model?	Domain experts suggested that no need to include the concept 'Policy Reference' as the recursive relation from one policy to its related policies is more general.
5	Do you agree that the above concepts and associated properties **correctly** represent the KHCC's policy related concepts?	✓
6	Are the data types that are used to describe the Data Properties correct?	✓
7	Are the relations (Object Properties) that are used to connect the concepts of the policy model, correct?	✓
8	Are the domain and the range for the relations between the concepts correct?	✓
9	Are the cardinality constraints of the relations correct?	A Role can be related to more than one department (Suggested)
10	Do you agree that the above concepts and associated properties are **consistent** with each other and do not lead to conflict (free of contradiction with other components in the model)?	✓

	Questions	Notes
11	Do you agree with the relation between the policy model and QR model?	✓
12	Do you agree with the process (Algorithms) of interaction provided?	✓ The interaction provided needs to provide more information regarding Roles as the roles identified in the policy model refers to a role conducted an action. So, extra information needed to identify who involves in the process of adherence to a particular policy.
13	Do you suggest any further concepts/ changes to be applied on increment 2?	N/A

As reflected from both Tables Table *5-3* and Table *5-5*, the main concepts of the OntoSoS.Policy model were validated by experts. Only few amendments were suggested that were implemented. For instance, the concept ‘Approving Committee’ was suggested to indicate the title of the committee that wrote or approved the policy. This could help to identify who needs to revisit a policy in the case of a conflict occuring. Another example was the cardinality between the concept ‘Role’ and ‘Department’ to be ‘ A Role belongs to at least one or more Departments’.

Based on the evaluation conducted in the second DSRM process increment, the following has been concluded:

i) The evaluation criteria (i.e. correctness, consistency and completeness) were satisfied. Using semi-structured interviews, walkthrough and checklists of both the OntoSoS.Policy ontology model and the linking of the policy model to the OntoSoS.QR model resulted in domain experts’ satisfaction. Both models were completely and correctly captured by the ontology as revealed by domain experts. Also, all concepts are structurally and logically consistent (i.e. free of contradiction with other components in the ontology), which was also confirmed by applying consistency checks using a reasoner tool installed with the Protégé ontology editor (the results of the evaluation support answering RQ4).

ii) Reusing and extending some existing ontologies in the literature helped building the OntoSoS.Policy ontology model. A comparison was conducted between the OntoSoS.Policy and other policy models from literature. The results showed that the OntoSoS.Policy model is mainly comprehensive and combines the strengths of some existing ontologies (e.g. examples of ontologies adapted from Table 5-2). This demonstrated that using ontology was suitable to define all related aspects of the policy model (this supports answering RQ2).

iii) It was shown in Table 5-5, point 12, that the process provided in Algorithms 5-1 and 5-2 lacked capturing some concepts that are important for checking the adherence to a policy. For example, the interaction provided needs to provide more information regarding 'Roles', as the roles identified in the policy model refer to who conducts an action. So, extra information is needed to identify who is involved in performing these actions and procedures identified in the policy document to check the full adherence to a policy. So, this initiates the need to go through the third DSRM increment that includes developing the OntoSoS.Process model, which contains information regarding actions and their related roles. Chapter 6 details the development of a process model that interacts with policies (this supports answering RQ3).

iv) Some minor amendments were suggested by the domain experts when applying the research cancer care case study which led to enhancements as reflected in the next section.

5.5. Revisiting Increment 2

After the validation of the OntoSoS.Policy model and the interaction between both the policy and QR models, some minor amendments were attempted to produce the second version of these two models, which are depicted in Figure 5-9 (amendments highlighted with blue colour). Some object properties (i.e. relations) were slightly changed. For instance, additions were suggested, e.g. 'constraintBy/interactWith', as it is a recursive relation to the 'policy' concept, e.g. the admission- related policies constrained by the registration policy. Also, adding the policy data property 'approvedBy' between a' Approving_Committee' and 'Policy' concepts.

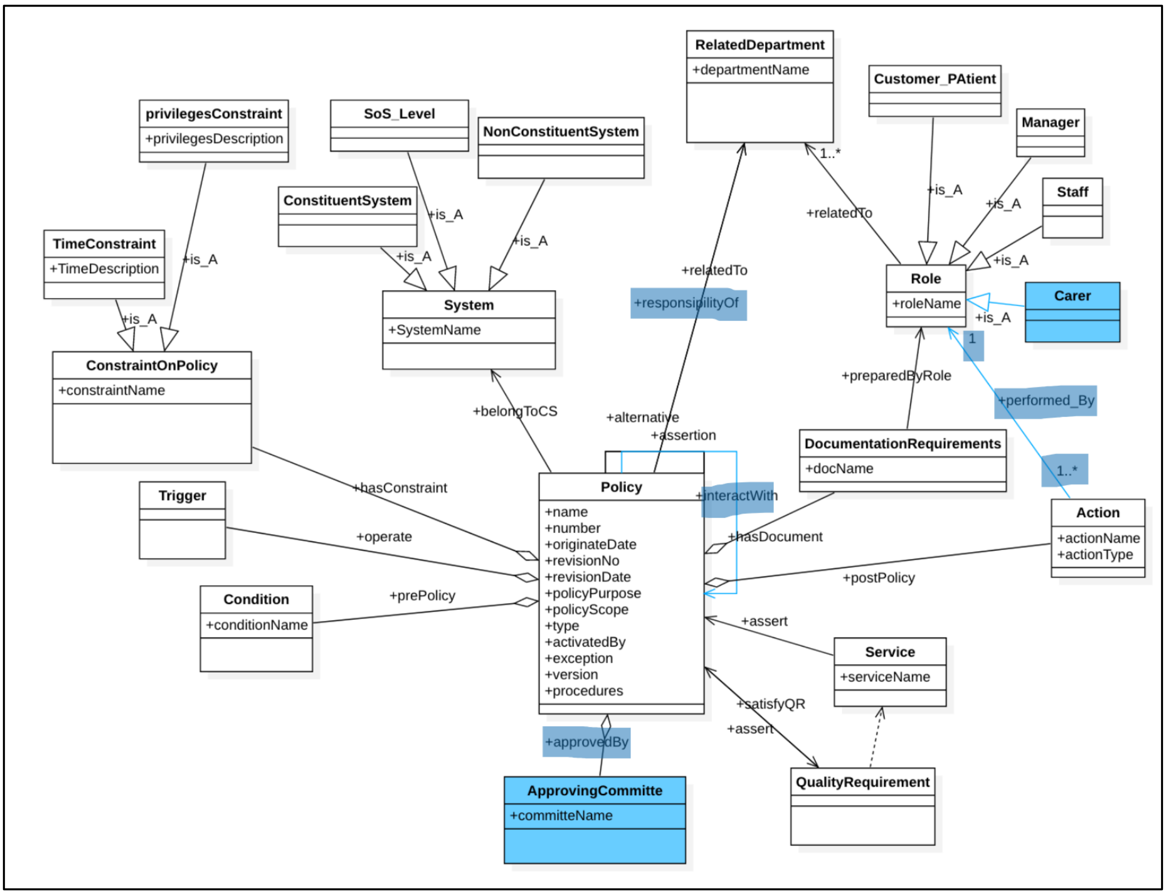


Figure 5-9: The Conceptual Representation of Version 2 of the OntoSoS.Policy

By the end of DSRM iteration 2, Table 5-6 provides an overview of the status towards answering the research questions. The tick symbol (✓) shows that the research question/sub-question has been answered, and the hourglass with flowing sand symbol (⏳) shows that the question is partially answered.

Table 5-6:Status towards Answering the Research Questions

RQ	Main RQ and RQs-Concerns (sub-questions)	Status	Notes	Chapter
RQ1	*What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*	✓		2
1.1	Can research gap analysis, by surveying the literature, identify these challenges ?	✓		2
RQ2	*How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*	⌛		2, 4,5,6 & 7
2.1	What are the components of the quality governance framework, and what are the initial specifications of each component ?	✓		2
2.2	Is ontology suitable to define all related aspects to quality governance ?			7
2.3	What are the elements/concepts of the ontology? Are there any ontologies that we can reuse ?	⌛	See Tables Table 4-1, Table 4-2, and Table 5-1	4,5,6 & 7
RQ3	*How will the semantically-enriched models of policies, processes, standards and quality requirements interact in the systems of systems context to identify and resolve semantic heterogeneities?*	⌛		5,6 & 7
3.1	Can we develop a process to detail the interaction between *policies, processes, standards and quality requirements to identifying and resolve semantic heterogeneities* ?	⌛	See Algorithms Algorithm 5-1 and *Algorithm 5-2*	5,6 & 7
3.2	Are there any limitations of using an ontology-based approach that restrict the interactions between *policies, processes, standards and quality requirements* ?	⌛		5,6 & 7
RQ4	*How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*	⌛		4,5,6 & 7
4.1	How the OntoSoS.QM.Gov ontology will be assessed ?			7
4.2	Can we validate each ontology model (i.e. each component of the quality governance framework) and then to validate the whole quality governance ontology framework ?	⌛	See Sections 4.4 and 5.4	4,5,6 & 7

5.6. Summary and Conclusion

After reviewing the literature in relation to policy models, they have been utilised to support the design of the OntoSoS.Policy model in the SoS context. After developing the first version of the OntoSoS.Policy, it was linked with the previously developed increment (i.e. QR model) to formulate the second increment of the quality governance framework. This increment was validated with domain experts from KHCC case study. The feedback from domain experts was applied to the first version of OntosoS.Policy leads into a second version of the OntoSoS.QR and OntoSoS.Policy models. However, only minor amendments were required to be reflected in the design.

After examining the interaction between policies and QRs, it was inferred that such interaction is not sufficient for checking the adherence to quality governance. This was attributed to the nature of concepts dealt with both models. For instance, the QRs model did not deal with actions and their related roles, while polices included actions and roles. However, the actions and roles- related concepts were part of the process model, which was then decided to be added in the next DSRM iteration.

After revisiting the model, the second DSRM increment was completed and the updated version of the OntoSoS.QR and OntoSoS.Policy models paved the ground to link them with the OntoSoS.Process in the third DSRM process increment which will be discussed in Chapter 6.

6. Chapter Six: The Processes Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment III)

6.1. Introduction

This chapter details the third increment of the DSRM process that has been followed in this research. Firstly, the design of the processes model (i.e. the OntoSoS.Process model) is described in detail. Then, the interaction between the previously developed models (i.e. the OntoSoS.QR and the OntoSoS.Policy models) with the new OntoSoS.Process model is described; followed by the demonstration phase of Increment 3 with the cancer care research case study. After this, the evaluation phase of Increment 3 is attained with feedback used to revisit the design of Increment 3.

The next section details the construction process of the process model (i.e. OntoSoS.Process) and the linkage between it and both the OntoSoS.QR and OntoSoS.Policy models. Then, the demonstration of Increment 3 by using examples from the cancer care case study is described in Section 6.3. Increment 3 evaluation phase is described in Section 6.4. Finally, Section 6.5 details the feedback reported by domain experts, which has been used to revisit the design of the OntoSoS.Process model ontology.

6.2. The Design of the OntoSoS.Processes Ontology Model Linked to Increment 2: Increment 3

Firstly, the domain and the scope of the OntoSoS.Process is articulated. Process components (i.e. activities, products, agents and tools) and their interactions (information flow, artefacts flow, etc.) can vary. Process implementation can differ depending on the organisations' level, scope, and goals (INCOSE, 2015, p38). In this research, and in particular, in this DSRM increment, the focus is on building business processes-related concepts that are consistent with BPMN concepts. Thus, the main concepts used in this research to model business processes were adopted from the BPMN notations. Also, the ontological models of the processes of the research case study had already been developed using BPMN and evaluated in the work of Odeh et al. (Odeh et al., 2018). Also, the process of transforming BPMN concepts to ontological concepts was identified and illustrated by Fan et al. (Fan et al., 2016), and the use of BPMN has been justified in Chapter 2 (Section 2.11).

Secondly, in this research, the main ontology process-related concepts depending on the definition made in the works of (Fan et al., 2016; Nicola et al., 2007; Rospocher et al., 2014) were investigated. The authors used the concepts of the BPMN to build their ontology process models. Also, BPMN concepts that were used in the work of (Fan et al., 2016; Natschlager, 2011; Nicola et al., 2007; Nicolae et al., 2009; Rospocher et al., 2014; Yousef et al., 2009b) were studied and used in the current work. This has resulted in the process concepts utilised in this research to build the SoS processes ontology.

Thirdly, after determining the main terms, a number of related process modelling concepts need to be introduced with other related concepts, for instance, the main concept ‘Role’ has three sub-classes: ‘Staff’, ‘Tool’, and ‘Patient’. Some of them needed to be organised in a class hierarchy. Figure 6-1 shows the conceptual representation of the OntoSoS.Process ontology model.

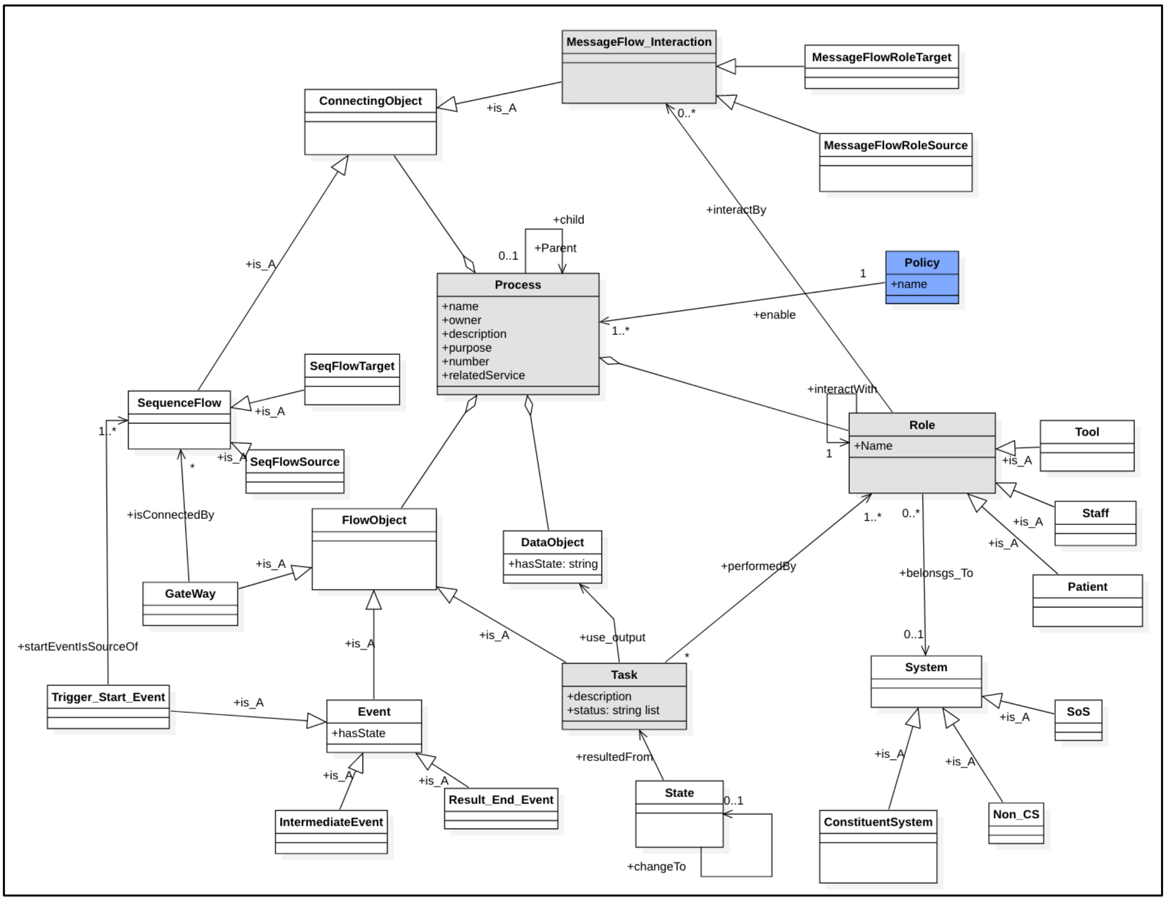


Figure 6-1: The Conceptual Representation of the of OntoSoS.Process Model

The next two steps related to identify data and object properties of the classes (slots) as well as to identify any restrictions on these slots. For instance, the data property ‘role name’ for the concept ‘role’ and the data property ‘process status’ for the concept ‘process’, etc. (Rospocher et al., 2014). Also, as illustrated by Noy and McGuinness (2011), restrictions in relation to the types of the slots values (e.g. string, number, Boolean, etc.)

had to be defined, as well as the slot cardinality to show how many values a slot and a domain can have, e.g. ‘role has *some* tasks’. Some objects properties (i.e. relations) were suggested, e.g. ‘task *reference to* task’ as it is a recursive relation to the ‘task concept’ (Rospocher et al., 2014).

Table 6-1: OntoSoS.Process Main Concepts and Traceability to Literature Sources

	Concept	Main Sources and Description	Properties and their types
1	Process	The root element that indicates a process (Fan et al., 2016; Natschlager, 2011; Nicola et al., 2007; Rospocher et al., 2014)	-**name**: string. The name of the process -**number**: string. The unique number of the process -**processOwner**: string. Role owner of a process -**processPurpose**: string. The main aim of the process - **description**: string. The description of a process -**satisfyService**: string. The related service
2	Role	Equivalents to pool and lane in BPMN. Represents a participant in a process (Fan et al., 2016; Natschlager, 2011; Nicola et al., 2007; Rospocher et al., 2014)	-**name**: string. The name of the actor -**roleSystem**: string. The related CS.
3	DataObject	A *data object* is used to represent how data and documents are used within a process (Nicola et al., 2007)	-**name**: string. The title of the artefact -**description**: string. The description of the artefact
4	Task	Represents an activity (Nicolae et al., 2009; Rospocher et al., 2014).	-**name**: string. The title of the task -**taskType**: string list (send, receive, manual, etc.) -**hasStatus**: string list {None, Ready, Active, Cancelled, Aborted, Completed}. The status of the task.
5	Gateway	Decisions and branching are represented by the following *gateways* (Nicola et al., 2007; Nicolae et al., 2009)	-**name**: string. The name of the Gateway -is connected by a gateway: Relation
6	Event	An *event* is something that “happens”, like a trigger or a result, during the execution of a business process affecting the flow of the *process* (Nicola et al., 2007) Has 3 types: Trigger, intermediate and result.	-**description**: string. The description of the action
7	Sequence Flow	used to represent the ordering of activities within a *process*. Their source and target must be *events*, *tasks*, and *gateway* (Fan et al., 2016; Natschlager, 2011; Nicola et al., 2007; Rospocher et al., 2014)	- **description**: string. The description of the sequence flow - **sourceFlow**: Relation -**distinationFlow**: Relation
8	Interaction Message	Message flow between roles. The *message flow* is used to show the flow of messages between two participants of a process (Fan et al., 2016; Natschlager,	- **description**: string. The description of the interaction between roles. -**sourceInteract**: Relation

	Concept	Main Sources and Description	Properties and their types
		2011; Nicola et al., 2007; Rospocher et al., 2014)	
9	System	A role can belong to the SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement. Domain experts and brainstorming to elicit SoS and CS concepts.	-**name**: string. The name of the constituent system
10	SoS	Brainstorming of concepts: A role can belong to a higher level i.e. SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement.	-**name**: string.
11	CS	Brainstorming of concepts: A role can belong to a higher level i.e. SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement.	-**name**: string.

As seen from Figures Figure 5-9 and Figure 6-1, processes and policies models have been connected together. A process needs to conform to one or more policies as shown in Figures Figure 6-2 and Figure 6-3. The main concepts of the processes need to be aligned with the related concepts of the policies. These concepts are the roles and their interactions, the performed tasks and the associated artefacts/ data objects (in Figure 6-3, the common concepts between process and policy models are highlighted in orange, and the common concepts between QRs and policy models are highlighted in blue).

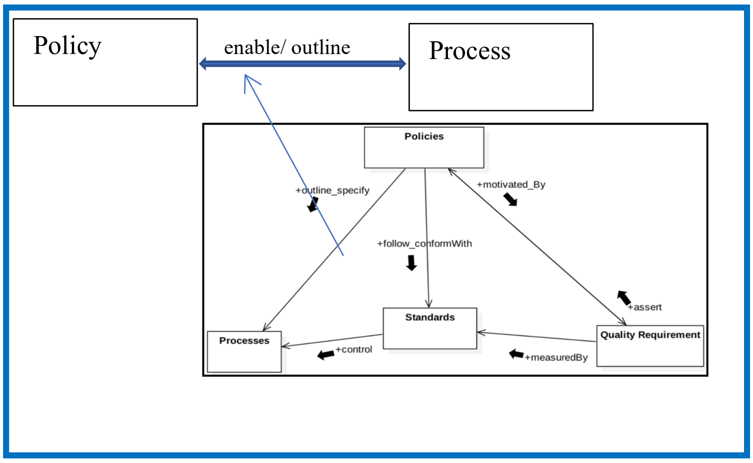


Figure 6-2: Abstract Level of the Proposed Quality Governance Model with Focus on the Link between Process and Policy Models

As depicted in Figures Figure 6-2 and Figure 6-3, both models (process, and policy) are connected via the relation ‘specify/outline’. Both models need to be aligned with each other. For instance, the conformance between roles, data objects/artefacts and tasks. One connection is to link a policy with a process via the ‘actions of the policy’ and the ‘tasks of the process’(Lee et al., 2002). This relation is depicted in Figure 6-3 as the relation ‘Policy *has_Post* Action' and that 'Action/Task *belongsTo* a Process'. (A larger view of Figure 6-3 in Appendix IV)

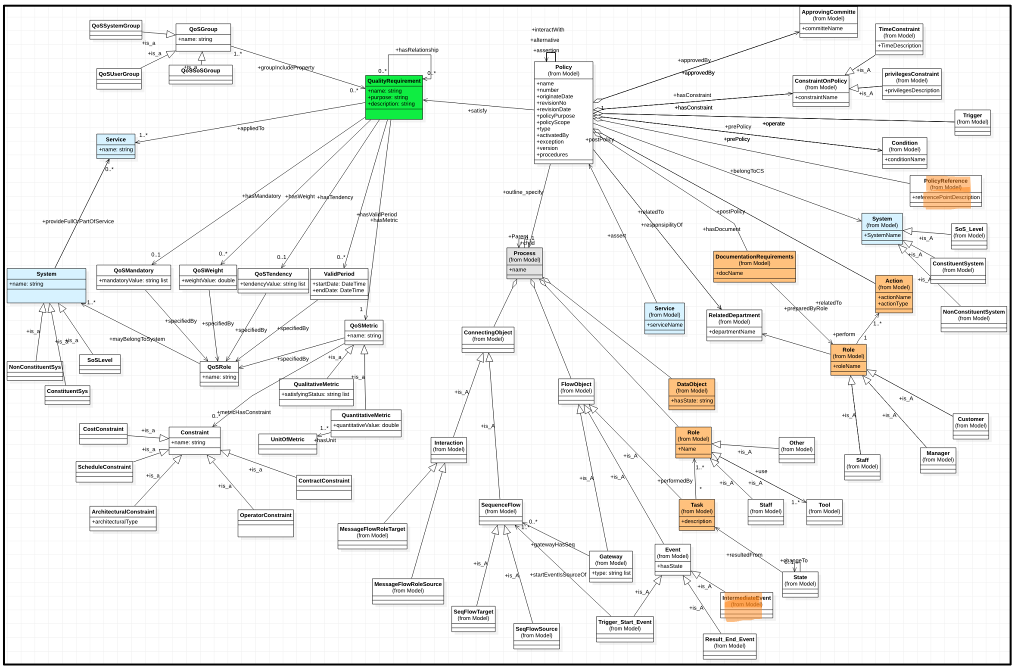


Figure 6-3: Conceptual View of Linking OntoSoS.QR, OntoSoS.Policy and OntoSoS.Process models in SoS context-Increment 3

In order to check the adherence for processes and policies, a process to indicate the interaction between both models is described in both Algorithms Algorithm 6-1 and Algorithm *6-2*.

Algorithm 6-1: Interaction between QRs, Policies and Processes (Alternative 1: QR-Policies-Processes)

Input:

QR$_i$ = The QR that we need to check governance for.

Output:

/*See Output of Step 1 (Algorithm 5-1: Interaction between QR and policies) */

CS.Policy_Set= { CS$_1$.Po$_1$, CS$_1$.Po$_2$, …, CS$_2$.Po$_1$, …, CS$_i$.Po$_j$, …, CS$_n$.Po$_p$}, the set of all policies Po$_j$ that are related to QR$_i$ in all constituent systems CS , where $0 < i <= n$, n=number of CSs; 0<j<=p, p=number of all policies in each CS.

Conflicts_Poicy_QR_Set= {CS$_1$.Role$_1$, CS$_2$.Role$_2$, CS$_1$.Constraint$_1$, CS$_2$.Constraint$_2$, CS$_1$.Service$_1$, … , CS$_i$.Role$_j$, CS$_i$.Constraint$_k$, CS$_i$.Service$_x$, …, CS$_n$.Role$_r$, CS$_i$.Constraint$_c$, CS$_i$.Service$_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=c, c=number of all constraints in each CS; and 0<x<=s, c=number of all services in each CS.

CS.QR.Properties_Set ={ CS$_1$.QR$_1$.Properties$_1$, CS$_1$.QR$_2$.Properties$_2$, …, CS$_i$.QR$_j$.Properties$_k$, …, CS$_n$. QR$_q$.Properties$_r$}, where $0 < i <=n$, n= number of CSs; 0<j<=q, q=number of all QRs in each CS; 0<k<=r, r=number of all Properties in each QR.

/* Other outputs*/

CS.Process_Set= { CS$_1$.Pr$_1$, CS$_1$.Pr$_2$, …, CS$_2$.Pr$_1$, …, CS$_i$.Pr$_j$, …, CS$_n$.Pr$_p$}, the set of all processes Pr$_j$ that are related to each policy in CS.Policy_Set in all constituent systems CS , where $0 < i <= n$, n=number of CSs, 0<j<=p, p=number of all processes in each CS.

Conflicts_Policy_Process_Set={CS$_1$.Role$_1$, CS$_1$.Task$_1$, CS$_2$.Documentation$_1$, CS$_1$.Action$_1$, CS$_1$.DataObject$_1$, …, CS$_i$. Role$_j$, CS$_i$.Task$_k$, CS$_i$.Action$_l$, CS$_i$.DataObject$_m$, CS$_i$. Documentation$_n$, …,CS$_n$. Role$_r$, CS$_i$.Task$_t$, CS$_i$.Action$_a$, CS$_i$.DataObject$_o$, CS$_i$. Documentation$_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where $0 < i <= n$, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=t, t=number of all tasks in each CS; 0<l<=a, a=number of all actions in each CS; 0<m<=o, o=number of all data objects in each CS; 0<n<=d, d=number of all documents in each CS;

```
Begin
/*Step 1: execute Algorithm 5-1 to elicit all related policies to the related QR and check the adherence of each policy to the QR*/
Execute Algorithm 5-1.
/*Step 2: Find all related processes to each policy resulted from Step1 */
For each policy Po$_k$ in Po_Set that resulted from step 1 do
    For each constituent system CS$_i$ do
        For each Process Pr$_j$ do
            Find all processes CS$_i$.Pr$_j$, where CS$_i$.Po$_k$ <<specify/outline>> CS$_i$.Pr$_j$
            add CS$_i$.Pr$_j$ from previous point to the processes set Pr_Set
        End for
    End for
End for

/*Step 3: Identify conflicts*/
For each policy Po$_j$ in Po_Set resulted from step 1 do
    For each process Pr$_j$ in Pr_Set resulted from step 2 do
    {
        For each Role in each policy
            find and compare between Po$_j$. Role. 'description' and Both (Pr. 'description' and Pr:'process' Owner').
            if doesnot match add results to Conflicts_Policy_Process_Set.
        End for
        For each document in each policy
            find and compare between Po$_j$. Documentation_ Required. 'name' and Pr. DataObject. 'name'.
            if doesnot match add results to Conflicts_Policy_Process_Set.
        End for
        For each Task in each policy
            find and compare between Po$_j$. Task. 'description' and Pr. Task. 'description' .
            if doesnot match add results to Conflicts_Policy_Process_Set
        End for
        For each Condition in each policy
            find and compare between Po$_j$. Condition. 'description' and Pr. Event. 'description' .
            if doesnot match add results to Conflicts_Policy_Process_Set
        End for
    }
    End for
End for
End
```

Algorithm 6-2: Interaction between QRs, Policies and Processes (Alternative 2: Process-Policies-QRs)

Input:

Pr_n = The Process that we need to check governance for.

Output:

/*See Output of Algorithm 5-2*/

CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies Po_j that are related to QR_i in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number of all policies in each CS.

Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each CS; and $0<x<=s$, c=number of all services in each CS.

CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, CS_n. $QR_q.Properties_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$, r=number of all Properties in each QR.

/* Other outputs*/

CS.Process_Set= { $CS_1.Pr_1$, $CS_1.Pr_2$, …, $CS_2.Pr_1$, …, $CS_i.Pr_j$, …, $CS_n.Pr_p$}, the set of all processes Pr_j that are related to each policy in CS.Policy_Set in all constituent systems CS , where $0 < i <= n$, n=number of CSs, $0<j<=p$, p=number of all processes in each CS.

Conflicts_Policy_Process_Set={$CS_1.Role_1$, $CS_1.Task_1$, $CS_2.Documentation_1$, $CS_1.Action_1$, $CS_1.DataObject_1$, …, CS_i. $Role_j$, $CS_i.Task_k$, $CS_i.Action_l$, $CS_i.DataObject_m$, CS_i. $Documentation_n$, …,CS_n. $Role_r$, $CS_i.Task_t$, $CS_i.Action_a$, $CS_i.DataObject_o$, CS_i. $Documentation_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=t$, t=number of all tasks in each CS; $0<l<=a$, a=number of all actions in each CS; $0<m<=o$, o=number of all data objects in each CS; $0<n<=d$, d=number of all documents in each CS;

```
Begin
/*Step 1: Find all related policies $Po_j$ to the entered process*/
 For each constituent system $CS_i$ in CSs do
     For each policy $Po_j$ do
         Find all policies $CS_i.Po_j$, where $CS_i.Po_j$ <<specify/outline>> $CS_i.Pr_n$.
         add $CS_i.Po_j$ from previous point to the process set Pr_Set
     End for
 End for
/*Step 2: Check the conformance of policies to the entered process*/
 For each policy $Po_j$ in the policy set Po_Set that resulted from step 1 do
    {
       For each Role in each policy
           find and compare between $Po_j$. Role. 'description' and Both (Pr. 'description' and Pr:'process' Owner').
            if doesnot match add results to Conflicts_Policy_Process_Set.
       End for
       For each document in each policy
              find and compare between $Po_j$. Documentation_ Required. 'name' and Pr. DataObject. 'name'.
             if doesnot match add results to Conflicts_Policy_Process_Set.
        End for
        For each Task in each policy
              find and compare between $Po_j$. Task. 'description' and Pr. Task. 'description' .
             if doesnot match add results to Conflicts_Policy_Process_Set
        End for
        For each Condition in each policy
              find and compare between $Po_j$. Condition. 'description' and Pr. Event. 'description' .
             if doesnot match add results to Conflicts_Policy_Process_Set
        End for

 End for
 /*Step 3: execute Algorithm 5-2 to check the conformance of policies to the related quality requirements*/
Execute Algorithm 5-2.
End
```

Algorithm 6-1 represents the process that checks the adherence to the entered quality requirement by checking if all its related policies in the SoS's constituent systems conform to their related processes. The related policies can be identified by executing Algorithm 5-1, which was discussed in Chapter 5 (Section 5.2). While Algorithm 6-2 illustrates the process to check if the entered process conforms to its related policies (Algorithm 6-2 is illustrated in Figure 6-8). The first step of Algorithm 6-2 is to find all related policies to the entered process. Then, for each resulting policy from the first step, a process will be conducted to check the conformance of that policy to the entered process. After this, Algorithm 5-2 will be executed in order to check the adherence of each policy to its related quality requirements.

These two algorithms represent the two possible alternatives as the relation between both models is bidirectional. So, the process is '*specified by one or more'* policies, and the policy can specify a process. If any policy or process satisfies this relation, they need to be selected to be in the policy set (Po_Set) or the process set (Pr_Set). Then, a comparison process is conducted through comparing the values of the main similar aspects, e.g. the values of the roles' titles involved, the performed tasks and other related concepts such as triggers and conditions of both models will be compared. Also, because there are recursive relations, e.g. 'Policy has relation with another Policy', more policies will be included in the process. An example will be shown in Section 6.3.

The output of executing the algorithms will be a list of conflicts (if any) between QRs, policies and processes along with the related CSs and related aspects, e.g. roles engaged, documents, tasks and all other results from executing the algorithms 5-1 and 5-2.

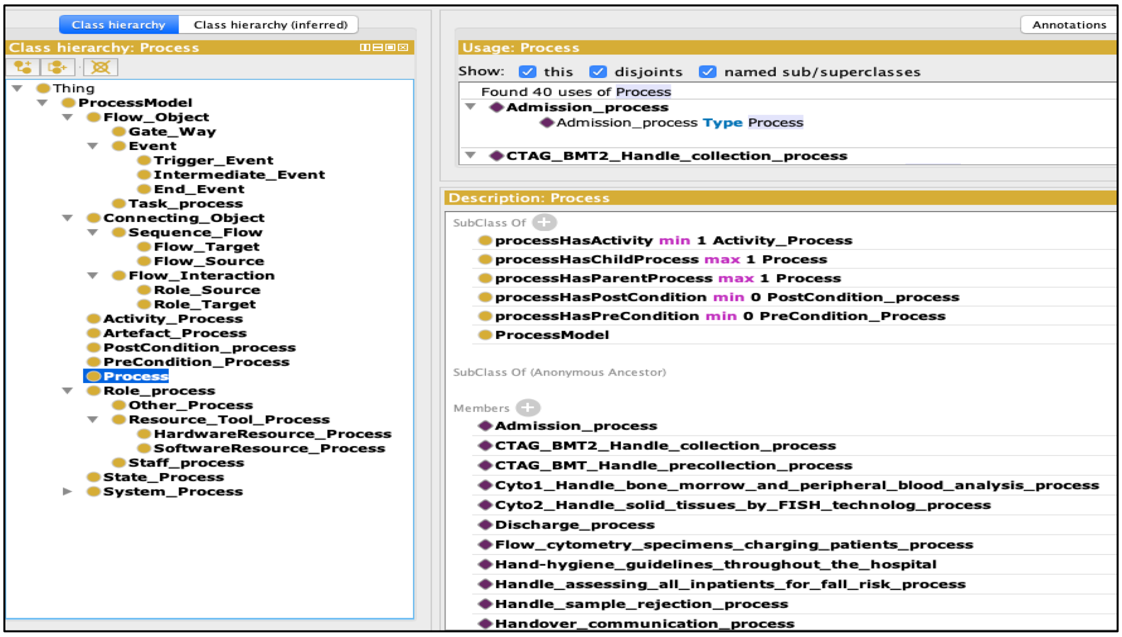
6.3. The Demonstration of Increment 3

In order to demonstrate Increment 3, some examples from the research case study were selected. A snapshot of the process ontology representation including some examples of instances from the case study is represented in Figure 6-4.

Figure 6-5 shows the BPMN of the process of 'Handling outpatients sample reception'. This process was chosen as a test case in this DSRM increment because it was described and validated in the literature (Odeh et al., 2018) and covers all concepts of the OntoSoS.Process. Also, it is connected to several policies and QRs, and it is a process that is related to the CTAG case study, which is part of the research case study described in Chapter 3 (Section 3.4). This process and its related policies and QRs are considered sufficiently representative test case for Increment 3 given that this selected process provides sufficient coverage for all concepts addressed in Increment 3 as shown in Table 6-2. The constituent systems attached to the process ‘Handling outpatients sample reception’ have been validated by the case study domain experts. In this process, seven roles are involved and represented using horizontal rectangles. The roles are CTAG main Receptionist, Patient, Flow Cytometry Receptionist, Pathologist, Flow Cytometry Technologist, Supervisor Flow Cytometry Technologist and Medical Director. This process is related to two main policies in the CTAG case study, the ‘Flow Cytometry test samples management policy’ and ‘Flow Cytometry charging and LIS policy’. Both of the former policies are related to some QRs, e.g. safety, usability and accuracy (see Figure 6-7). Only few conflicts identified that are related to the few ‘Roles’. For examples the roles ‘Pathologist’, ‘Main receptionist’ and ‘Patient’ are existing in the process but not in the policy ‘Flow Cytometry test samples management policy’ and the role ‘Main receptionist’ is existing in the policy ‘Flow Cytometry charging and LIS policy’ but not in the process.

Table 6-2: Testing Case Coverage Matrix

Concepts	Process	Start Event	End Event	Trigger	Message Flow	Trigger	Task	State	Data Object	Role	Role Target	Role Source	Exist of related QR	Exist of other related Policy	CS and Non-CS
Test 1	‘Handling outpatients sample reception’ process	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓ Safety, Accuracy, Usability	✓	✓



```
Thing
  ProcessModel
    Flow_Object
      Gate_Way
      Event
        Trigger_Event
        Intermediate_Event
        End_Event
      Task_process
    Connecting_Object
      Sequence_Flow
        Flow_Target
        Flow_Source
      Flow_Interaction
        Role_Source
        Role_Target
    Activity_Process
    Artefact_Process
    PostCondition_process
    PreCondition_Process
    Process
    Role_process
      Other_Process
      Resource_Tool_Process
        HardwareResource_Process
        SoftwareResource_Process
      Staff_process
    State_Process
    System_Process
```

Show: ☑ this ☑ disjoints ☑ named sub/superclasses

Found 40 uses of Process

- Admission_process
  - Admission_process Type Process
- CTAG_BMT2_Handle_collection_process

SubClass Of

- processHasActivity min 1 Activity_Process
- processHasChildProcess max 1 Process
- processHasParentProcess max 1 Process
- processHasPostCondition min 0 PostCondition_process
- processHasPreCondition min 0 PreCondition_Process
- ProcessModel

SubClass Of (Anonymous Ancestor)

Members

- Admission_process
- CTAG_BMT2_Handle_collection_process
- CTAG_BMT_Handle_precollection_process
- Cyto1_Handle_bone_morrow_and_peripheral_blood_analysis_process
- Cyto2_Handle_solid_tissues_by_FISH_technolog_process
- Discharge_process
- Flow_cytometry_specimens_charging_patients_process
- Hand-hygiene_guidelines_throughout_the_hospital
- Handle_assessing_all_inpatients_for_fall_risk_process
- Handle_sample_rejection_process
- Handover_communication_process

Figure 6-4: Snapshot to show a Part of the Process Ontology Model

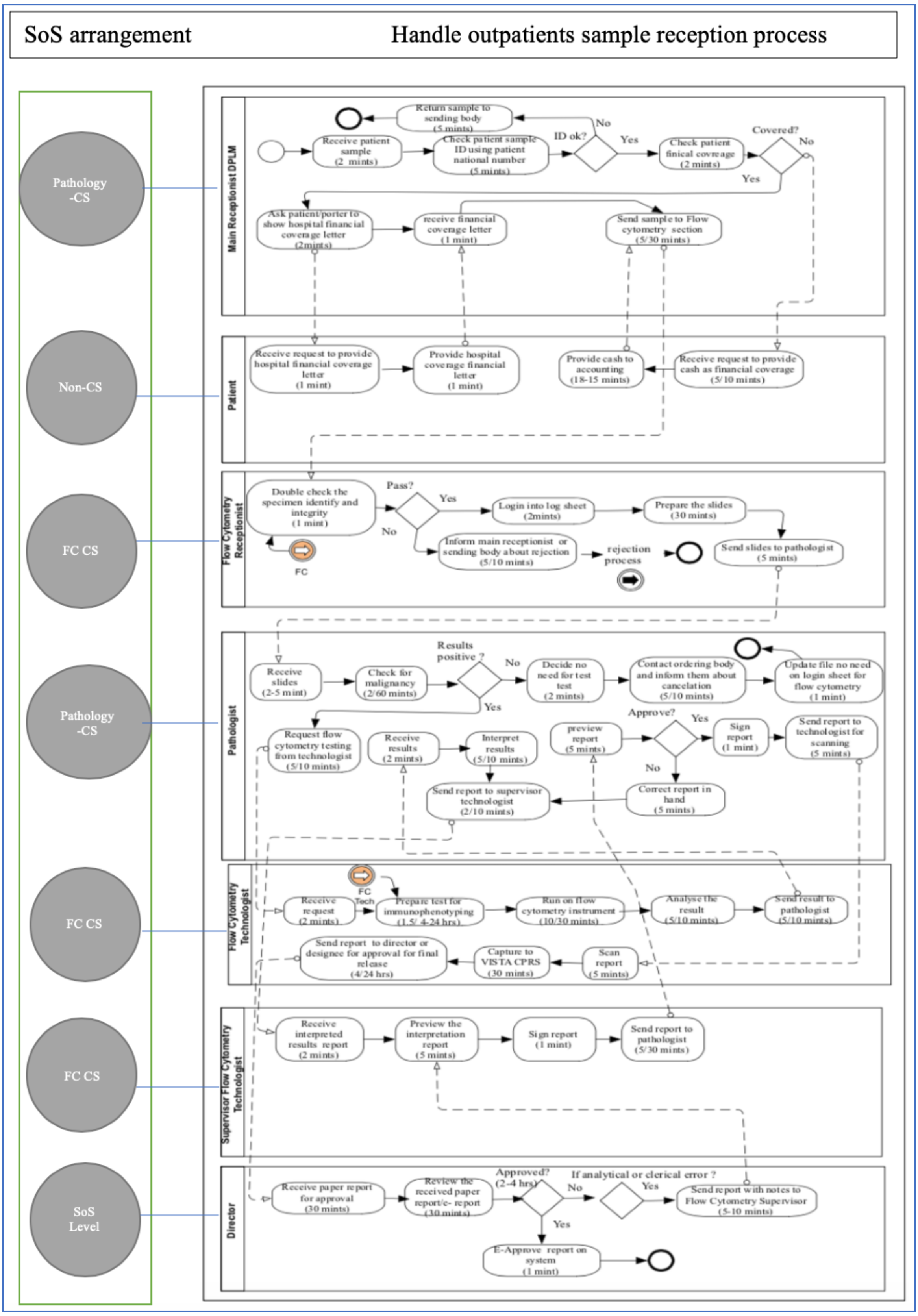

SoS arrangement
Handle outpatients sample reception process
Pathology -CS
Non-CS
FC CS
Pathology -CS
FC CS
FC CS
SoS Level
Main Receptionist DPLM
Return sample to sending body (5 mints)
Receive patient sample (2 mints)
Check patient sample ID using patient national number (5 mints)
ID ok?
No
Yes
Check patient finical covreage (2 mints)
Covered?
No
Yes
Ask patient/porter to show hospital financial coverage letter (2mints)
receive financial coverage letter (1 mint)
Send sample to Flow cytometry section (5/30 mints)
Patient
Receive request to provide hospital financial coverage letter (1 mint)
Provide hospital coverage financial letter (1 mint)
Provide cash to accounting (18-15 mints)
Receive request to provide cash as financial coverage (5/10 mints)
Flow Cytometry Receptionist
Double check the specimen identify and integrity (1 mint)
FC
Pass?
Yes
No
Login into log sheet (2mints)
Prepare the slides (30 mints)
Inform main receptionist or sending body about rejection (5/10 mints)
rejection process
Send slides to pathologist (5 mints)
Pathologist
Receive slides (2-5 mint)
Check for malignancy (2/60 mints)
Results positive ?
No
Yes
Decide no need for test test (2 mints)
Contact ordering body and inform them about cancelation (5/10 mints)
Update file no need on login sheet for flow cytometry (1 mint)
Request flow cytometry testing from technologist (5/10 mints)
Receive results (2 mints)
Interpret results (5/10 mints)
preview report (5 mints)
Approve?
Yes
No
Sign report (1 mint)
Send report to technologist for scanning (5 mints)
Send report to supervisor technologist (2/10 mints)
Correct report in hand (5 mints)
Flow Cytometry Technologist
FC Tech
Receive request (2 mints)
Prepare test for immunophenotyping (1.5/ 4-24 hrs)
Run on flow cytometry instrument (10/30 mints)
Analyse the result (5/10 mints)
Send result to pathologist (5/10 mints)
Send report to director or designee for approval for final release (4/24 hrs)
Capture to VISTA CPRS (30 mints)
Scan report (5 mints)
Supervisor Flow Cytometry Technologist
Receive interpreted results report (2 mints)
Preview the interpretation report (5 mints)
Sign report (1 mint)
Send report to pathologist (5/30 mints)
Director
Receive paper report for approval (30 mints)
Review the received paper report/e- report (30 mints)
Approved? (2-4 hrs)
No
Yes
If analytical or clerical error ?
Yes
Send report with notes to Flow Cytometry Supervisor (5-10 mints)
E-Approve report on system (1 mint)

Figure 6-5: BPMN model of the process of 'Handling outpatients sample reception' (Odeh et al., 2018) © [2018] IEEE and related systems in the SoS arrangement

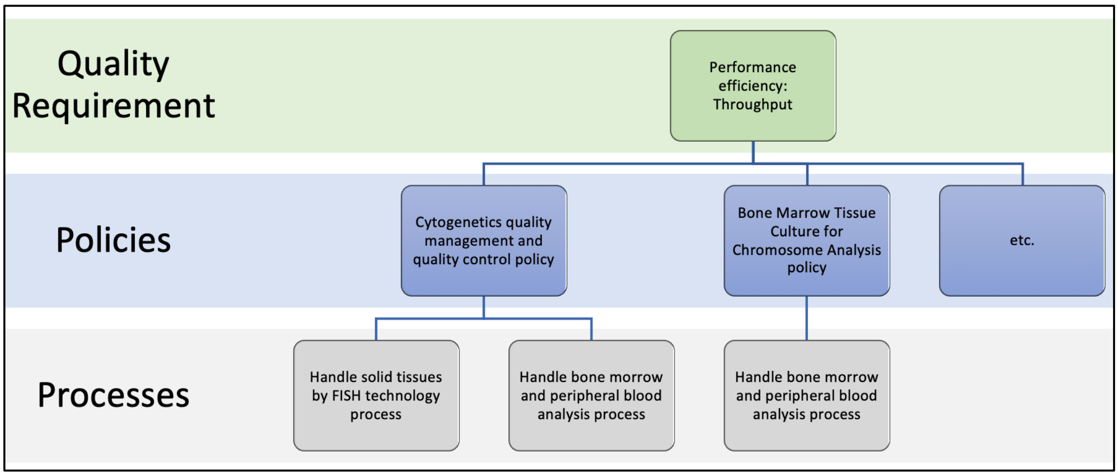

Quality Requirement
Performance efficiency: Throughput
Policies
Cytogenetics quality management and quality control policy
Bone Marrow Tissue Culture for Chromosome Analysis policy
etc.
Processes
Handle solid tissues by FISH technology process
Handle bone morrow and peripheral blood analysis process
Handle bone morrow and peripheral blood analysis process

Figure 6-6: Partial Example of the Interactions in Increment 3- Algorithm 6.1 (Alternative 1)

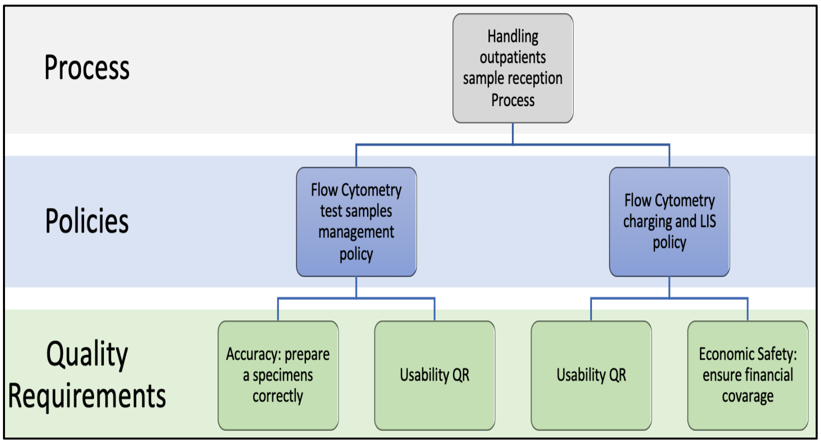

Process
Handling outpatients sample reception Process
Policies
Flow Cytometry test samples management policy
Flow Cytometry charging and LIS policy
Quality Requirements
Accuracy: prepare a specimens correctly
Usability QR
Usability QR
Economic Safety: ensure financial covarage

Figure 6-7: Partial Example of the Interaction in Increment 3- Algorithm 6.2 (Alternative 2)

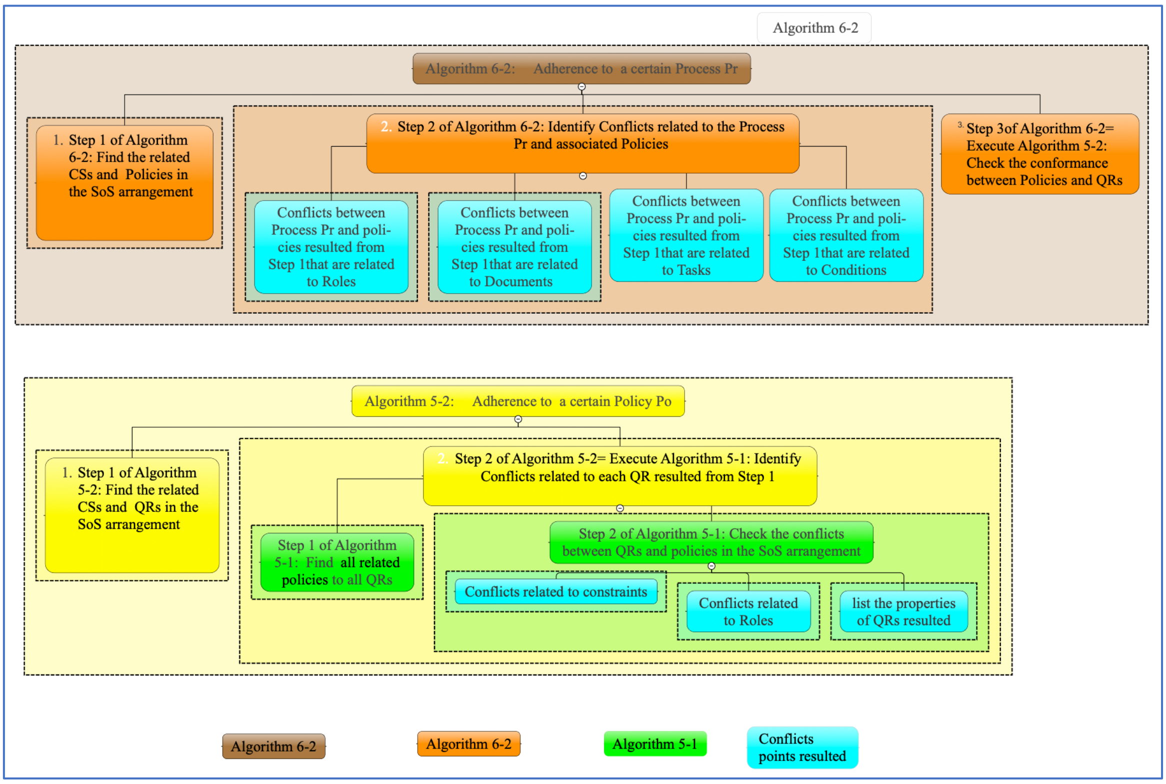

Algorithm 6-2
Algorithm 6-2: Adherence to a certain Process Pr
1. Step 1 of Algorithm 6-2: Find the related CSs and Policies in the SoS arrangement
2. Step 2 of Algorithm 6-2: Identify Conflicts related to the Process Pr and associated Policies
3. Step 3of Algorithm 6-2= Execute Algorithm 5-2: Check the conformance between Policies and QRs
Conflicts between Process Pr and policies resulted from Step 1that are related to Roles
Conflicts between Process Pr and policies resulted from Step 1that are related to Documents
Conflicts between Process Pr and policies resulted from Step 1that are related to Tasks
Conflicts between Process Pr and policies resulted from Step 1that are related to Conditions
Algorithm 5-2: Adherence to a certain Policy Po
1. Step 1 of Algorithm 5-2: Find the related CSs and QRs in the SoS arrangement
2. Step 2 of Algorithm 5-2= Execute Algorithm 5-1: Identify Conflicts related to each QR resulted from Step 1
Step 1 of Algorithm 5-1: Find all related policies to all QRs
Step 2 of Algorithm 5-1: Check the conflicts between QRs and policies in the SoS arrangement
Conflicts related to constraints
Conflicts related to Roles
list the properties of QRs resulted
Algorithm 6-2
Algorithm 6-2
Algorithm 5-1
Conflicts points resulted

Figure 6-8: Conceptualisation Process of Algorithm 6-2

6.4. The Evaluation of Increment 3

An evaluation process was conducted to evaluate the process model and then, to validate the relation and the interaction between processes, policies and QRs. The feedback gained from the interviewees was applied to Increment 3. Thus, at the end of this phase, Increment 3 was validated and the output was the second version of Increment 3.

In order to evaluate the ontology structure of the OntoSoS.Process model and the interaction between processes, policies and QRs. Table 6-3 and Table 6-4 show the main interview questions and the check list used with domain experts from KHCC. The interviewees were selected based on their ability to provide the necessary knowledge and had experience in the area of processes, modelling, policies, conflicts and quality requirements. They were able to provide useful and credible information to this research. Also, they expressed their satisfaction of providing the time needed for their participation in interviews.

Table 6-3: Checklist to Validate the Processes Ontology Model Using Walkthrough Approach

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
1	Validate the concept "process" and its Data Properties (DP) and Object Properties (OP).	Concept: Process	*		
		DP: process Name	*		
		DP: process Owner	*		No need to have it in the KHCC case study
		DP: process description	*		
		DP: process Purpose	*		
		DP: number	*		
2	Validate the concept "Role" and its properties	Concept: Role	*		It was suggested to add the constraint (0 or more) on the recursive relation 'Role interacts with another Role ' to become 'Role may interact with none or more roles'
		DP: name	*		

	Goal	**Concepts and properties**	**Result**		**Remarks from Domain Experts**
			Satisfied	unsatisfied	
		DP: description	*		
3	Validate the concept “Staff” and its Properties	Class: Staff	*		
		OP: is_A	*		
4	Validate the concept “Tool” and its Properties	Class: Tool	*		
		OP: is_A	*		
5	Validate the concept “Patient” and its Properties	Class: patient	*		Carer can be included as well
		OP: is_A	*		
6	Validate the concept “Data Object” and its properties	Concept: Data Object	*		
		DP: Name	*		
		DP: description	*		
		DP: has Status	*		
47	Validate the concept “Task” and its properties	Concept: Task	*		
		DP: name	*		
		DP: task Type	*		
8	Validate the concept “Gateway” and its properties	Concept: Gateway	*		
		DP: name	*		
		OP: is connected By	*		
9	Validate the concept “Event” and its properties	Concept: Event	*		
		DP: description	*		
10	Validate the concept “Sequence Flow” and its Properties	Concept: Sequence Flow	*		
		DP: description	*		
11	Validate the concept “Interaction” and its Properties	Concept: Interaction	*		
		DP: description	*		
12	Validate the concept “System” and its Properties	Concept: System	*		

	Goal	Concepts and properties	Result		Remarks from Domain Experts
			Satisfied	unsatisfied	
		DP: name	*		
13	Validate the concept “ConstituentSystem” and its Properties	Concept: ConstituentSystem	*		
		OP: is_A	*		
14	Validate the concept “NonConstituentSystem” and its Properties	Concept: NonConstituentSystem	*		
		OP: is_A	*		
15	Validate the concept “SoS_Level” and its Properties	Concept SoS_Level	*		
		OP: is_A	*		

Table 6-4: The Outcomes of the Main Questions of the Interviews Conducted with Domain Experts to Validate *SoS Processes Model and the Relations with Other Models in Increment* 3

	Questions	Notes
1	Do you agree that the above concepts and associated properties **completely** represent the KHCC’s process related concepts?	√
2	Are there any missing concepts or properties?	The concept Carer as a Role
3	Are there any other properties that could enhance the model?	No
4	Are there any extra concepts or properties that would be better removed from the model?	It was suggested to remove the property ‘process owner’, as it is not applicable at the case study
5	Do you agree that the above concepts and associated properties **correctly** represent the KHCC’s process related concepts?	√
6	Are the data types that are used to describe the Data Properties correct?	√

7	Are the relations (Object Properties) that are used to connect the concepts of the process model correct?	√
8	Are the domain and the range for the relations between the concepts correct?	√
9	Are the cardinality constraints of the relations correct?	The 'Role may interact with none or more roles' was suggested
10	Do you agree that the above concepts and associated properties are **consistent** with each other and do not lead to conflict (free of contradiction with other components in the model)?	√
11	Do you agree with the relation between the policy model and the processes model?	√
12	Do you agree with the process (Algorithms) of interaction provided?	√ The interaction needs to provide more information regarding the conformance with the applied quality- related standards. This is to check that the procedures identified in the policies and corresponding processes are conform with the followed standards.
13	Do you suggest any further concepts/ changes to be applied on increment 3?	NA

As reflected from Tables 6-3 and 6-4, the main concepts of the OntoSoS.Process model were validated by domain experts. Only few amendments were suggested that were implemented. For instance, the relation 'Role *interact with* another Role' was suggested to indicate the set of roles that interact together. Other suggested examples were the relations 'Interaction *interact By* Role' and 'Data Object *usedBy/output* Task'.

Based on the evaluation conducted in the third DSRM iteration, the following conclusions were made:

i) The evaluation criteria (i.e. correctness, consistency and completeness) were satisfied. Using semi-structured interviews, walkthrough and checklists of the OntoSoS.Process ontology model and its linking to the OntoSoS.Policy and the OntoSoS.QR models resulted in domain experts' satisfaction, and they were completely and correctly captured by the ontology. Also, all concepts are structurally and logically consistent (i.e. free of contradiction with other components in the ontology), which was also confirmed by applying consistency checks using a reasoner tool installed with the Protégé ontology editor (the results of the evaluation support answering RQ4).

ii) Reusing some existing ontologies in the literature helped building the OntoSoS.Process ontology model. Domain experts were satisfied regarding the extent of the completeness and correctness of the process ontology model and its concepts, which were captured from the definitions proposed in the ontological process models of (Fan et al., 2016; Nicola et al., 2007; Rospocher et al., 2014). The concepts of the BPMN were used by these authors to build their ontology process models. This demonstrated that using ontology was suitable to define all processes-related aspects (this supports answering RQ2).

iii) It was shown in Table 6-4, point 12, that the process provided in Algorithms 6-1 and 6-2 lacked capturing the quality- related standards to check that the procedures identified in the policies and the corresponding processes conform to the applied standards. So, this has paved the road to start the next DSRM increment which includes interactions that link standards model with the procedures identified in the policies and their corresponding processes. Chapter 7 details the development of the OntoSoS.Stand ontology model and its interactions with QRs, policies and process ontology models.

iv) Some minor amendments were suggested by domain experts from the case study, which were used to enhance the OntoSoS.Process model. The next section reflects these suggestions.

6.5. Revisiting Increment 3

After the validation of the OntoSoS.Process model and the interaction between processes, policies and QRs models, some minor amendments were applied to produce the second version of Increment 3, which is coloured in orange as depicted in Figure 6-9. Some object properties (i.e. relations) and their cardinalities (i.e. constraints) were added, for instance, the addition of an optional recursive relation to reflect the interaction with another Role.

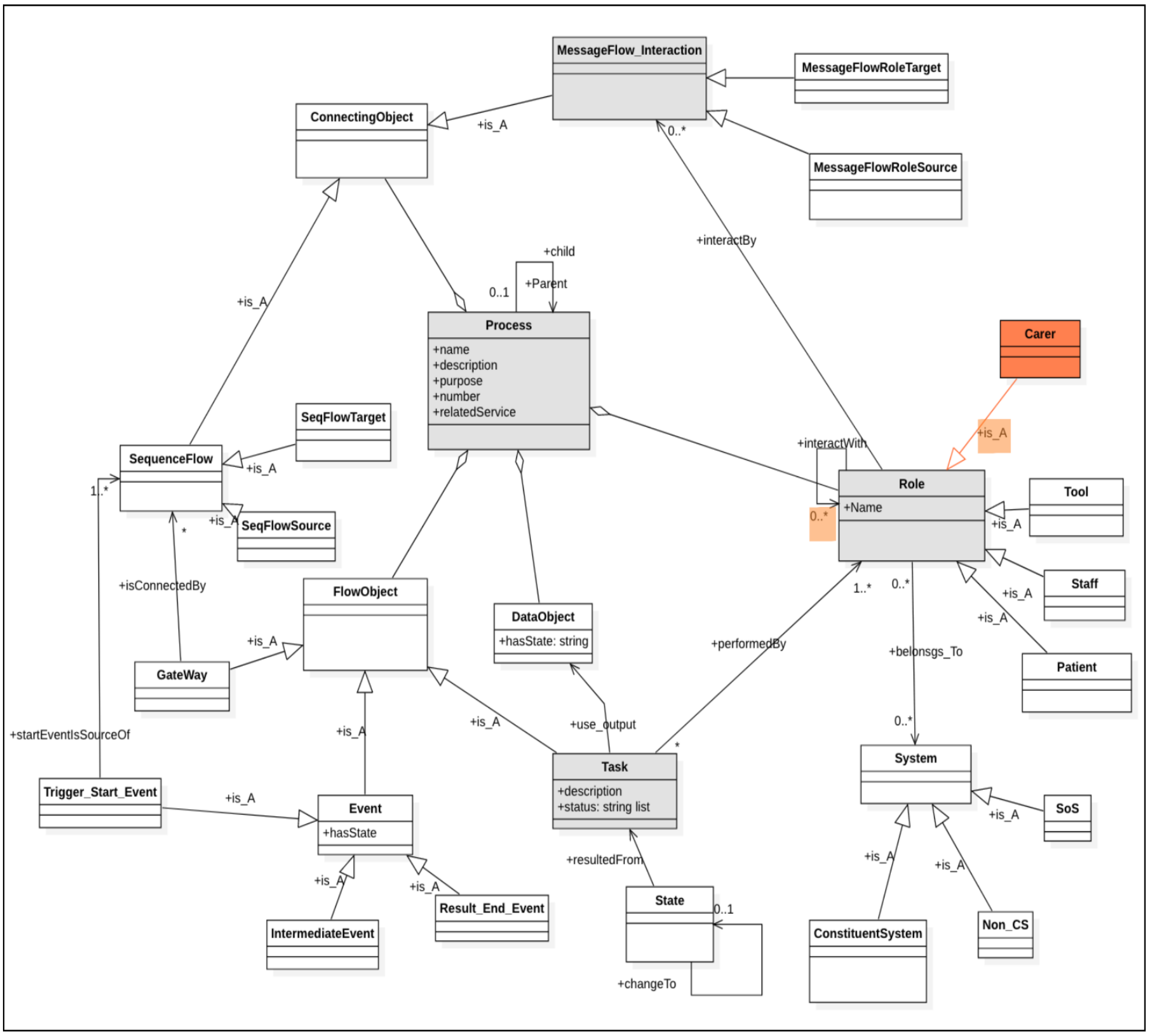


Figure 6-9: The Conceptual Representation of Version 2 of OntoSoS.Process

By the end of DSRM iteration 3, Table 6-5 provides an overview of the status towards answering the research questions. The tick symbol (✓) shows that the research question/sub-question has been answered, and the hourglass with flowing sand symbol (⏳) shows that the status of answering the research question/sub-question is still in progress and the question has only been partially answered so far.

Table 6-5:Status towards Answering the Research Questions

RQ	Main RQ and RQs-Concerns (sub-questions)	Status	Notes	Chapter
RQ1	*What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*	✓		2
1.1	Can research gap analysis, by surveying the literature, identify these challenges ?	✓		2
RQ2	*How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*	⏳		2, 4,5,6 & 7
2.1	What are the components of the quality governance framework, and what are the initial specifications of each component ?	✓		2
2.2	Is ontology suitable to define all related aspects to quality governance ?			7
2.3	What are the elements/concepts of the ontology? Are there any ontologies that we can reuse ?	⏳	See Tables Table 4-1, Table 4-2, Table 5-1, and Table 6-1	4,5,6 & 7
RQ3	*How will the semantically-enriched models of policies, processes, standards and quality requirements interact in the systems of systems context to identify and resolve semantic heterogeneities?*	⏳		5,6 & 7
3.1	Can we develop a process to detail the interaction between *policies, processes, standards and quality requirements to identifying and resolve semantic heterogeneities* ?	⏳	See Algorithms Algorithm 5-1, Algorithm *5-2*, Algorithm 6-1, and Algorithm *6-2*	5,6 & 7
3.2	Are there any limitations of using an ontology-based approach that restrict the interactions between *policies, processes, standards and quality requirements* ?	⏳		5,6 & 7
RQ4	*How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*	⏳		4,5,6 & 7
4.1	How the OntoSoS.QM.Gov ontology will be assessed ?		General evaluation process presented in Chapter 3	7
4.2	Can we validate each ontology model (i.e. each component of the quality governance framework) and then to validate the whole quality governance ontology framework ?	⏳	See Sections 4.4, 5.4, and 6.4	4,5,6 & 7

6.6. Summary and Discussion

After reviewing the literature, many process models that use the concepts of BPMN modelling notations exist. They were adapted to develop the OntoSoS.Process model. It was combined with the previously developed increment (i.e. QR and policy models) to formulate the third increment of the quality governance framework and then this increment was validated with domain experts from KHCC.

The feedback obtained from domain experts was applied to produce a further version of the third DSRM increment. However, only minor amendments were required to be reflected in the design.

After examining the interaction between processes, policies and QRs, it was inferred that such interaction was not sufficient to check the adherence to quality governance. Also, no direct connection between processes and QRs is formally existed. Such interaction, on the other hand, could exist via the standards. This is consistent with the proposed definition of quality governance and consistent with the DSRM iterations followed during this research.

After revisiting the model, the third DSRM iteration was completed and the updated version of the combined models (i.e. OntoSoS.QR, OntoSoS.Policy and OntoSoS.Process) act as an input to the fourth DSRM iteration, which will be discussed in the Chapter 7.

7. Chapter Seven: The Standards Ontology Model and Its Interaction in the OntoSoS.QM.Gov Framework (DSRM Process Increment IV)

7.1. Introduction

This chapter details the fourth and last increment of the DSRM process that has been followed in this research to incrementally develop the OntoSoS.QM.Gov framework. Firstly, the design of the SoS ontology standards model (i.e. OntoSoS.Stand model) is described in Section **Error! Reference source not found.**. Then, the interaction between the previously developed models (i.e. the OntoSoS.Process, the OntoSoS.QR and the OntoSoS.Policy) and the new OntoSoS.Stand model is described in Section 7.2; followed by the demonstration phase of Increment 4 using the cancer care research case study in Section 7.3. After this, the evaluation phase of Increment 4 is described in Section 7.4. Two evaluation steps have been followed; the first step is aimed at examining the structural correctness of the SoS standards ontology model (i.e. OntoSoS.Stand) and the linking relationships with the OntoSoS.QR, OntoSoS.Policy and OntoSoS.Process models. Then, the resulting feedback reported by domain experts has been used to revisit the design of Increment 4 (as shown in Section 7.5). Subsequently, a second evaluation step was conducted to validate the effectiveness of the second version of the OntoSoS.QM.Gov model compared to the quality governance process used at KHCC as shown in Section 7.6.

7.2. The Design of the OntoSoS.Stand Ontology Model Linked to Increment 3

The first step to design the OntoSoS.Stand model is to determine the domain and the scope of the model. In this research, the standards model includes the main component of a standard without specific details regarding one particular standard. Secondly, the ontology standards models in the literature were studied in order to inform the existence of an ontological model to be adopted. However, most of the models in the literature were built to represent a specific standard. Hence, the models in the literature were investigated in order to identify the common concepts of the SoS standards model. The main concepts that were utilised in this research have been defined by (Alexander, 2005; Castillo-Barrera et al., 2013; Howarth and Watson, 2011; *INCOSE*, 2015; "ISO 9001," 2015; "ISO/IEC 17065," 2012; "ISO/IEC 21827," 2008; "ISO/IEC Guide 65," 1996; Sommerville, 2010).

After the main concepts were determined, other related standards concepts are needed to organise and link to other concepts to form a class hierarchy. For instance, the main concept 'Resource' has four sub-classes: 'Human/Role' resources, 'Software', 'Hardware/equipment' and 'Financial recourses'. The details of the concepts and the traceability of concepts to the literature are provided in Table 7-1 and the conceptual representation of the OntoSoS.Stand model is depicted in Figure 7-1. The next two steps relate to the data and object properties (i.e. slots) of the classes and any restrictions on these slots. For example, the relation 'Standard *specifies one or more* Goals' ("ISO/IEC 21827," 2008) indicates that the object property '*specifies*' links the concepts 'Standard' and 'Goal' together. Also, this relation has restrictions which suggests that a standard can be linked with one or more quality- related goals. Another example is the relation between the concepts 'Condition' and 'Standard' which has 'One or more Conditions *need to be satisfied* to implement a Standard' (Castillo-Barrera et al., 2013; Howarth and Watson, 2011).

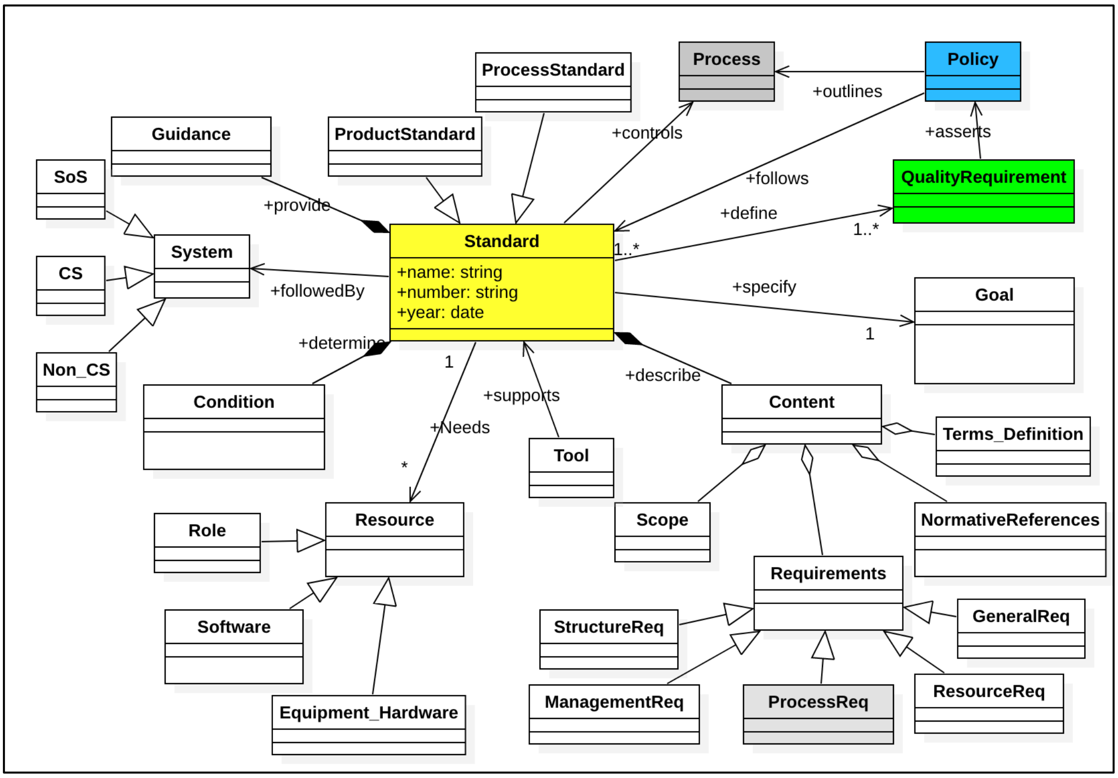


Figure 7-1: The Conceptual Representation of Version 1 of the OntoSoS.Stand Ontology Model

Table 7-1: The OntoSoS.Stand Main Concepts and Traceability to Literature Sources

	Concept	Main Sources and Description	Properties and their types
1	Standard	The root element that indicates a standard. It mainly has two categories: product standards and process standard (Sommerville, 2011)	-**name**: string. The name of the standard or the branch of a standard.
			-**number**: string. The unique number of the standard.
			-**standard year**: date. The year of the used standard.
			-**version**: string. The version of the standard.
			- **description**: string. The description of a standard.
			-**satisfy Goal:** string. The related goal or sub-goal of a standard.
2	Tool	It is a software tool that supports the deployment of a standard (Sommerville, 2011, p. 660). Also, it could be a hand tool (*INCOSE*, 2015)	-**name**: string. The name of the tool.
			-**support**: relation as the tool supports a standard **.**
3	Resource	The standard needs many resources to be applied. For instance, Personal resources to adopt the different roles (Castillo-Barrera et al., 2013; Howarth and Watson, 2011), Equipment resources e.g. laboratory equipment, Financial resources to cover the needed cost (Alexander, 2005; *INCOSE*, 2015; "ISO 9001," 2015).	-**name**: string. The title of the resource.
			-**description**: string. The description of the resource.
4	Guidance	The standards provide guidance to be followed by the institution to enable the adherence to a standard (Sommerville, 2011)	- **description**: string. The description of the Guidance described by the standard.
			-**hasMandatory**: string list {mandatory, optional}. The mandatory status of the guidance to be followed.
5	Goal	The main goal of the branch of a standard ("ISO/IEC 21827," 2008)	-**name**: string. The name of the goal.
6	Condition	The conditions to be satisfied to implement the standard and then, to support adherence to it (Castillo-Barrera et al., 2013; Howarth and Watson, 2011; Sommerville, 2011, p. 98)	- **description**: string. The description of the condition.
7	Content	The basic points to be described by a standard. The content consists of Scope, Normative References, Terms Definitions, General Requirements, Structural Requirements, Resource Requirements, Process Requirements and Management System Requirements ("ISO/IEC 17065," 2012; "ISO/IEC Guide 65," 1996)	- **description**: string. The description of the content.
8	System	A standard may belong to a higher level i.e. SoS level or to a constituent system or to a system from outside the boundary of the SoS arrangement.	-**name**: string. The name of the system.
9	SoS	Brainstorming of concepts: A standard may belong to a higher level i.e. SoS level.	-**name**: string. The name of the system/ SoS.

10	CS	Brainstorming of concepts: A standard may belong to a constituent system	**-name**: string. The name of the system.
11	Non_CS	Brainstorming of concepts: A standard may belong to a system from outside the boundary of the SoS arrangement.	**-name**: string. The name of the system.

As depicted in Figures Figure *6-9* and Figure 7-1, the models of Increment 3 (i.e. the OntoSoS.QR, the OntoSoS.Process and the OntoSoS.Policy ontology models) and the OntoSoS.Stand ontology model are linked together. The main relationships between the models are depicted in Figures Figure *7-2* and

Figure 7-3, where are; 'Standard' is *followed by one or more* 'Policies', 'Quality Requirement' *conforms to* 'Standards' and a 'Standard' *relates to one or more* 'Processes'. The relationship between the process and the policy models is illustrated in Chapter 6 and the relation between the quality requirements and the policy models was illustrated in the Chapter 5.

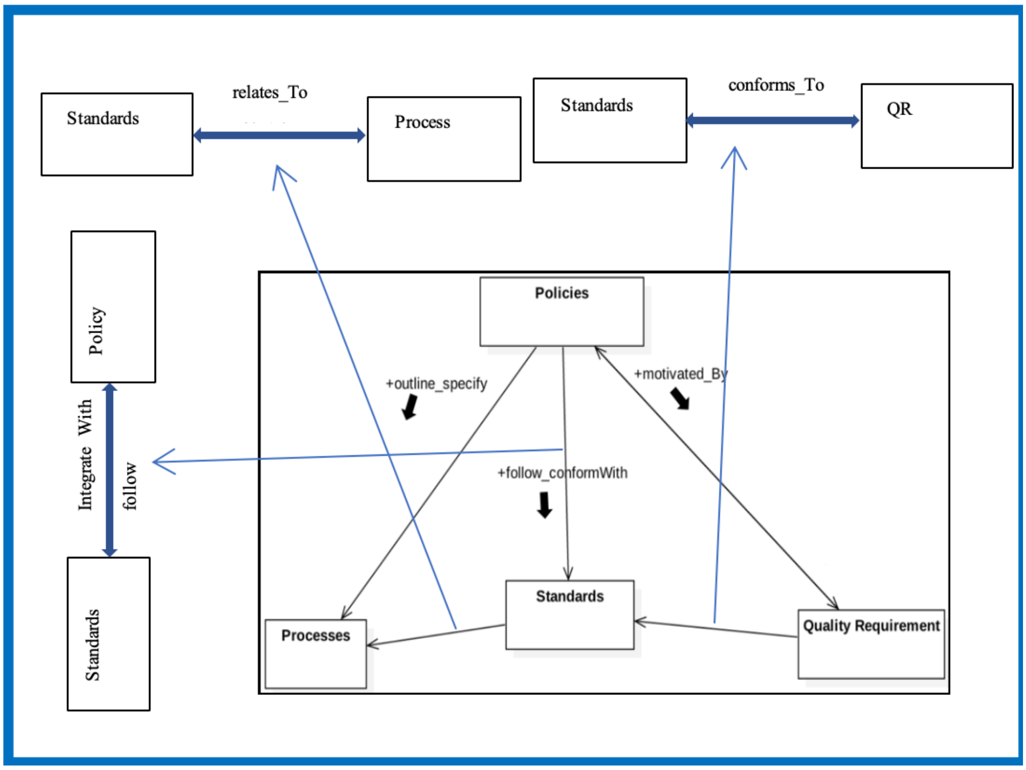


Figure 7-2: Abstract Level of the Proposed Quality Governance Model with Focus on the Links between the Standards Model and other Models

In the SoS and CSs levels, software quality requirements need to be consistent with their related policies and standards (Robinson, 2006). The standards '*define*' the QR(s) and the standards need several resources of personal, financial and tools to be deployed to achieve the required level of quality. Also, the policies and the processes in all CSs need '*to be aligned with/ to follow'* the standards in order to achieve the required quality.

Figure 7-3 shows the linking relations between the four models (coloured in pink) (Appendix IV shows a bigger view for Figure 7-3). For instance, the artefacts such as the documents used by a role to perform a task or the artefacts resulting after performing a task need to be similar in the models in all CSs.

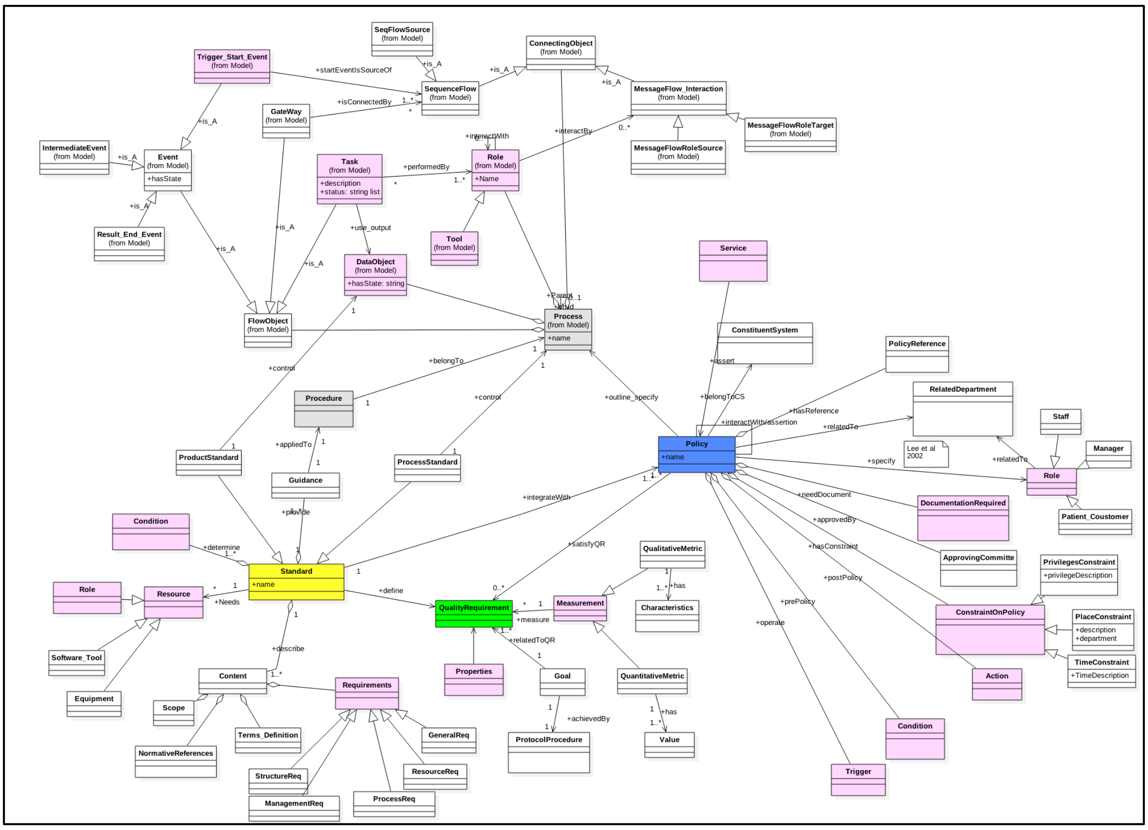


Figure 7-3: A Snapshot of the Conceptual View of Increment 4

In order to check the adherence to a standard, the process detailing the interaction between the four governance component models in the SoS arrangement is described in Algorithms Algorithm 7-1, Algorithm 7-2, Algorithm 7-3 and 7-4. These four algorithms represent the possible alternatives to show the interaction between the four SoS quality governance models. The demonstration of theses algorithms is shown in Section 0 using examples from the research case study.

Algorithm 7-1: The Interaction between SoS Standards, Quality Rrequirements, Policies and Processes Models (Alternative 1: QR-Policies-Processes-Standards)

Input:

QR$_i$ = The quality requirement that we need to check governance for.

Output:

/* Output of Step 1 (i.e. Output of Algorithm 6-1: interaction between SoS QR, policies and processes)*/

CS.Policy_Set= { CS$_1$.Po$_1$, CS$_1$.Po$_2$, …, CS$_2$.Po$_1$, …, CS$_i$.Po$_j$, …, CS$_n$.Po$_p$}, the set of all policies Po$_j$ that are related to QR$_i$ in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number of all policies in each CS.

Conflicts_Poicy_QR_Set= {CS$_1$.Role$_1$, CS$_2$.Role$_2$, CS$_1$.Constraint$_1$, CS$_2$.Constraint$_2$, CS$_1$.Service$_1$, … , CS$_i$.Role$_j$, CS$_i$.Constraint$_k$, CS$_i$.Service$_x$, …, CS$_n$.Role$_r$, CS$_i$.Constraint$_c$, CS$_i$.Service$_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each CS; and $0<x<=s$, c=number of all services in each CS.

CS.QR.Properties_Set ={ CS$_1$.QR$_1$.Properties$_1$, CS$_1$.QR$_2$.Properties$_2$, …, CS$_i$.QR$_j$.Properties$_k$, …, CS$_n$. QR$_q$.Properties$_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$, r=number of all Properties in each QR.

CS.Process_Set= { CS$_1$.Pr$_1$, CS$_1$.Pr$_2$, …, CS$_2$.Pr$_1$, …, CS$_i$.Pr$_j$, …, CS$_n$.Pr$_p$}, the set of all processes Pr$_j$ that are related to each policy in CS.Policy_Set in all constituent systems CS , where $0 < i <= n$, n=number of CSs, $0<j<=p$, p=number of all processes in each CS.

Conflicts_Policy_Process_Set={CS$_1$.Role$_1$, CS$_1$.Task$_1$, CS$_2$.Documentation$_1$, CS$_1$.Action$_1$, CS$_1$.DataObject$_1$, …, CS$_i$. Role$_j$, CS$_i$.Task$_k$, CS$_i$.Action$_l$, CS$_i$.DataObject$_m$, CS$_i$. Documentation$_n$, …,CS$_n$. Role$_r$, CS$_i$.Task$_t$, CS$_i$.Action$_a$, CS$_i$.DataObject$_o$, CS$_i$. Documentation$_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=t$, t=number of all tasks in each CS; $0<l<=a$, a=number of all actions in each CS; $0<m<=o$, o=number of all data objects in each CS; $0<n<=d$, d=number of all documents in each CS;

/* Other outputs*/

Standards set St_Set = {CS$_1$.St$_1$, CS$_1$.St$_2$, …, CS$_i$.St$_j$, …, CS$_n$.St$_s$), where $0 < i <= n$, n=number of CSs; $0<j<=s$, s=number of all standards in each CS.

Conflicts_Standard_Process_Set= {CS$_1$.Role$_1$, CS$_1$.Documentation$_1$, CS$_2$.Condition$_1$, CS$_1$.Tool$_1$,…, CS$_i$. Role$_j$, CS$_i$. Documentation$_k$, CS$_i$. Condition$_l$, CS$_i$.Tool$_m$, …,CS$_i$. Role$_r$, CS$_i$. Documentation$_d$, CS$_i$. Condition$_c$, CS$_i$.Tool$_t$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=d$, d=number of all documents in each CS; $0<l<=c$, c=number of all conditions in each CS; $0<m<=t$, t=number of all tools in each CS.

```
Begin
/*Step 1: check the adherence of the quality requirement QR$_i$ to its related policies and processes*/
Execute Algorithm 6-1.
/*Step 2: Find all related standards to each process resulting from Step1*/
For each process Pr$_k$ in Pr_Set resulted from step 1 do
    For each constituent system CS$_i$ in all CSs do
        For each Standards St$_j$ in each CS$_i$ do
            Find all standards CS$_i$.St$_j$, where CS$_i$.Pr$_k$ <<controlled By>> CS$_i$.St$_j$
                add CS$_i$.St$_j$ from previous point to the standards set St_Set
        End for
    End for
End for
/*Step 3: Identify conflicts*/
    For each process Pr$_k$ in Pr_Set resulted from step1 do
        For each standard St$_j$ in Pt_Set resulted from step 2 do
        {   For each Role in each process do
                find and compare between St$_j$. Role. 'description' and Both (Pr$_k$. Role 'description' and Pr$_k$.'process's Owner').
                If does not match add results to Conflicts_Standard_Process_Set.
                End for
            For each documents in each process do
                find and compare between St$_j$. Documentation_ Requirement. 'description' and Pr$_k$. DataObject. 'description'.
                If does not match add results to Conflicts_Standard_Process_Set.
                End for
            For each Tool in each process do
                find and compare between St$_j$. Tool. 'description' and Pr$_k$. Tool. 'description'
                If does not match add results to Conflicts_Standard_Process_Set.
            End for
            For each Condition in each standard do
                find and compare between St$_j$. Condition. 'description' and Pr$_k$. Event. 'description'
                If does not match add results to Conflicts_Standard_Process_Set.
            End for
        }End for
    End for
```

Algorithm 7-2: The Interaction between SoS Standards, Quality Rrequirements, Policies and Processes (Alternative 2: Standard- QRs- Policies-Processes)

Input:

St_i = The standard that we need to check governance for.

Output:

/*See Output of Step 2 (i.e. Output of Algorithm 7-1: interaction between SoS QR, policies, processes and standards)*/

CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies Po_j that are related to QR_i in all constituent systems CS , where 0 < i <= n, n=number of CSs; 0<j<=p, p=number of all policies in each CS.

Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=c, c=number of all constraints in each CS; and 0<x<=s, c=number of all services in each CS.

CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, CS_n. $QR_q.Properties_r$}, where 0 < i <=n, n= number of CSs; 0<j<=q, q=number of all QRs in each CS; 0<k<=r, r=number of all Properties in each QR.

CS.Process_Set= { $CS_1.Pr_1$, $CS_1.Pr_2$, …, $CS_2.Pr_1$, …, $CS_i.Pr_j$, …, $CS_n.Pr_p$}, the set of all processes Pr_j that are related to each policy in CS.Policy_Set in all constituent systems CS , where 0 < i <= n, n=number of CSs, 0<j<=p, p=number of all processes in each CS.

Conflicts_Policy_Process_Set={$CS_1.Role_1$, $CS_1.Task_1$, $CS_2.Documentation_1$, $CS_1.Action_1$, $CS_1.DataObject_1$, …, CS_i. $Role_j$, $CS_i.Task_k$, $CS_i.Action_l$, $CS_i.DataObject_m$, CS_i. $Documentation_n$, …,CS_n. $Role_r$, $CS_i.Task_t$, $CS_i.Action_a$, $CS_i.DataObject_o$, CS_i. $Documentation_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=t, t=number of all tasks in each CS; 0<l<=a, a=number of all actions in each CS; 0<m<=o, o=number of all data objects in each CS; 0<n<=d, d=number of all documents in each CS;

Standards set St_Set = {$CS_1.St_1$, $CS_1.St_2$, …, $CS_i.St_j$, …, $CS_n.St_s$), where 0 < i <= n, n=number of CSs; 0<j<=s, s=number of all standards in each CS.

Conflicts_Standard_Process_Set= {$CS_1.Role_1$, $CS_1.Documentation_1$, $CS_2.Condition_1$, $CS_1.Tool_1$,…, CS_i. $Role_j$, CS_i. $Documentation_k$, CS_i. $Condition_l$, $CS_i.Tool_m$, …,CS_i. $Role_r$, CS_i. $Documentation_d$, CS_i. $Condition_c$, $CS_i.Tool_t$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts with their values, where 0 < i <= n, n=number of CSs; 0<j<=r, r=number of all roles in each CS; 0<k<=d, d=number of all documents in each CS; 0<l<=c, c=number of all conditions in each CS; 0<m<=t, t=number of all tools in each CS.

/* Other outputs*/

Quality requirements set QR_Set = {$CS_1.QR_1$, $CS_1.QR_2$, …, $CS_i.QR_j$, …, $CS_n.QR_q$), where 0 < i <= n, n=number of CSs; 0<j<=q, q=number of all QR in each CS.

```
Begin
/*Step 1: Find all related QRs $QR_j$ in the SoS arrangement to the standard $St_i$ */

 For each constituent system $CS_i$ in all CSs do
      For each quality requirement $QR_j$ in each $CS_i$ do
          Find all QR $CS_i.QR_j$, where $CS_i.QR_j$ <<specified By/defined By>> $CS_i.St_i$
          add $CS_i.QR_j$ resulting from previous point to the QRs set QR_Set.
      End for
 End for

/*Step 2: Identify conflicts*/

For each quality requirement $QR_j$ in QR_Set do

Execute Algorithm 7-1

End for
End
```

Algorithm 7-1 represents the process to check the adherence to SoS quality governance by starting with a certain QR and checking the adherence to this QR among the SoS arrangement, while Algorithm 7-2 illustrates the process to check the adherence to a certain standard among the SoS arrangement by checking if this standard conforms with the policies, the QRs and the processes that follow that standard in all CSs. Any inconsistencies in the common concepts between them in all CSs need to be reported. Also, Algorithm 7-3 illustrates the process to check adherence of a certain process to its related policies, QRs and standards in all CSs in the SoS arrangement. Lastly, Algorithm 7-4 shows the process to check the adherence of a policy among the SoS arrangement.

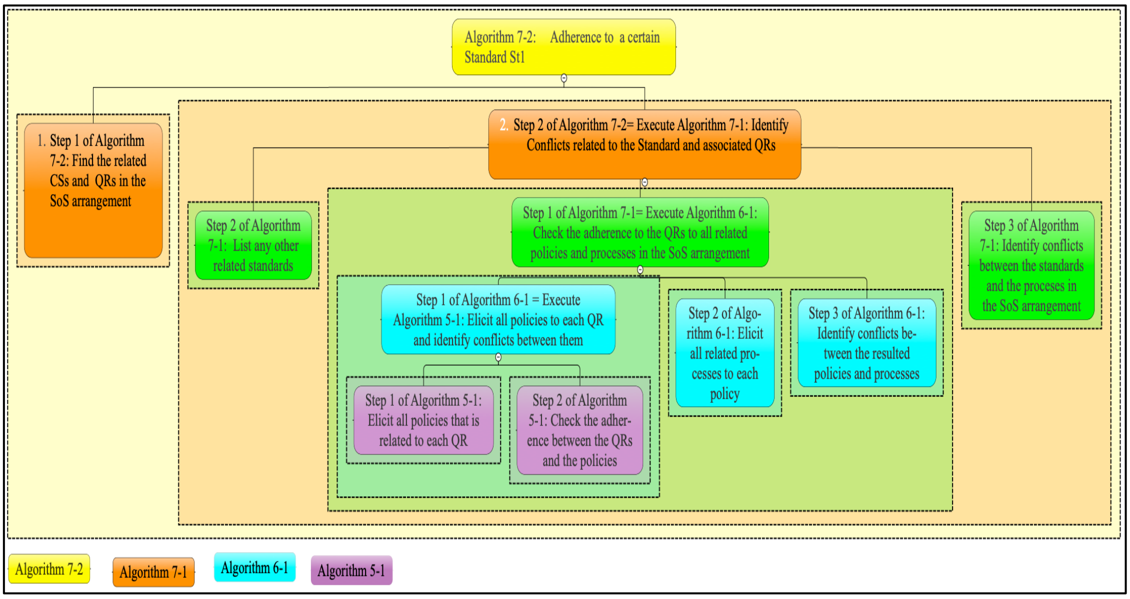

Algorithm 7-2: Adherence to a certain Standard St1
1. Step 1 of Algorithm 7-2: Find the related CSs and QRs in the SoS arrangement
2. Step 2 of Algorithm 7-2= Execute Algorithm 7-1: Identify Conflicts related to the Standard and associated QRs
Step 2 of Algorithm 7-1: List any other related standards
Step 1 of Algorithm 7-1= Execute Algorithm 6-1: Check the adherence to the QRs to all related policies and processes in the SoS arrangement
Step 3 of Algorithm 7-1: Identify conflicts between the standards and the processes in the SoS arrangement
Step 1 of Algorithm 6-1 = Execute Algorithm 5-1: Elicit all policies to each QR and identify conflicts between them
Step 2 of Algorithm 6-1: Elicit all related processes to each policy
Step 3 of Algorithm 6-1: Identify conflicts between the resulted policies and processes
Step 1 of Algorithm 5-1: Elicit all policies that is related to each QR
Step 2 of Algorithm 5-1: Check the adherence between the QRs and the policies
Algorithm 7-2
Algorithm 7-1
Algorithm 6-1
Algorithm 5-1

Figure 7-4: Conceptualisation Process of Algorithm 7-2

One of the possible interactions is to check adherence to a certain standard (e.g. St_i) as explained in Algorithm 7-2. Such interaction is depicted in Figure 7-4 (the direction illustrated in Figure 7-4 can be read from top to down and from left to right). Algorithm 7-2 works as follows; Step 1 of Algorithm 7-2 elicits the QRs that should be satisfied by the standard, and list them in a QRs set (QR_Set). Step 2 of Algorithm 7-2 finds all the conflicts that are resulting from the interactions between the policies and the processes that are supposed to satisfy the standard St_i and the QRs in the QRs set. The execution of the second step of Algorithm 7-2 results in identifying the conflicts between the standard St_i and its related QRs, the Algorithm 7-1 will be executed to identify policies in the SoS arrangement as the related policy set 'policy_Set'. Algorithm 7-1 consists of three main steps. In Step 1 of Algorithm 7-1, all the policies in the SoS arrangement that are related to the QRs need to be listed and then, the conflicts between each policy in the policy set and its related processes in each CS need to be identified. To perform this task, the process described in Algorithm 6-1 will be followed due to the similarity of the functions between Algorithm 6-1 and Step 1 of Algorithm 7-1. Algorithm 6-1 consists of two main steps, step 1 of Algorithm 6-1 is to elicit all the policies that are related to the QRs and to identify the conflicts between them which is the same function of Algorithm 5-1. So, Algorithm 5-1 will be executed. In all former steps, comparisons between similar concepts of every two models will be conducted and the results of the comparisons will be listed in the conflicts sets.

Other options of interactions between standards, processes, policies and quality requirements are provided in Algorithms 7-1, 7-3 and 7-4. Examples regarding all options are depicted in Figures Figure 7-7, Figure 7-8, Figure 7-9, and Figure 7-10. In order to facilitate the understanding of Figure 7-4 and the process depicted in Algorithm 7-2, an example demonstrating the application of Algorithm 7-2 from the research case study is illustrated in Section 7.3.

Algorithm 7-3: The Interaction between Processes, Quality Rrequirements, Policies and Standards (Alternative 3: Process- Policies-QRs- Standards)

Input:

Pr_i = The process that we need to check governance for.

Output:

/*See Output of Step 1 (i.e. Output of Algorithm 6-2: Interaction between SoS processes, policies, and QR) */

CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies Po_j that are related to QR_i in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number of all policies in each CS.

Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each CS; and $0<x<=s$, c=number of all services in each CS.

CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, $CS_n. QR_q.Properties_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$, r=number of all Properties in each QR.

CS.Process_Set= { $CS_1.Pr_1$, $CS_1.Pr_2$, …, $CS_2.Pr_1$, …, $CS_i.Pr_j$, …, $CS_n.Pr_p$}, the set of all processes Pr_j that are related to each policy in CS.Policy_Set in all constituent systems CS , where $0 < i <= n$, n=number of CSs, $0<j<=p$, p=number of all processes in each CS.

Conflicts_Policy_Process_Set={$CS_1.Role_1$, $CS_1.Task_1$, $CS_2.Documentation_1$, $CS_1.Action_1$, $CS_1.DataObject_1$, …, $CS_i. Role_j$, $CS_i.Task_k$, $CS_i.Action_l$, $CS_i.DataObject_m$, $CS_i. Documentation_n$, …,$CS_n. Role_r$, $CS_i.Task_t$, $CS_i.Action_a$, $CS_i.DataObject_o$, $CS_i. Documentation_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=t$, t=number of all tasks in each CS; $0<l<=a$, a=number of all actions in each CS; $0<m<=o$, o=number of all data objects in each CS; $0<n<=d$, d=number of all documents in each CS;

/* Other outputs*/

Standards set St_Set = {$CS_1.St_1$, $CS_1.St_2$, …, $CS_i.St_j$, …, $CS_n.St_s$), where $0 < i <= n$, n=number of CSs; $0<j<=s$, s=number of all standards in each CS.

Conflicts_Standard_Process_Set= {$CS_1.Role_1$, $CS_1.Documentation_1$, $CS_2.Condition_1$, $CS_1.Tool_1$,…, $CS_i. Role_j$, $CS_i. Documentation_k$, $CS_i. Condition_l$, $CS_i.Tool_m$, …,$CS_i. Role_r$, $CS_i. Documentation_d$, $CS_i. Condition_c$, $CS_i.Tool_t$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=d$, d=number of all documents in each CS; $0<l<=c$, c=number of all conditions in each CS; $0<m<=t$, t=number of all tools in each CS.

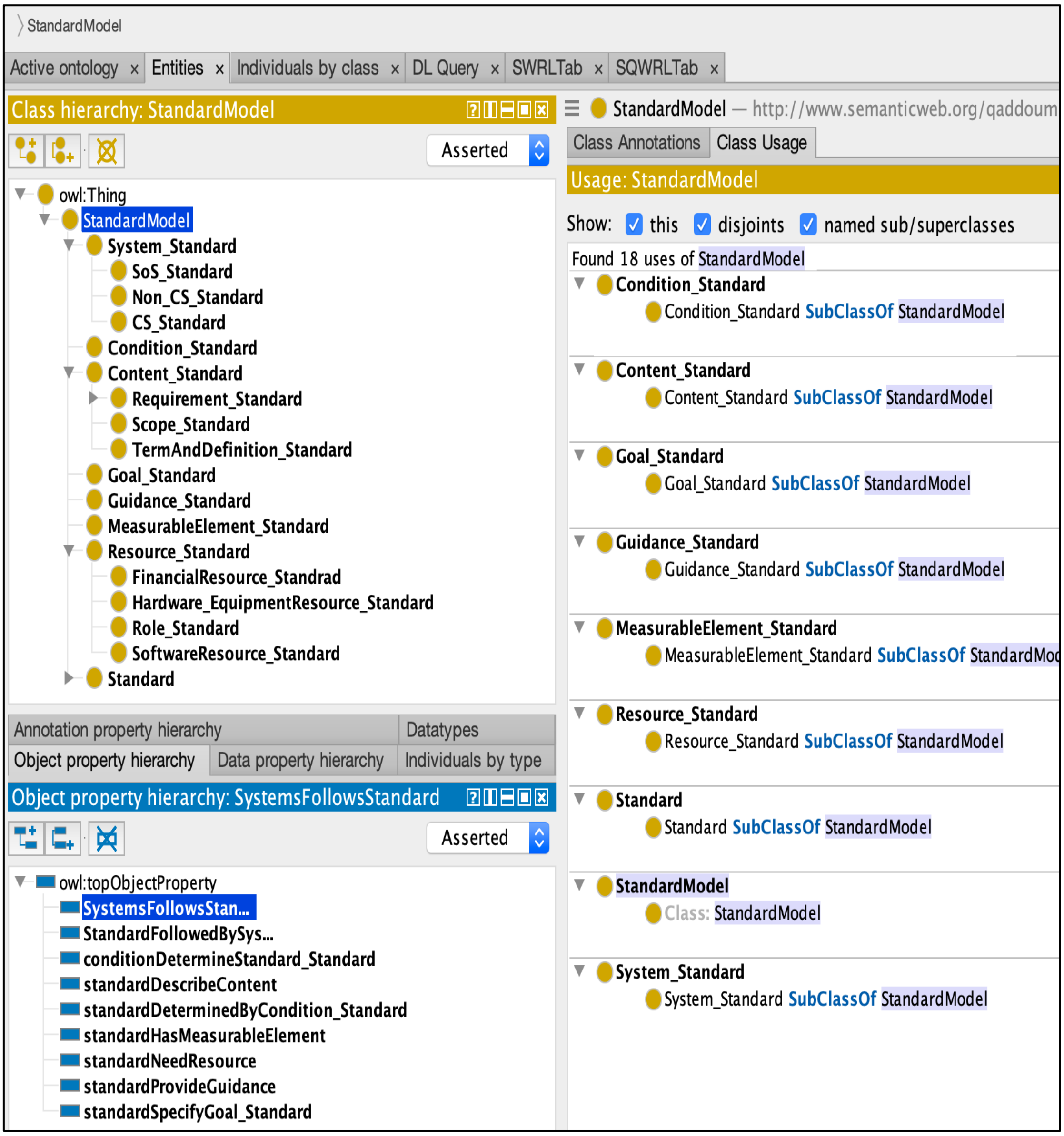
```
Begin
/*Step 1: check the adherence of the process $Pr_i$ to its related policies and processes*/
Execute Algorithm 6-2
/*Step 2: Find all related standards to each QR resulted from Step1
For each Quality Requirement $QR_k$ in QR_Set resulted from step 1 do
    For each constituent system $CS_i$ in all CSs do
        For each Processes $Pr_j$ in each $CS_i$ do
            Find all processes $CS_i.Pr_j$, where $CS_i.Po_k$ <<specify/outline>> $CS_i.Pr_j$
            add $CS_i.Pr_j$ from previous point to the processes set Pr_Set.
        End for
    End for
End for

/*Step 3: Identify conflicts*/
For each quality requirement $QR_k$ in QR_Set resulted from step 1 do
    For each standard $St_j$ in St_Set resulted from step 2 do
    {
        Find and Compare between $St_j$. Resources. Financial: 'quantity'  and $QR_k$. Metric. Constraint. Cost Constraint. 'quantity'.
        if not match add results to Conflicts_Standard_QR_Set
        Find and Compare between both ($St_j$. Resources. Personal. 'description' and $St_j$. Role. 'description') and $QR_k$. Metric. Role. 'description' .
        if not match add results to Conflicts_Standard_QR_Set
        Find and Compare between $St_j$. Content. Requirement. 'description'  and $QR_k$.Standard $QR_k$. Requirement. 'description'.
        if not match add results to Conflicts_Standard_QR_Set
        Find and Compare between St. Product Standard. Artefact. Product/Service. 'description' and QR. Product QR. Product/Service. 'description'.
        if not match add results to Conflicts_Standard_QR_Set
    }
    End for
End for
End
```

Algorithm 7-4: The Interaction between the SoS Standards, Quality Requirements, Policies and Processes Models (Alternative 4: Policy-Processes-Standards-QRs)

```
Input:
    $Po_i$ = The policy that we need to check governance for.
Output:
Quality requirements set QR_Set = {$CS_1.QR_1$, $CS_1.QR_2$, …, $CS_i.QR_j$, …, $CS_n.QR_q$), where  $0 < i <= n$, n=number of CSs; $0<j<=q$, q=number of all QR in each CS.
Standards set St_Set = {$CS_1.St_1$, $CS_1.St_2$, …, $CS_i.St_j$, …, $CS_n.St_s$), where $0 < i <= n$, n=number of CSs; $0<j<=s$, s=number of all standards in each CS.
CS.Policy_Set= { $CS_1.Po_1$, $CS_1.Po_2$, …, $CS_2.Po_1$, …, $CS_i.Po_j$, …, $CS_n.Po_p$}, the set of all policies $Po_j$ that are related to $QR_i$ in all constituent systems CS , where $0 < i <= n$, n=number of CSs; $0<j<=p$, p=number of all policies in each CS.
CS.QR.Properties_Set ={ $CS_1.QR_1.Properties_1$, $CS_1.QR_2.Properties_2$, …, $CS_i.QR_j.Properties_k$, …, $CS_n$. $QR_q.Properties_r$}, where $0 < i <=n$, n= number of CSs; $0<j<=q$, q=number of all QRs in each CS; $0<k<=r$, r=number of all Properties in each QR.
CS.Process_Set= { $CS_1.Pr_1$, $CS_1.Pr_2$, …, $CS_2.Pr_1$, …, $CS_i.Pr_j$, …, $CS_n.Pr_p$}, the set of all processes $Pr_j$ that are related  to each policy in CS.Policy_Set in all constituent systems CS , where $0 < i <= n$, n=number of CSs, $0<j<=p$, p=number of all processes in each CS.
Conflicts_Standard_Process_Set= {$CS_1.Role_1$, $CS_1.Documentation_1$, $CS_2.Condition_1$, $CS_1.Tool_1$,…, $CS_i$. $Role_j$, $CS_i$. $Documentation_k$, $CS_i$. $Condition_l$, $CS_i.Tool_m$ , …,$CS_i$. $Role_r$, $CS_i$. $Documentation_d$, $CS_i$. $Condition_c$, $CS_i.Tool_t$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=d$, d=number of all documents in each CS; $0<l<=c$, c=number of all conditions in each CS; $0<m<=t$, t=number of all tools in each CS.
Conflicts_Poicy_QR_Set= {$CS_1.Role_1$, $CS_2.Role_2$, $CS_1.Constraint_1$, $CS_2.Constraint_2$, $CS_1.Service_1$, … , $CS_i.Role_j$, $CS_i.Constraint_k$, $CS_i.Service_x$, …, $CS_n.Role_r$, $CS_i.Constraint_c$, $CS_i.Service_s$}, the set of all roles, constraints and services in the constituent systems that have conflicts with their values, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=c$, c=number of all constraints in each CS; and  $0<x<=s$, c=number of all services in each CS.
Conflicts_Policy_Process_Set={$CS_1.Role_1$, $CS_1.Task_1$, $CS_2.Documentation_1$, $CS_1.Action_1$, $CS_1.DataObject_1$, …, $CS_i$. $Role_j$, $CS_i.Task_k$, $CS_i.Action_l$, $CS_i.DataObject_m$, $CS_i$. $Documentation_n$, …,$CS_n$. $Role_r$, $CS_i.Task_t$, $CS_i.Action_a$, $CS_i.DataObject_o$, $CS_i$. $Documentation_d$ }, the set of all roles, tasks, actions, data objects and documents in the constituent systems that have conflicts, where $0 < i <= n$, n=number of CSs; $0<j<=r$, r=number of all roles in each CS; $0<k<=t$, t=number of all tasks in each CS; $0<l<=a$, a=number of all actions in each CS; $0<m<=o$, o=number of all data objects in each CS; $0<n<=d$, d=number of all documents in each CS;
Conflicts_Standard_QR_Set= { $CS_1.Role_1$, $CS_1$. Resources $_1$, $CS_2.Cost_1$, $CS_1$. Requirement $_1$, $CS_2.Product_1$, … , $CS_i$. $Role_j$, $CS_i$. Resources $_k$, $CS_i$. Requirement $_l$, $CS_i.Product_m$ , …, $CS_n$. $Role_r$, $CS_i$. Resources $_r$, $CS_i$. Requirement $_q$, $CS_i.Product_p$}, the set of all roles, financial resources, requirements, and products in the constituent systems that have conflicts with their values.
    Begin
    /*Step 1: prepare the policy set */
    add the entered policy $Po_i$ to the policy set Po_Set
    /*Step 2: select related processes*/
    Execute Step 2 of Algorithm 6.1
    /*Step 3:  interaction between policies and processes*/
    Execute Step 3 of Algorithm 6.1
    /*Step 4: select related standards for each process*/
    Execute Step 2 of Algorithm 7.1
    /*Step 5: interaction between processes and standards*/
    Execute Step 3 of Algorithm 7.1
    /*Step 6: select related QRs for each standard */
    Execute Step 1 of Algorithm 7.2
    /*Step 7: interaction between standards and QRs*/
    Execute Step 3 of Algorithm 7.2
    End
```

7.3. The Demonstration of Increment 4

A snapshot of the SoS standard ontology representation including examples of instances from the standards used by KHCC is represented in Figure 7-5. Conceptualisation snapshots of examples from the research case study which demonstrates the interaction between the CS's four quality governance components; standards, policies, processes and quality requirements are partially depicted in Figures 7-7, 7-8, 7-9 and 7-10.



```
▼ owl:Thing
  ▼ StandardModel
    ▼ System_Standard
        SoS_Standard
        Non_CS_Standard
        CS_Standard
      Condition_Standard
    ▼ Content_Standard
      ▶ Requirement_Standard
        Scope_Standard
        TermAndDefinition_Standard
      Goal_Standard
      Guidance_Standard
      MeasurableElement_Standard
    ▼ Resource_Standard
        FinancialResource_Standrad
        Hardware_EquipmentResource_Standard
        Role_Standard
        SoftwareResource_Standard
    ▶ Standard
```

```
▼ owl:topObjectProperty
    SystemsFollowsStan...
    StandardFollowedBySys...
    conditionDetermineStandard_Standard
    standardDescribeContent
    standardDeterminedByCondition_Standard
    standardHasMeasurableElement
    standardNeedResource
    standardProvideGuidance
    standardSpecifyGoal_Standard
```

Show: ☑ this ☑ disjoints ☑ named sub/superclasses

```
Found 18 uses of StandardModel
▼ Condition_Standard
      Condition_Standard SubClassOf StandardModel
▼ Content_Standard
      Content_Standard SubClassOf StandardModel
▼ Goal_Standard
      Goal_Standard SubClassOf StandardModel
▼ Guidance_Standard
      Guidance_Standard SubClassOf StandardModel
▼ MeasurableElement_Standard
      MeasurableElement_Standard SubClassOf StandardMod
▼ Resource_Standard
      Resource_Standard SubClassOf StandardModel
▼ Standard
      Standard SubClassOf StandardModel
▼ StandardModel
      Class: StandardModel
▼ System_Standard
      System_Standard SubClassOf StandardModel
```

Figure 7-5:Snapshot of the Ontological Representation of Standards Model

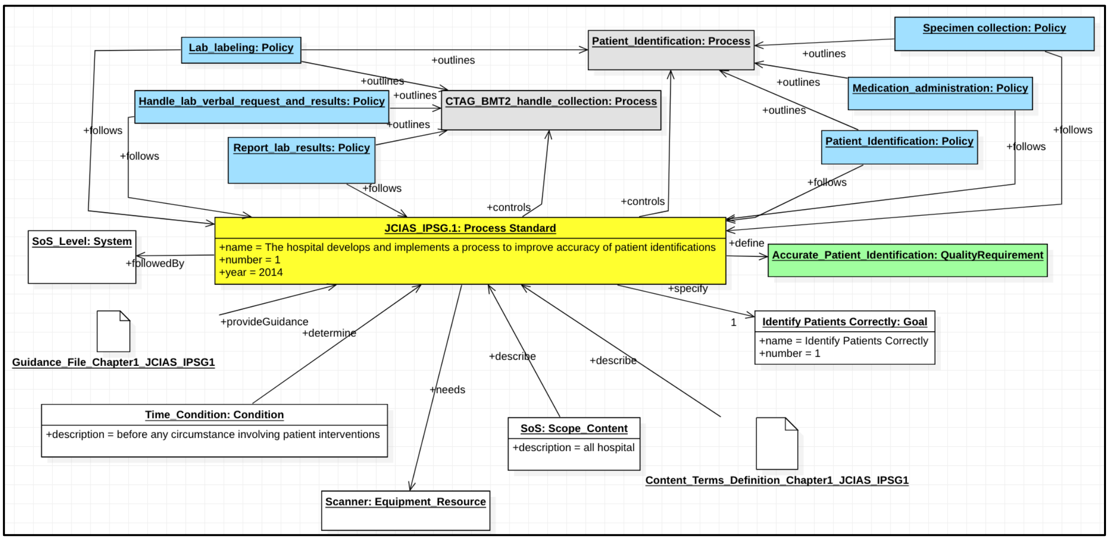

Lab_labeling: Policy
Patient_Identification: Process
Specimen collection: Policy
+outlines
+outlines
+outlines
Handle_lab_verbal_request_and_results: Policy
+outlines
CTAG_BMT2_handle_collection: Process
+outlines
Medication_administration: Policy
+outlines
+follows
+outlines
Report_lab_results: Policy
Patient_Identification: Policy
+follows
+follows
+follows
+follows
+follows
+controls
+controls
JCIAS_IPSG.1: Process Standard
+name = The hospital develops and implements a process to improve accuracy of patient identifications
+number = 1
+year = 2014
SoS_Level: System
+followedBy
+define
Accurate_Patient_Identification: QualityRequirement
+specify
1
Identify Patients Correctly: Goal
+name = Identify Patients Correctly
+number = 1
+provideGuidance
+determine
Guidance_File_Chapter1_JCIAS_IPSG1
+describe
+describe
+needs
Time_Condition: Condition
+description = before any circumstance involving patient interventions
SoS: Scope_Content
+description = all hospital
Content_Terms_Definition_Chapter1_JCIAS_IPSG1
Scanner: Equipment_Resource

Figure 7-6: A Snapshot of a Standard Object Model

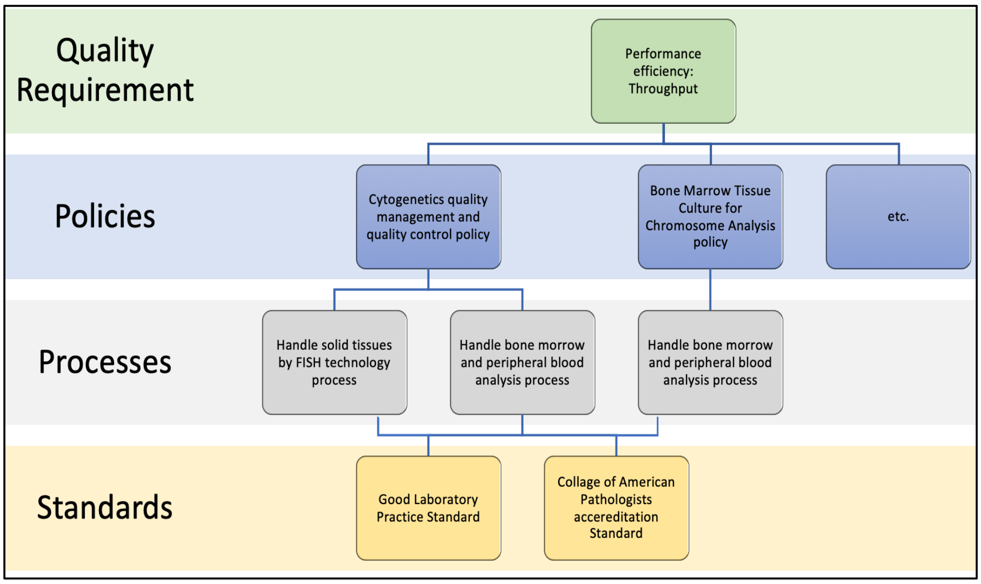

Quality Requirement
Performance efficiency: Throughput
Policies
Cytogenetics quality management and quality control policy
Bone Marrow Tissue Culture for Chromosome Analysis policy
etc.
Processes
Handle solid tissues by FISH technology process
Handle bone morrow and peripheral blood analysis process
Handle bone morrow and peripheral blood analysis process
Standards
Good Laboratory Practice Standard
Collage of American Pathologists acceredititation Standard

Figure 7-7: Partial Example of the Interaction in Increment 4- Algorithm 7.1 (Alternative 1)

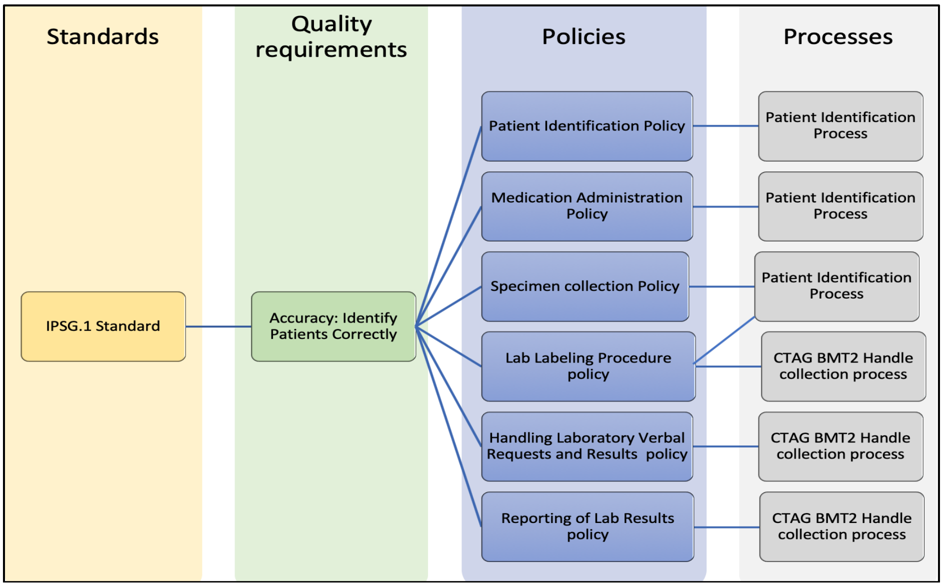
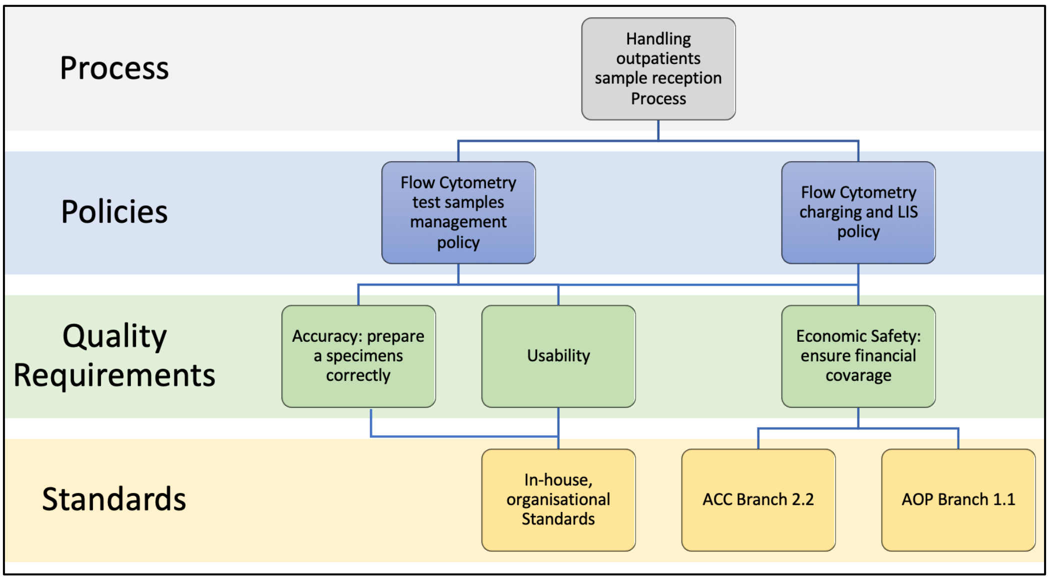
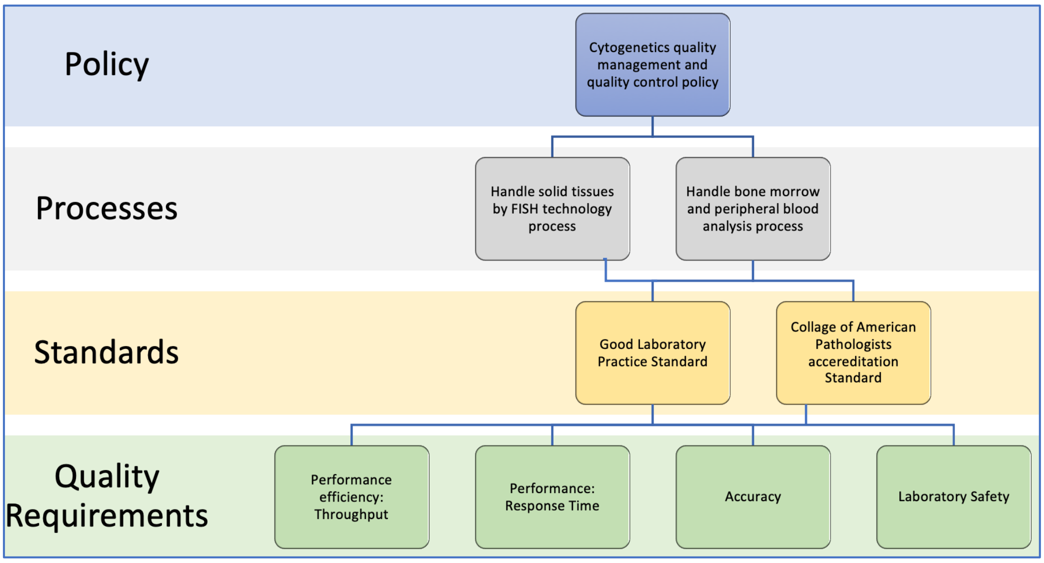

Standards
Quality requirements
Policies
Processes
Patient Identification Policy
Patient Identification Process
Medication Administration Policy
Patient Identification Process
Specimen collection Policy
Patient Identification Process
IPSG.1 Standard
Accuracy: Identify Patients Correctly
Lab Labeling Procedure policy
CTAG BMT2 Handle collection process
Handling Laboratory Verbal Requests and Results policy
CTAG BMT2 Handle collection process
Reporting of Lab Results policy
CTAG BMT2 Handle collection process

Figure 7-8: Partial Example of the Interaction in Increment 4- Algorithm 7.2 (Alternative 2)



Process
Handling outpatients sample reception Process
Policies
Flow Cytometry test samples management policy
Flow Cytometry charging and LIS policy
Quality Requirements
Accuracy: prepare a specimens correctly
Usability
Economic Safety: ensure financial covarage
Standards
In-house, organisational Standards
ACC Branch 2.2
AOP Branch 1.1

Figure 7-9: Partial Example of the Interaction in Increment 4 Algorithm 7.3 (Alternative 3)



Policy
Cytogenetics quality management and quality control policy
Processes
Handle solid tissues by FISH technology process
Handle bone morrow and peripheral blood analysis process
Standards
Good Laboratory Practice Standard
Collage of American Pathologists acceredititation Standard
Quality Requirements
Performance efficiency: Throughput
Performance: Response Time
Accuracy
Laboratory Safety

Figure 7-10: Partial Example of the Interaction in Increment 4 Algorithm 7.4 (Alternative 4)

In order to describe the interaction between standards, policies, quality requirements and processes, an example from the CTAG case study is partially depicted in Figures Figure 7-6 and Figure 7-8 which demonstrates the adherence to the first section of the standard ‘Joint Commission International Accreditation Standards (JCIAS) (“JCI,” 2017)- International Patient Safety Goals 1(IPSG1) : In this context, a hospital develops and implements a process to improve accuracy of patient identifications’, following the process in Algorithm 7-2 and Figure 7-4. Firstly (as depicted in Figure 7-4: Step 1 with orange colour), all related CSs and quality requirements that satisfy the relation ‘Standard *defines* QR’(as depicted in Figures 7-3 and 7-4) will be retrieved from all CSs in the SoS arrangement and listed in a QRs set to be used in Step 2 of Algorithm 7-2. For instant, the

QRs that defined by the standard '*IPSG1: The hospital develops and implements a process to improve accuracy of patient identifications*'. The result of Step 1 is the '*accuracy QR: Identify patients correctly*'. The second step of Algorithm 7-2 is to identify all conflicts in all CSs that are related to the resulted '*Identify patients correctly' QR*. Here, the process in Algorithm 7-1 will be followed (as depicted in Figure 7-4: Step 2 with orange colour). The first step of Algorithm 7-1 is to check the adherence of the '*accuracy QR: Identify patients correctly*' to its related policies and processes in all CSs (as depicted in Figure 7-4: Step 1 with green colour). In order to check such adherence, the same three steps process illustrated earlier in Algorithm 6-1 will be followed (as depicted in Figure 7-4: Steps 1, 2 and 3 with blue colour). The first step of Algorithm 6-1 is to elicit all related policies to the QR *identify patients correctly* and the second step is to elicit all the related processes to each policy resulted from step 1 of Algorithm 6-1. Then, the third step is to identify the conflicts.

The resulting policies that are related to the '*identify patients correctly'* QR are (as depicted in Figure 7-4: Step 1 with blue colour):

1) Po1: *Cord Blood Collection Procedure Policy IPPCTAG-BMT-15R4–* belongs to BMT CS

2) Po2: *Patient/Donor Preparation Procedure IPPCTAG-BMT-35R5–* belongs to BMT CS

3) Po3: *Patient Identification Policy POLPIC-01R6–* belongs to SoS/quality management.

4) Po4: *Medication Administration Policy (POLNUR-11R7) –* belongs to Nursing CS.

5) Po5: *Blood Product Administration & Monitoring (POLNUR-10R7) –* belongs to Nursing CS.

6) Po6: *Specimen collection/handling instructions manual (IPPLAB-PHL01/03R5) –* belongs to Pathology and Laboratory CS.

7) Po7: *Patient Transportation Policy+ Ticket to ride (POLNUR-22R6) –* belongs to Nursing CS.

8) Po8: *Radiation Oncology - Patient Identification (POLRAD -06R5) –* belongs to Radiology Oncology CS.

9) Po9: *Patient movement and Care in Radiology (POLDGR -97R3) –* belongs to Diagnostic Radiology CS.

10) Po10: *Radiology request form Oncology (POLDGR-67R7) –* belongs to Diagnostic Radiology CS.

11) Po11: *DPLM Labeling Procedure IPPLAB-GEN.33R5*– belongs to Pathology and Laboratory CS.

12) Po12: *Handling Laboratory Verbal Requests and Results IPPLAB-GEN.24R4*– belongs to Pathology and Laboratory CS.

13) Po13: *Reporting of Lab Results IPPLAB-GEN.32R5*– belongs to Pathology and Laboratory CS.

So, the outputs are listed in the policy set 'Policy_Set' and the CS set 'CS_Set':

- Policy_Set={01R6, 11R7, 10R7, 03R5, 22R6, 06R5, 97R3, 67R7, 33R5, 24R4, 32R5, 15R4, and 35R5}.
- CS_Set={SoS, Pathology and Laboratory, Diagnostic Radiology, Nursing, BMT}

The resulting processes in the SoS arrangement that has the relation '*specified_By/outlined_By*' that are related to each policy resulted from step 2 of Algorithm 6-1 are retreived (as depicted in Figure 7-4: Step 2 with blue colour). Then, the third step of Algorithm 6-1 is to identify the main conflicts between the policies and processes that are related to the QR '*Identify patients correctly*' (as depicted in Figure 7-4: Step 3 with blue colour). The results of the conflicts that are represented in the Conflicts_Policy_Process_Set are as follows:

1) **Pr$_1$:** *'Patient identification'* process from the SoS level, which is related to the policies:

a) Patient Identification Policy (POLPIC-01R6) – belongs to the SoS/quality management.

- No conflicts.

b) Medication Administration Policy (POLNUR-11R7)- belongs to the Nursing CS.

- Primary and secondary identifications are not existing in the policy.

c) Blood Product transfusion & Monitoring (PONUR-10R7)- belongs to the Nursing CS.

- Only secondary identification is written in the policy which is the patients' medical record number MRN.

d) Patient Transportation Policy (POLNUR-22R6)- belongs to the Nursing CS.

- No conflicts.

e) Radiation Oncology - Patient Identification (POLRAD -06R5) – belongs to Radiology Oncology CS.

- No mention of full name needed as primary identification.

f) Patient movement and Care in Radiology (POLDGR -97R3) – belongs to Diagnostic Radiology CS.

- No conflicts.

g) Radiology request form Oncology (POLDGR-67R7) – belongs to Diagnostic Radiology CS.

- Only secondary identification is written in the policy which is patients' medical record number MRN.

h) CTAG DPLM Labeling Procedure IPPLAB-GEN.33R5 – belongs to Pathology and Laboratory Medicine CS.

- No conflicts.

i) Specimen collection/handling instructions manual (IPPLAB-PHL01/03R5) – belongs to Pathology and Laboratory Medicine CS.

- No conflicts.

2) **Pr$_2$:** *'Handle Sample reception'* process from the MDI CS, which is related to the policies:

a) MDI Samples Reception Instructions IPPLAB-MDI.01R8 – belongs to MDI CS.

- Roles do not match: Main receptionist, Medical Director and Patient are existing in the process but not in the policy.
- Roles do not match: DPLM night shift staff, DPLM phlebotomy and MDI section head are existing in the policies but not in the process.

b) DNA Extraction and Purification IPPLAB-MDI.07-R4 – belongs to MDI CS.

- Only secondary identification is written in the policy which is MDI lab ID number is used as patient identification.

c) RNA extraction IPPLAB-MDI.16-R5 – belongs to MDI CS.

- Only secondary identification is written in the policy which is MDI lab ID number is used as patient identification.

d) Qualitative Real-Time Detection of 29 Somatic Mutations in the EGFR Oncogene IPPLAB-MDI.63-R2 – belongs to MDI CS.

- Only secondary identification is written in the policy which is MDI lab ID number is used as patient identification.
- Roles do not match: Main receptionist, Medical Director and Patient are existing in the process but not in the policy.

3) **Pr$_3$:** *'Handle outpatient sample reception'* process from FC CS, which is related to the policies:

a) Test Samples management IPPCTAG.FC04-R6 policy– belongs to FC CS.

- Roles do not match: Pathologist, Main receptionist and Patient are existing in the process but not in the policy.
- Secondary identification is not written in the policy.

b) Flow Cytometry Charging and LIS IPPLAB.FC26-R1 policy – belongs to FC CS.

- Role does not match: Main receptionist is existing in the process but not in the policy

4) **Pr$_4$:** '*Handle inpatient sample reception'* process from FC CS, which is related to the policies:

a) Test Samples management IPPCTAG.FC04-R6 policy– belongs to FC CS.

- Role does not match: Main receptionist is existing in the process but not in the policy.

b) Flow Cytometry Charging and LIS IPPLAB.FC26-R1 policy – belongs to FC CS.

- Role does not match: Main receptionist is existing in the process but not in the policy.

5) **Pr$_5$:** '*Handle bone morrow and peripheral blood analysis'* process from Cyto CS, which is related to the policies:

a) Cytogenetics Quality Management and Quality Control Plan IPPLAB-CYG.11-R6 policy– belongs to Cyto CS.

- Role does not match: Main receptionist is existing in the process but not in the policy.
- Secondary identification is not written in the policy.

6) **Pr$_6$:** '*Handle solid tissues by FISH technology'* process from Cyto CS, which is related to the policies:

a) Cytogenetics Quality Management and Quality Control Plan IPPLAB-CYG.11-R6 policy– belongs to Cyto CS.

- Role does not match: Main receptionist is existing in the process but not in the policy.
- Identification ways are not existing in the policy.

7) **Pr$_7$:** '*Handle breakage analysis'* process from Cyto CS, which is related to the policies:

a) Cytogenetics Quality Management and Quality Control Plan IPPLAB-CYG.11-R6 policy– belongs to Cyto CS.

- Role does not match: Main receptionist is existing in the process but not in the policy.
- Primary and secondary identifications are not existing in the policy.

The last step of Algorithm 7-1 (as depicted in Figure 7-4: Step 3 with green colour) is to identify any conflicts between the standard '*IPSG1: The hospital develops and implements a process to improve accuracy of patient identifications*' and the processes in the SoS arrangement. The consistency between the main concepts of the standard and the processes need to be checked. In the '*IPSG'* standard, the aim is to identify the conditions that are related to identify patients related to the timing of two main tasks. The timing relates to the '*before*' the tasks 'Treatment' and 'Diagnosis'. The treatment task includes many aspects, e.g. administering medications, blood, or blood products; serving a restricted diet tray; providing radiation therapy; or performing procedures. The diagnosis task includes several aspects, e.g. taking blood and other specimens for clinical testing, or performing a cardiac catheterization or diagnostic radiology procedure. The conditions required with this standard are: firstly, to have two patient's identification ways, e.g. the patient's name, identification number, birth date or a bar-coded wristband. The second condition is that the patient's room number or location cannot be used for identification. In this standard, no particular role is required.

The conflicts between the standard and the processes are as follows:

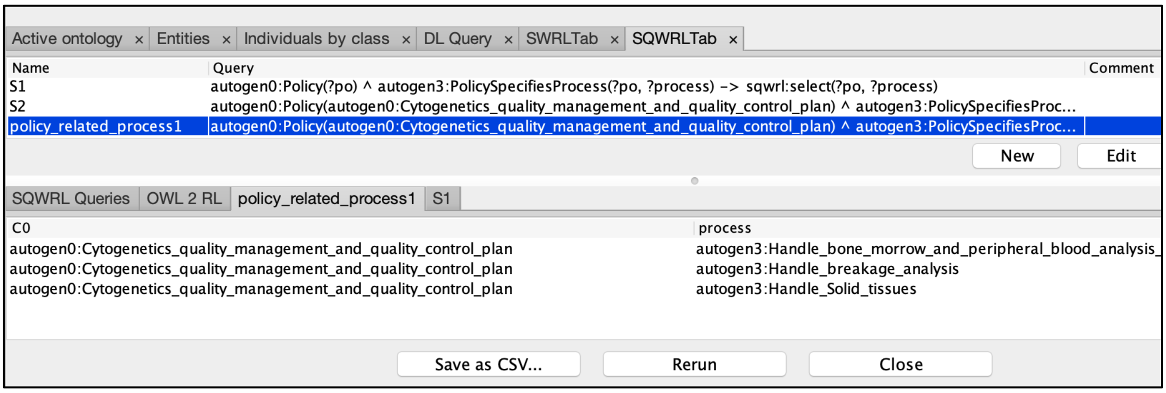
1) **Pr$_1$**: No conflicts.
2) **Pr$_2$**: No conflicts.
3) **Pr$_3$**: Only one way of identification is included (i.e. patient's National number).
4) **Pr$_4$**: Only one way of identification is included (i.e. patient's sample MRN).
5) **Pr$_5$**: Only one way of identification is included (i.e. patient's sample ID).
6) **Pr$_6$**: Only one way of identification is included (i.e. patient's sample ID).
7) **Pr$_7$**: No mention of the way of identification.

Section 7.4 details the evaluation process followed to evaluate both the SoS standard ontology model and Increment 4 (i.e. OntoSoS.QM.Gov framework).

Knowledge retrieval using the Protégé tool for the above steps is represented in Table 7-2 using SQWRL statements. For instance, Figure 7-11 shows the output of the SQWRL statement that retrieve all processes that are related to a certain policy e.g. Cytogenetics_quality_management_and_quality_control_plan policy is '*Policy (Cytogenetics_quality_management_and_quality_control_plan) ^ PolicySpecifiesProcess (Cytogenetics_quality_management_and_quality_control_plan,?pr) -> sqwrl:select (Cytogenetics_quality_management_and_quality_control_plan,?pr)*' which results listing the three processes: '*Handle bone morrow and peripheral blood analysis'*, '*Handle solid tissues'* and '*Handle breakage analysis'*. These three processes are connected with policy 'Cytogenetics_quality_management_and_quality_control_plan' via the relation: *'Policy specifies Process* '.

Table 7-2: Examples of The Main SQWRL-based Knowledge Retrieval Statements to Enable Retrieving Knowledge

Statement	SQWRL Statement
To retrieve all related CSs and quality requirements that satisfy the relation '*definedBy*' the standard *IPSG1*	Standard (IPSG1) ^ StandardFollowedBySystem (IPSG1,?system) ^ StandardDefinesQR (IPSG1,?qualityRequirement) -> sqwrl:select (IPSG1,?system, ? qualityRequirement)
To retrieve all related policies to the '*accuracy QR: Identify patients correctly*'	QualityRequirement(Identify_patients_correctly) ^ QR_Asserts_Policy (Identify_patients_correctly,?po) -> sqwrl:select (Identify_patients_correctly,?po)
To retrieve all processes that are related to policies via the relation: Policy '*specifies'* Process	Policy(?po) ^ PolicySpecifiesProcess(?po, ?pr) -> sqwrl:select(?po, ?pr)
To retrieve all processes that are related to a certain policy (i.e. Cytogenetics_quality_management_and_quality_control_plan policy) via the relation: Policy '*specifies'* Process	Policy (Cytogenetics_quality_management_and_quality_control_plan) ^ PolicySpecifiesProcess (Cytogenetics_quality_management_and_quality_control_plan,?pr) -> sqwrl:select (Cytogenetics_quality_management_and_quality_control_plan,?pr)
To retrieve the conflicts between standards and processes that are related to Roles (Satisfied if the first argument is the same as the second argument (upper/lower case ignored)	Standard(?st) ^ needsResource(?st, ?role) ^ roleHasDescription(?st, ?descriptionSt) ^ Process(?pr) ^ hasRole(?pr, ?role)^ roleHasDescription(?pr, ?descriptionPr)^ swrlb:stringEqualIgnoreCase(?descriptionSt, ? descriptionPr) -> sqwrl:select (?role,?po) ^ sqwrl:select (?role,?st)
To retrieve the conflicts between the conditions needed for standards and events in processes (Satisfied if the first argument is the same as the second argument (upper/lower case ignored)	Standard(?st) ^ determinedBy(?st, ?condition) ^ conditionHasDescription(?st, ?descriptionSt) ^ Process(?pr) ^ hasEvent(?pr, ?event)^ eventHasDescription(?pr, ?descriptionPr)^ swrlb:stringEqualIgnoreCase(?descriptionSt, ? descriptionPr) -> sqwrl:select (?condition,?po) ^ sqwrl:select (?condition,?st)
To retrieve the conflicts between the valid time duration in quality requirements and the time constraint in a policy (Satisfied if the first argument and the second argument are not the same)	QualityRequirement(Identify_patients_correctly) ^ QR_Start_Period (Identify_patients_correctly, ?startPeriod) ^ QR_End_Period (Identify_patients_correctly, ?endPeriod) ^ swrlb: subtract(?period, ?end_Period, ?start_period) ^ policy_has_Constraint(?po, ?timeConstraint)^ swrlb: notEqual((?period, ? timeConstraint) -> sqwrl:select (Identify_patients_correctly,?qr)^ sqwrl:select (?po, ?timeConstraint)^ sqwrl:select (Identify_patients_correctly,?hasMandatory, ?hasTendency, ?hasWeight)



Name	Query	Comment
S1	autogen0:Policy(?po) ^ autogen3:PolicySpecifiesProcess(?po, ?process) -> sqwrl:select(?po, ?process)	
S2	autogen0:Policy(autogen0:Cytogenetics_quality_management_and_quality_control_plan) ^ autogen3:PolicySpecifiesProc...	
policy_related_process1	autogen0:Policy(autogen0:Cytogenetics_quality_management_and_quality_control_plan) ^ autogen3:PolicySpecifiesProc...	

C0	process
autogen0:Cytogenetics_quality_management_and_quality_control_plan	autogen3:Handle_bone_morrow_and_peripheral_blood_analysis_
autogen0:Cytogenetics_quality_management_and_quality_control_plan	autogen3:Handle_breakage_analysis
autogen0:Cytogenetics_quality_management_and_quality_control_plan	autogen3:Handle_Solid_tissues

Figure 7-11: A Snapshot of executing a SQWRL statement

7.4. The Evaluation of Increment 4: Step 1

After having developed the first version of the OntoSoS.Stand model, an evaluation process was conducted to evaluate the SoS standards ontology model and then, to evaluate the interaction process between the SoS policies, processes, QRs and standards ontology models. The feedback gained from the domain experts was applied to Increment 4. Thus, at the end of this phase, Increment 4 was validated and the output was the second version of Increment 4, which represents the OntoSoS.QM.Gov model.

The first evaluation step of the general evaluation method detailed in Section 3.8.4 will not conducted for the standard model as it was developed to include only the common key terms of existing standards models. The second evaluation step was to evaluate the ontology structure of the first version of the OntoSoS.Stand model and the interaction between the four SoS quality governance models (i.e. SoS standards, processes, policies and QRs models). This will report the satisfaction of domain experts regarding the correctness of the structure of the OntoSoS.Stand model. Tables Table 7-3 and Table 7-4 show the semi-structured and chick list conducted with domain experts from KHCC. The interviewees were selected based on the ability to provide the necessary knowledge and had experience in the area of standards, processes, modelling, policies, conflicts and quality requirements.

Table 7-3: Checklist to Validate the Standards Ontology Model Using Walkthrough Approach

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
1	Validate the concept “standard” and its Data Properties (DP) and Object Properties (OP).	Concept: Standard	*		-
		DP: standard Name	*		-
		DP: standard Description	*		-
		DP: satisfy Goal		*	Could be related to more than one goal
		DP: standard Version	*		-
		DP: standard Number	*		-
		DP: standard Year	*		-
		OP: define QR	*		-
		OP: integrate with Policy	*		-
		OP: control Process	*		-
		OP: followed_By System	*		-
2	Validate the concept “Resource” and its properties	Concept: Resource	*		-
		DP: name	*		-
		DP: description	*		-
		DP: related CS	*		-
3	Validate the concept “Tool” and its properties			*	Tool neds to be combined with the needed resources for any standard.
4	Validate the concept “Guidance” and its properties	Concept: Guidance	*		-
		DP: Name	*		-
		DP: description	*		-
5	Validate the concept “Goal” and its properties	Concept: Goal	*		-
		DP: name	*		-
		DP: description	*		-

	Goal	Concepts and properties	Result		Remarks from Domain Experts (if not satisfied)
			Satisfied	unsatisfied	
		OP: standard satisfy a goal		*	Each standard may support more than one goal.
6	Validate the concept “Condition” and its properties	Concept: Condition	*		-
		DP: description	*		-
7	Validate the concept “Content” and its properties	Concept: Content	*		-
		DP: description	*		-
8	Validate the concept “System” and its properties	Concept: System	*		-
		OP: is_A	*		-
9	Validate the concept “SoS” and its properties	Concept: SoS	*		-
10	Validate the concept “CS” and its properties	Concept: CS	*		-
11	Validate the concept “Non_CS” and its properties	Concept: Non_CS	*		-

Table 7-4: The Outcomes of the Main Questions of the Interviews Conducted with Domain Experts to Validate SoS Standards Model and the Relations with Other Models in Increment 4

	Questions	Notes by Domain Experts
1	Do you agree that the above concepts and associated properties **completely** represent the KHCC’s standards related concepts?	√
2	Are there any missing concepts or properties?	The concepts Measurable_Element and Financial resources were suggested.
3	Are there any other properties that could enhance the model?	The relation ‘each Standard has one or Measurable Elements’
4	Are there any extra concepts or properties -that would be better removed from the model?	The relation between the concept Standard and the concept Tool can be removed as it considered one of the

	Questions	Notes by Domain Experts
		Resources Concepts (i.e. Software Resources).
5	Do you agree that the above concepts and associated properties **correctly** represent the KHCC’s standards related concepts?	√
6	Are the data types that used to describe the Data Properties correct?	√
7	Are the relations (Object Properties) that are used to connect the concepts of the standards model correct?	√
8	Are the domain and the range for the relations between the concepts correct?	√
9	Are the cardinality constraints of the relations correct?	The relation between the Standard concept and the Goal concept was suggested to be ‘each Standard specifies one or more Goals’ (i.e. not just to only one Goal).
10	Do you agree that the above concepts and associated properties **consistent** with each other and do not lead to conflict (free of contradiction with other components in the model)?	√
11	Do you agree with the relation between the policy model and the standards model?	√
12	Do you agree with the relation between the processes model and the standards model?	√
13	Do you agree with the relation between the quality requirements model and the standards model?	√
14	Do you agree with the process (Algorithms) of interaction provided?	√
15	Do you suggest any further concepts/ changes to be applied on increment 4?	The changes that have been suggested above.

As observed in Tables Table 7-3 and Table 7-4, the main concepts of the OntoSoS.Stand model were validated by domain experts with feedback used to revisit the design of the model. The amendments have been applied as discussed in Section 7.5.

7.5. Revisiting Increment 4

After the validation of the OntoSoS.Stand model and the interaction between standards, processes, policies and QRs models, only few amendments were suggested by domain expert and were implemented to produce the second version of increment 4 which is coloured in orange as depicted in Figure 7-12. For instance, the concept ‘Measurable Element’ was suggested to indicate a value determined by a standard, e.g. the ratio between space and number of beds in a hospital room, speed of retrieve information, etc. Other suggested amendment was to link the concept ‘Tool’ with the concept ‘Resource’. Also, the concept ‘Financial Recourse’ was created as part of the resources needed to implement a standard. Another amendment was to modify the restriction on the relation between the concepts Goal and Standard to be ‘each Standard may have one or more Goals’. After enhancing the standards model, the relations between it and other models were reviewed and the interaction between them are reported as satisfied.

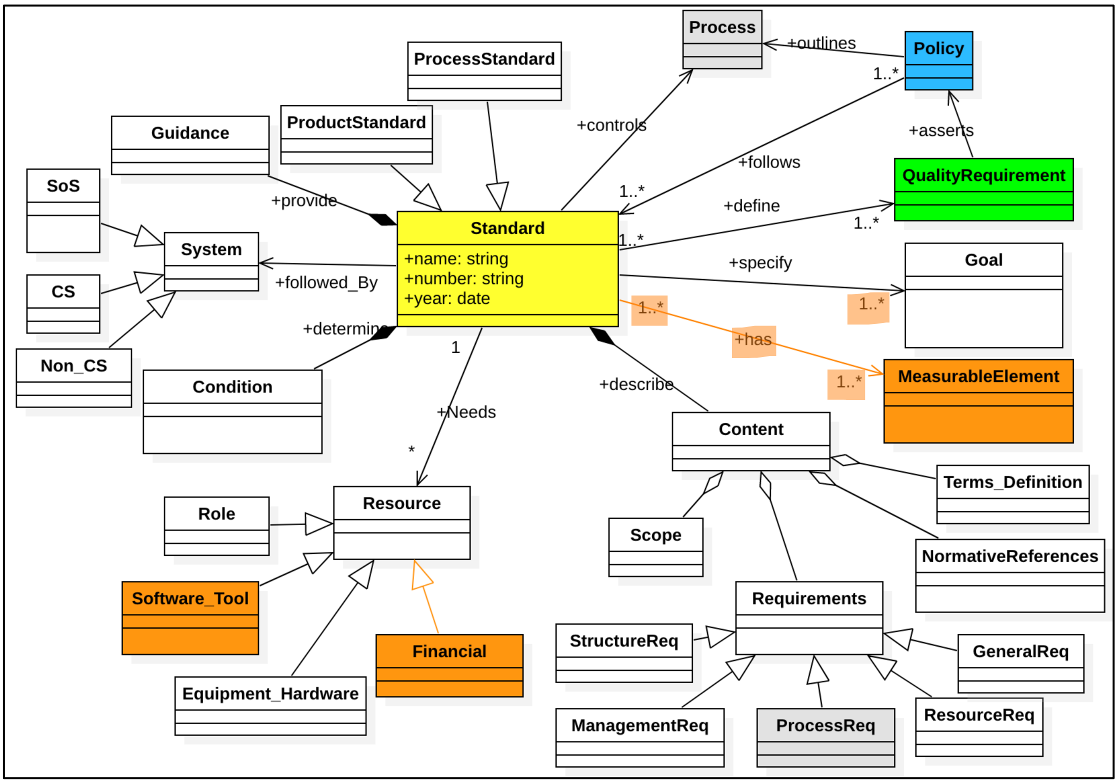


Figure 7-12: The Conceptual Representation of Version 2 of OntoSoS.Stand

7.6. The Evaluation of the OntoSoS.QM.Gov Framework: Step 2

After completing the incremental development, demonstration and evaluation of the SoS component models of the OntoSoS.QM.Gov framework, this evaluation step focused on assessing the effectiveness of the overall SoS quality governance process with regard to checking the adherence to the SoS quality governance. As identified in Section 2.13, one of the main gaps identified was the need to create a quality governance process that involves interaction between processes, policies and standards to inform the adherence to quality in the SoS context. In order to check such adherence, semi-structured interviews were conducted with domain experts from KHCC to inform the satisfaction of the results of the whole interaction process illustrated in Section 7.2 and demonstrated in Section 7.3. Also, the effectiveness of the process identified in the OntoSoS.QM.Gov is assessed against the current methods followed with the quality department at the SoS level at KHCC.

Interviewing domain experts at the KHCC reported that the approach applied currently at KHCC to check the adherence to the quality governance does not follow a structured process to check the adherence to quality in SoS context.

As a proof of concept, the KHCC's documents that consist of the policies and processes that are related to quality standards, semi-structured interviews and checklist walkthrough have been used to support the comparison as a step to assess the effectiveness of the OntoSoS.QM.Gov. The comparison between the current quality governance approach followed at the research case study and the OntoSoS.QM.Gov framework is shown in Table 7-5.

Table 7-5: The Comparison Conducted to Assess the Effectiveness of the OntoSoS.QM.Gov Framework against the Case study Process Followed

	Comparison Aspect	**The CTAG Approach**	**The OntoSoS.QM.Gov Framework**	**Remarks from Domain Experts** (if not satisfied)
1	Conceptualisation of the main Quality governance- related aspects	- Only text-based representation without clear separation between the main concepts for each component model. - Lack of model-based conceptualisation. Only partial views of some processes are existing.	-All main aspects of the four constituent models are explicitly conceptualised. - Models for the four constituent models are existing.	✓
2	Formal specification of the quality governance models-related aspect	- Lacks formal representation	-Formal ontology-based specification.	✓
3	Knowledge representation of the policies at the CTAG case study	- Lacks knowledge representation, only paper documents, pdf and MS Word files.	-Ontology-based representation. -Model-based representation.	✓
4	Knowledge representation of the standards at the CTAG case study	- Lacks knowledge representation, only hard copies and pdf copies of the standards chapters.	-Ontology-based representation. -Model-based representation.	✓

	Comparison Aspect	The CTAG Approach	The OntoSoS.QM.Gov Framework	**Remarks from Domain Experts** (if not satisfied)
5	Knowledge representation of the processes at the CTAG case study	- Lacks knowledge representation, only partial views of some processes are existing as flowcharts.	-Ontology-based representation. -Model-based representation.	✓
6	Knowledge representation of the QRs at the CTAG case study	- Lacks knowledge representation, only paper documents, pdf and MS Word files.	-Ontology-based representation. -Model-based representation.	✓
7	The representation of the relations between the four quality governance models at the CTAG case study	- Lacks clear interactions between the quality governance models. Only not-documented implicit and representation by key stakeholders by the CTAG case study.	-Ontology-based representation. Many relations can be inferred by other relations using protégé tool. -Model-based representation.	✓
8	The representation of the relations between the global SoS level, the constituent systems and non-constituent systems at the case study	- Lacks clear interactions between the SoS level, constituent systems and non-constituent. Relations cannot be easily inferred by non-knowledgeable stakeholders. The relations are done manually.	-All models are conceptualised by taken into consideration the relations between the SoS level, constituent systems and non-constituent.	✓
9	The representation of the quality governance semantic heterogeneities	- Lack of formal identification of the synonyms that are related to the concepts used in processes, policies, different standards and QRs definitions, e.g. QRs measurements' values, roles titles, terminologies used in different standards in different constituent systems, etc.	- All synonyms in all different constituent systems can be identified and linked together using ontology tools.	✓
10	The representation of the adherence to SoS quality governance- related models	- Only paper- based and computer-based files are documented for the purpose of showing some level of adherence. This done by individual efforts of some key stakeholders at the research case study. - Depending on observations and manual process check, they report by filling 'event report' forms.	- Several algorithms that identify the interaction points and related aspects between the four models of the quality governance framework can be implemented to identify any conflicts that may affect the adherence to the SoS quality governance.	✓
11	Knowledge retrieval capabilities (e.g. retrieve related models objects to other models, constituent systems that are related to a specific model object, etc.)	- Lack of clear process to retrieve knowledge. Manual comparisons conducted to retrieve information from paper- documents. - Difficulties to retrieve all constituent systems engaged in a conflict.	- Knowledge retrieval is illustrated using clear processes which are represented in algorithms and conducted using SQWRL rules.	✓
12	Time consuming to conduct retrieval	- Time consuming manual process that may need several days to be finished.	- More efficient with regard to time consuming to show results. However, Reasoner tool used to check the consistency is a time consuming with	✓

	Comparison Aspect	The CTAG Approach	The OntoSoS.QM.Gov Framework	Remarks from Domain Experts (if not satisfied)
			taken into consideration that it needs to be conducted only few times while developing the knowledge base and when extend the ontology model.	
13	Accuracy of knowledge retrieval	- Sometimes accuracy depends on the experience of the person who conducts the retrieval process from paper-documents. - Conflicts may not be discovered due to the low experience of knowing all synonyms of the concepts in all systems.	- Mostly were accurate but with few limitations: it would be better if more concepts were included, e.g. the concepts that are related to externally related non-functional qualities such as adherence to legal issues and constraints that are related to contract and service level agreements between constituent systems themselves and non- constituent systems. -Also, a certain level of experience is needed to create SQWRL retrieval statements. However, no need for high level experience with regards to processes, policies, standards and knowledge of all constituent systems and SoS as the process of retrieval will depend on the accuracy of saving information into the Protégé tool.	✓

The evaluation conducted in Step 1 and the comparison presented in Table 7-5 show that the OntoSoS.QM.Gov framework provides significant contributions to the domain of SoS quality governance. These contributions are (i) the generalised processes that were presented in several algorithms and followed to check the adherence of the SoS quality governance by checking the adherence of each SoS governance component model (i.e. processes, QRs, policies and standards); (ii) the utilising of knowledge retrieval capabilities of SQWRL rules; (iii) the conceptualisation of the SoS models that showed clear views regarding the relations between the SoS quality governance component models and constituent systems in the SoS arrangement; (iv) the linking between the different roles, constituent systems, constraints and conditions related to each identified conflict; (v) the formal representation of the different component models and their relations to each other; (vi) reducing the semantic heterogeneities by providing the synonyms and applying semantic constraints features provided by ontology, and (vii) providing a semi-automated approach to detect conflicts in comparison to the manual checking with paper-based documents at the case study.

7.7. Summary and Discussion

After reviewing the literature in relation to standards models, they have been taken into consideration to elicit the common concepts between them to design the OntoSoS.Stand model in the SoS context. After developing the first version of the OntoSoS.Stand, it was linked with the previously developed increments (i.e. QR, policy and processes models) to formulate the fourth and the final increment of the SoS quality governance framework (i.e. OntoSoS.QM.Gov). This increment was demonstrated using the CTAG case study and validated with domain experts to check the adherence to governance. Also, the interactions between standards, QRs, processes and standards ontology models were investigated and validated. The feedback obtained was applied to produce a newer version of Increment 4. However, only minor amendments were required to be reflected in the design.

After revisiting the model, another validation step was conducted to investigate the effectiveness of the OntoSoS.QM.Gov framework. In this step, mainly a comparative validation was conducted against the quality governance process followed by the research case study. The result of this comparison reported the effectiveness of this research with regards to the novelty of the OntoSoS.QM.Gov framework and the benefits of the interaction between the linked SoS quality governance component models (i.e. processes, QRs, standards and policies models) as the combination between OWL and SQWRL offers an expressive platform as a step towards constructing a generic automated SoS quality governance framework for future research direction discussed in Chapter 8.

By the end of iteration 4 of the DSRM process, Table 7-6 provides an overview of the status towards answering the research questions. The tick symbol (✓) shows that the research questions/sub-questions have been answered. As reflected from Table 7-6, research questions 2,3 and 4 were only partially answered in the previous DSRM iterations due to not completing the whole SoS quality governance framework in particular, and not completing the interaction process between the four component models of the SoS quality governance. The shortcomings resulted of not involving all models and the need for other models to be developed and linked with each previously developed increment are concluded at the end of each DSRM iteration which were represented in the chapters 4 to 7.

RQ1 has been answered previously in Chapter2. This was after identifying the main challenges and components of the SoS quality governance framework. Then, the four SoS quality governance components were developed and linked using ontology constructed with the process of Noy and McGuiness (2001). All identified aspects of each model with adaptations including the relations between them were brainstormed from previously developed ontology models, and therefore supports answering RQ2.

Table 7-6:Status towards Answering the Research Questions

RQ	Main RQ and RQs-Concerns (sub-questions)	Status	Notes	Chapter
RQ1	*What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*	✓		2
1.1	Can research gap analysis, by surveying the literature, identify these challenges ?	✓		2
RQ2	*How to represent and model the quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*	✓		2, 4,5,6 & 7
2.1	What are the components of the quality governance framework, and what are the initial specifications of each component ?	✓		2
2.2	Is ontology suitable to define all related aspects to quality governance ?	✓		7
2.3	What are the elements/concepts of the ontology? Are there any ontologies that we can reuse ?	✓	See Tables Table 4-1, Table 4-2, Table 5-1, Table 6-1, and Table 7-1.	4,5,6 & 7
RQ3	*How will the semantically-enriched models of policies, processes, standards and quality requirements interact in the systems of systems context to identify and resolve semantic heterogeneities?*	✓		5,6 & 7
3.1	Can we develop a process to detail the interaction between *policies, processes, standards and quality requirements* *to identifying and resolve semantic heterogeneities* ?	✓	See Algorithms Algorithm 5-1, *Algorithm 5-2*, Algorithm 6-1, *Algorithm 6-2*, Algorithm 7-1, Algorithm 7-2, Algorithm 7-3 and 7-4.	5,6 & 7
3.2	Are there any limitations of using an ontology-based approach that restrict the interactions between *policies, processes, standards and quality requirements* ?	✓		5,6 & 7
RQ4	*How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*	✓		4,5,6 & 7
4.1	How the OntoSoS.QM.Gov ontology will be assessed ?	✓	General evaluation process presented in Chapter 3	7
4.2	Can we validate each ontology model (i.e. each component of the quality governance framework) and then to validate the whole quality governance ontology framework ?	✓	See Sections 4.4, 5.4, 6.4, 7.4 and 7.6.	4,5,6 & 7

In Increment 4, the SoS standards model was constructed and linked with the previous developed models. In this step, all concepts that have the same semantics to other synonym concepts in other SoS quality governance models were investigated as a step towards identifying any semantic heterogeneities that may occur after comparing the values of the synonym concepts, e.g. ‘documentation concept’ in policy models can be referred to as a ‘data object’ in processes models. Another example can be related to the definition of the concept ‘child’ as it can be identified in a standard in a constituent system as patients under 16 years while it can be identified in a policy in another constituent system as patients under 18 years. Such examples have been resolved by identifying semantic rules in the related ontologies. Therefore, it can be concluded that using semantically- developed models has contributed to resolving such conflicts and identifying other conflicts. However, still some other conflicts can be identified with the need for human involvement to be resolved suggests the RQ3 has been answered substantially with still human intervention in the case study.

Furthermore, the process followed to validate the consistency, correctness and completeness of the structure of the SoS standards model building on previously validating processes, QRs and policy governance models followed by validating the linking relations between the SoS standards model with the other three models, and then validating the whole SoS.QM.Gov framework as illustrated in Section 7.6 support answering RQ4.

In conclusion, the development, demonstration and evaluation of the research framework (i.e. SoS.QM.Gov framework) is completed by the end of this chapter building on the incremental answering of research questions 2, 3, and 4 in chapters 4- 6.

8. Chapter Eight: Conclusion

8.1. Introduction

This research investigated the feasibility to semantically enrich quality requirements, processes, standards and policies of the heterogenous constituent systems in the SoS arrangement and to formally represent the interactions between them in order to report an effective SoS quality governance process by identifying and resolving semantic conflicts emerging from the interactions between the component models in the heterogenous constituent systems in the SoS arrangement.

This research adopted the DSRM methodology. The DSRM consists of six phases: (1) Problem Identification and Motivation, (2) Suggestions, (3) Design and Development, (4) Demonstration, (5) Evaluation and (6) Communication. The phases 3-5 have an iterative nature. In this research, the four main models of quality governance framework (i.e. OntoSoS.QR, OntoSoS.Policy, OntoSoS.Process and OntoSoS.Stand) are developed incrementally with each increment informing the next increment. When the fourth increment was completed, the interaction between the four SoS quality governance models of this research framework was achieved reflecting an incremental design of the SoS quality governance framework (OntoSoS.QM.Gov), evaluated using a representative case study in cancer care.

This chapter is organised as follows. Section 8.2 addresses the answering of the research hypothesis and associated research questions. Section 8.3 summarises the main research contributions to knowledge. Section 8.4 highlights future research directions.

8.2. Answering the Research Hypothesis

Each DSRM phase and its related research activities contributed to the fulfilment of the research objectives in order to evaluate the current research and finally accept or reject the research hypothesis. Figure 8-1 illustrates the thesis roadmap to answer the research questions. RQ 1 is fully satisfied in Chapter 2 while RQ2 is partially answered in Chapter 2. Due to the nature of the incremental development, demonstration and evaluation of the quality governance framework, it was not possible to fully answer all other research questions in each DSRM increment; but RQs have been incrementally answered in each DSRM increment. By the fourth DSRM increment, RQ2 to RQ5 have been completely answered.

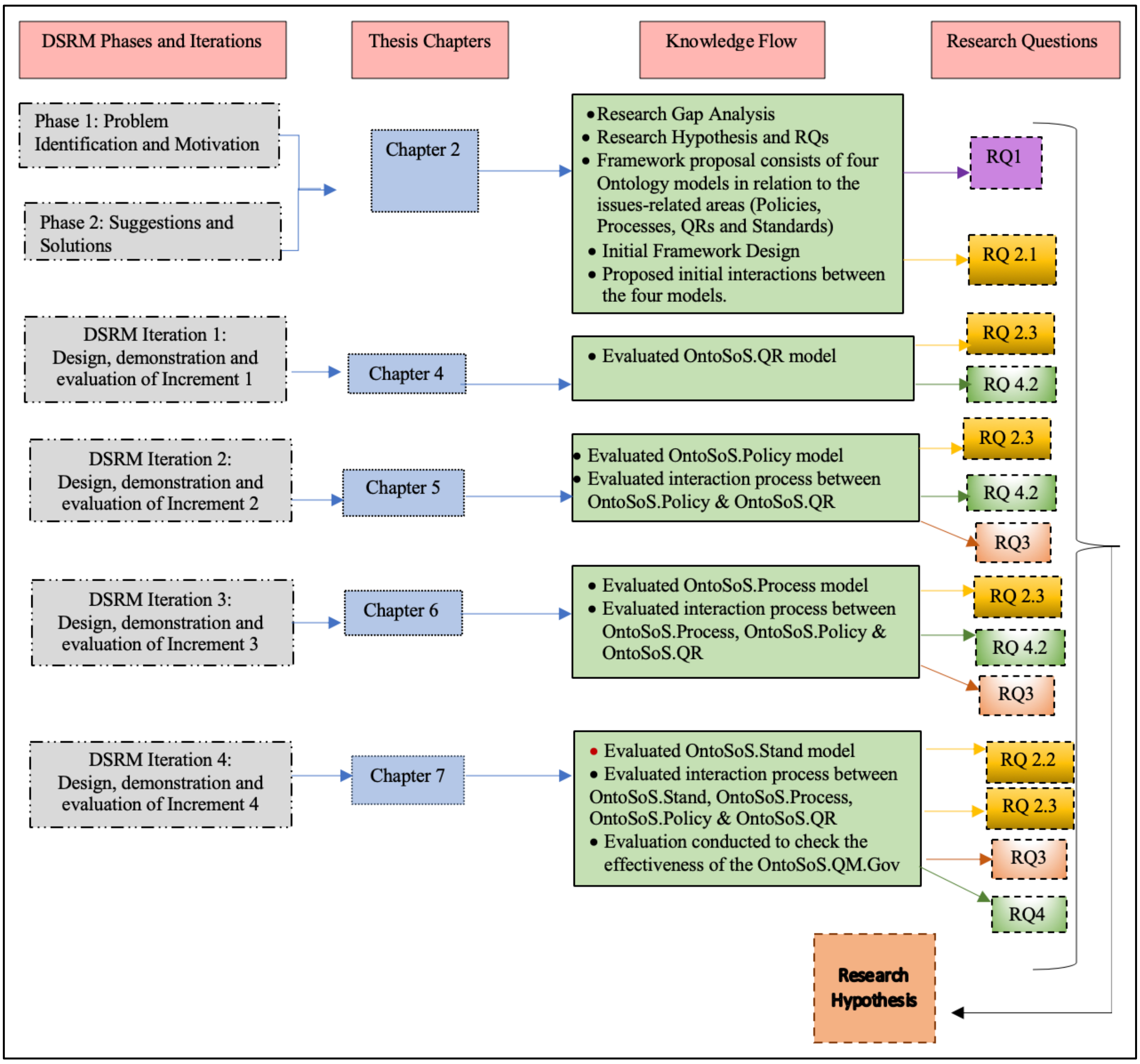

DSRM Phases and Iterations
Thesis Chapters
Knowledge Flow
Research Questions
Phase 1: Problem Identification and Motivation
Phase 2: Suggestions and Solutions
Chapter 2
• Research Gap Analysis
• Research Hypothesis and RQs
• Framework proposal consists of four Ontology models in relation to the issues-related areas (Policies, Processes, QRs and Standards)
• Initial Framework Design
• Proposed initial interactions between the four models.
RQ1
RQ 2.1
DSRM Iteration 1: Design, demonstration and evaluation of Increment 1
Chapter 4
• Evaluated OntoSoS.QR model
RQ 2.3
RQ 4.2
DSRM Iteration 2: Design, demonstration and evaluation of Increment 2
Chapter 5
• Evaluated OntoSoS.Policy model
• Evaluated interaction process between OntoSoS.Policy & OntoSoS.QR
RQ 2.3
RQ 4.2
RQ3
DSRM Iteration 3: Design, demonstration and evaluation of Increment 3
Chapter 6
• Evaluated OntoSoS.Process model
• Evaluated interaction process between OntoSoS.Process, OntoSoS.Policy & OntoSoS.QR
RQ 2.3
RQ 4.2
RQ3
DSRM Iteration 4: Design, demonstration and evaluation of Increment 4
Chapter 7
• Evaluated OntoSoS.Stand model
• Evaluated interaction process between OntoSoS.Stand, OntoSoS.Process, OntoSoS.Policy & OntoSoS.QR
• Evaluation conducted to check the effectiveness of the OntoSoS.QM.Gov
RQ 2.2
RQ 2.3
RQ3
RQ4
Research Hypothesis

Figure 8-1: Answering of the Research Questions and Research Hypothesis

The following reports are the outcomes of answering the four research questions in order to accept or reject the research hypothesis:

1- RQ1: *What are the main quality governance issues that have not been addressed in the literature in relation to the interaction between policies, processes, standards and quality requirements models in a system of systems context ?*

Based on the literature review conducted in the *Problem Identification and Motivation* phase of the adopted DSRM process (Chapter 2), the SoS quality governance domain revealed deficiencies in relation to the adherence to SoS quality governance. For instance, the conflicts resulting from the interoperability and heterogeneity between the heterogenous constituent systems in the SoS arrangement, the lack of utilising the SoS domain governance processes that facilitate achieving higher level of quality, the need to create a general SoS quality process, the shortage of applying SoS quality governance within the cancer care domain, the need of a SoS quality governance framework that contains interaction process between the governance related areas (i.e. policies, QRs, processes and standards). All the aforementioned gaps have not been fully answered in the SoS context. They can be categorised into four areas (i) quality requirements, (ii) standards (iii) policies (iv) processes and (v) heterogeneity and interoperability between the constituent systems in relations to the quality requirements, policies, standards and business processes.

In order to respond to these quality governance related gaps, a SoS quality governance framework has been proposed. It consists of four Ontology models in relation to the areas related to the SoS quality governance (i.e. policies, quality requirements, standards and business processes). Several interactions between these four models were identified between these models. These interaction points can be related to roles, their activities and the conditions of these activities. These interactions can be occurred within the four models in the same system or with other systems.

The involvement of the concepts that are related to domain processes (e.g. roles, activities, etc.) and the interaction between them in the four models, the recommendations in the literature to use an incremental and iterative development method led the proposal of the research design methodology adopted in this research.

2- RQ2: *How to represent and model the SoS quality governance issues in relation to policies, processes, standards and quality requirements using a semantically enriched approach ?*

In order to semantically represent the SoS quality governance related areas, literature review was conducted to inform these areas which are the policies, quality requirements, processes and standards, using ontologies. However, because there are many ontology construction methods in the literature, a comparison between several methods was conducted in Chapter 2 and concluded with the method of Noy and McGuiness (2001) and its ability to build and reuse existing ontologies and brainstorming extra concepts that are related to the SoS and constituent systems.

Each SoS model of the four component models of the OntoSoS.QM.Gov framework was developed, demonstrated and evaluated in the chapters from 4 to 7. In this research, the aim of identifying and solving heterogeneities had an impact on the design of the four models. The interaction points were identified in RQ1 and been taken into consideration in the design of each model, especially, in the case of reusing existing ontologies.

Firstly, to develop the SoS quality requirements ontology model (i.e. OntoSoS.QR), some key concepts and design requirements was identified after reviewing the literature (as discussed in Section 2.8), e.g. the concepts of metrics, dependency relations with other quality requirements, the mandatory and priority characteristics of the quality requirements, the concepts of the SoS and constituent systems, etc.

Secondly, to develop the SoS policy requirements ontology model, several key concepts that need to be governed were adopted from the literature, e.g. the roles and their main tasks, the conditions that need to be satisfied, the quality goals of the policies, the concepts of the SoS and constituent systems, etc.

Thirdly, to develop the SoS processes ontology model, the main concepts identified were adopted from the BPMN notations due to its expressiveness, which was proved in the literature. Also, the processes of the selected case study were developed and evaluated in other research (Odeh et al., 2018). However, the concepts of the SoS and constituent systems were added.

Fourthly, to develop the SoS standards ontology model, the main components of the standards were identified by eliciting the common concepts between standards without specific details regarding one particular standard. This is because the exist of several standards followed by each constituent system in the SoS arrangement. For example, the concepts of conditions, roles, resources, the concepts of the SoS and constituent systems, etc.

After finishing all components in Chapter 7, it was revealed that using semantics data models using diagrams and using ontologies can capture most of the concepts that are related to each model including the interaction relations between the models. However, some limitations were reported at the end of Chapter 7 which led to suggest some future research.

3- RQ3: *How will the semantically enriched models of policies, processes, standards and quality requirements interact in the system of systems context to identify and resolve semantic heterogeneities?*

In this research, a novel interaction process has been developed and evaluated incrementally. After developing each component model, the interaction process was examined to check its sufficiency for checking the adherence to SoS quality governance. The result of the sufficiency test informed the next model to be developed. The result of the sufficiency was attributed to the nature of the concepts in the models. For example, after the SoS quality requirement model has been developed, it was inferred that the quality requirements model was not sufficient for checking the adherence to quality governance. This was due to the lack of the concepts related to the constraints affecting the quality values and the services related to the quality requirements. However, these concepts were part of the policies model, which was then decided to be added in the second DSRM increment. Then, it was inferred that the interaction between policies and quality requirements was not sufficient for checking the adherence to quality governance, which was attributed due to the lack of the concepts related to the actions performed by the roles and the documentations or forms needed by these roles, which were part of the business processes model, which was then added to the third DSRM increment. After this, it was inferred that the interaction between the constituent models of increment 3 (i.e. quality requirements, policies and processes models) lacked capturing the quality- related standards to check that the procedures identified in the policies and the corresponding processes conformed to the applied standards which paved the road to start the fourth DSRM increment, which include the standards model.

In each DSRM increment, the interaction between the component models was illustrated using Algorithms, and the constraint rules between them were applied using the Protégé tool and SQWRL rules which facilitated the process of identifying the conflicts between the models in the SoS arrangement, and then, an interaction process was developed and applied to the SoS cancer care case study, and fully evaluated with domain experts. It was observed that this fit- for- purpose interaction process determined and resolved conflicts with the need for human intervention.

So, the OntoSoS.QM.Gov provides the processes that can be used to identify conflicts between the constituent systems' processes, policies, quality requirements and standards. Although, the current research scope does not extend to automatically resolving conflicts, it facilitates user intervention in order to resolve any reported conflicts. The conflicts identifying process reports all the information needed to facilitate resolving the reported conflicts. For example, it lists all the associated constituent systems that are involved in the conflicts. Also, it lists all related policies, QRs, processes and standards that are affected in the conflicts. Then, the user, e.g. quality officer can report the conflict to the managers of the constituent systems involved.

Due to the incremental and iterative nature of the design, demonstration and evaluation of each component model, the construction of the interaction process applied in Chapters 4-6.

4- RQ 4: *How can we evaluate the effectiveness of the process developed in RQ3 to identify and resolve semantic heterogeneities ?*

The interaction process identified in the previous research question led to inform the effectiveness of the SoS quality governance. Part of checking the adherence to quality governance was automated due to the features provided by Ontologies, e.g. using synonyms. However, part of checking the adherence to quality governance was semi-automated, as still some conflicts can be identified with the need for human involvement to be resolved.

Thus, it can be derived that using semantically- developed models helped with resolving such conflicts and identifying other conflicts. This led us to conclude that using Ontologies supported resolving conflicts. Also, the conceptualisation of the standards, the policies, the quality requirements and the domain processes using process modelling support the evaluation of the artefacts of this research.

As illustrated in Figure 8-1, RQ1 has been answered in Chapter 2, while the research questions 2,3 and 4 were incrementally answered through four DSRM increments. Thus, the four answered research questions accept the research hypothesis presented in Section 1.4 and it is concluded that ***"The interaction between semantically- enriched policies, standards, processes and quality requirements of a System of Systems arrangement and constituent heterogeneous systems results in an effective System of Systems quality governance framework"***.

8.3. Summary of Research Contributions

The main contribution to knowledge resulted in this research is to minimise the research gaps identified in Chapter 2 by developing the SoS quality governance framework that semantically enriches the main domain processes, QRs, policies and standards models interacting in SoS context to inform adherence to policies, processes, quality requirements and standards.

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

1) The OntoSoS.QM.Gov Framework is the principal research artefact using four main ontological models: standards, policy, quality requirements and processes. This framework provides a general approach to inform the adherence to quality governance in SoS context with semantic heterogeneities between constituent systems. As a step towards generalising the framework, the quality governance process applied using a sufficient and representative case study from the cancer care domain. This has been detailed in Section 3.4. Also, the design of the component models was conducted following the general and widely used ontology construction method of Noy and McGuiness (2001). They were developed based on combining the strengths of existing ontology models and the brainstorming approach to extend the functionality of deploying them in the SoS context. The OntoSoS.QM.Gov framework provides a more effective solution to existing

problems. This was reflected in the comparison conducted with the current manual methods used at the case study. For example, the process of identifying conflicts took less time than using manual methods of reviewing the documents at the case study. More examples and details provided in Table 7-5. Moreover, the scalability feature of the ontological representation of the OntoSoS.QM.Gov facilitates extending this framework to include more sub-models and more concepts within the four component models which make this framework a generalised framework.

2) The SoS quality requirements ontology model namely the OntoSoS.QR model. It is a semantically enriched model using Ontology. It was developed, demonstrated and evaluated. It can be generalised due to its expressiveness as it captures the main concepts related to quality and can be used in a different context.

3) The SoS policies ontology model namely the OntoSoS.Policy model. It is a semantically enriched model using Ontology. It was developed, demonstrated and evaluated. It was constructed using a generalised ontology construction method that enable capturing the main policy related concepts. So, it can be generalised and used in different contexts and domains.

4) The SoS processes ontology model namely the OntoSoS.Process model. It is a semantically enriched model using Ontology. It was developed, demonstrated and evaluated. The concepts of the OntoSoS.Process reflects the main concepts of BPMN notations as it was confirmed in the literature that the BPMN is a rich process modelling notation that can be effectively used to model business processes that are understandable by all stakeholders at all levels (Pant and Juric, 2008). BPMN contributes to reducing the gap between business processes and systems (Odeh et al., 2018)

5) The SoS standards ontology model namely the OntoSoS.Stand model. It is a semantically enriched model using Ontology. It was developed, demonstrated and evaluated. In order to construct the standards model, several standards in the literature were reviewed in order to identify the common concepts between them. So, the standard model included the main components of a standard without specific details regarding one particular standard. Thus, the OntoSoS.Stand is a generalised model that can be used in different contexts or domains.

6) A fit for purpose SoS quality governance process that can employ selecting quality governance related policies, processes, standards and quality requirements in SoS context. This facilitates determining and resolving conflicts with minimum human intervention. The formal representation of the SoS quality governance component

models using OWL and SQWRL statements facilitate the process of knowledge retrieval from both constituent systems level and SoS global level in relation to the concepts related to policy, process, standard and quality requirements.

7) The research design framework developed and followed in this research can be adopted and reused in similar research. The incremental and iterative method following the DSRM methodology combined with literature and case study methods provide a rigours research method that assure producing a mature version of the research framework.

8.4. Future Research Directions

Following the research findings and answering research questions and hypotheses, the following research are suggested to be taken into consideration for future development:

1) Increase the scope of the current research by enhancing the current functionalities of SoS quality governance in the OntoSoS.QM.Gov framework with conflicts impacted by the existing contracts, e.g. service level agreements between one or more constituent systems with other non-constituent systems from outside the SoS arrangement, which may include other implications especially in the case of existing non-constituent systems in other SoS arrangement. So, adding a component that deals with legal considerations and contracts violation will enhance the OntoSoS.QM.Gov framework. It was detailed in that the current framework is a scalable framework that can be extended to include more concepts and more sub-models.
2) Extend the research framework to be applied in different domains e.g. aerospace, manufacturing, etc.
3) To inform governance by linking the BPMN process model with Business workflow management systems for automating data extraction from business process domain.
4) To build a machine learning component to enhance inform quality governance from incrementally adding quality governance cases.
5) The OntoSoS.QM.Gov ontology does not deal with version control to manage different versions with different changes of the policies, processes, quality requirements and standards, instead it deals with different versions as different items. So, including the governance of change management and control is proposed to be dealt with in future.
6) Automating the instantiation of the OntoSoS.QM.Gov process and creating an automated tool with providing the facility of easy to use and navigate user interface.

Creating queries using SQRWL reported difficulties , e.g. the need of certain level of experience. Therefore, building automated tool would bring benefits for dealing with ontologies. One way to facilitate such automation is by the translation of ontologies into relational database queries, and then, to use a programming language tools to create a graphical user interface to retrieve data. Another way is by connecting the ontology to a programming language, e.g. Java programming language. A similar approach used by Ahmad (2015).

7) To provide a mechanism for Event Report Trace for SoS quality governance.

8) To extend the OntoSoS.QM.Gov framework to include the Key Performance Indicators.

9) To link the OntoSoS.QM.Gov framework to current Enterprise Information Architecture, for example, linking with the BPAOntoEIA framework (Ahmad, 2015).

10) To link the OntoSoS.QM.Gov framework with Service Oriented Architecture that map business processes to software services such as BPAOntoSoA framework (Yousef, 2010).

11) To extend the research to include the governance of change management of constituent systems' processes, policies, standards where their business processes change.

12) To apply the current research to the current COVID-19 problem. The research framework can be extended to include other constituent systems. For example, the systems related to NHS, transport, airports, government, etc. need to interoperable with each other in order to increase the quality of care to residents.

References

Aburub, F., Odeh, M., Beeson, I., 2007. Modelling non-functional requirements of business processes. Inf. Softw. Technol. 49, 1162–1171. https://doi.org/10.1016/j.infsof.2006.12.002

Ahmad, M., 2015. Semantic derivation of enterprise information architecture from Riva-based business process architecture (Ph.D.). University of the West of England.

Al Balushi, T.H., Sampaio, P.R.F., Dabhi, D., Loucopoulos, P., 2007. ElicitO: A Quality Ontology-Guided NFR Elicitation Tool, in: Sawyer, P., Paech, B., Heymans, P. (Eds.), Requirements Engineering: Foundation for Software Quality, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 306–319.

Al Balushi, T.H., Sampaio, P.R.F., Loucopoulos, P., 2013. Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool. Expert Syst. 30, 129–151.

Alexander, D., 2005. Towards the development of a standard in emergency planning. Disaster Prev. Manag. Int. J. 14, 158–175. https://doi.org/10.1108/09653560510595164

Aljawawdeh, H., 2019. An interactive metaheuristic search framework for software service identification from business process models (Ph.D.). University of the West of England.

Avizienis, A., Laprie, J.-, Randell, B., Landwehr, C., 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. 1, 11–33. https://doi.org/10.1109/TDSC.2004.2

Bannerman, P.L., 2009. Software Development Governance: A Meta-management Perspective, in: Proceedings of the 2009 ICSE Workshop on Software Development Governance, SDG ’09. IEEE Computer Society, Washington, DC, USA, pp. 3–8. https://doi.org/10.1109/SDG.2009.5071329

Bartens, Y., Haes, S. de, Lamoen, Y., Schulte, F., Voss, S., 2015. On the Way to a Minimum Baseline in IT Governance: Using Expert Views for Selective Implementation of COBIT 5, in: 2015 48th Hawaii International Conference on System Sciences. Presented at the 2015 48th Hawaii International Conference on System Sciences, pp. 4554–4563. https://doi.org/10.1109/HICSS.2015.543

Benammar, O., Elasri, H., Jebbar, M., Abderrahim, S., 2015. SECURITY POLICIES MATCHING THROUGH ONTOLOGIES ALIGNMENT TO SUPPORT SECURITY EXPERTS. J. Theor. Appl. Inf. Technol. 71, 10.

Benbasat, I., Goldstein, D.K., Mead, M., 1987. The Case Research Strategy in Studies of Information Systems. MIS Q. 11, 369–386. https://doi.org/10.2307/248684

Bergin, R., Stokes, D., 2006. Methodology or “methodolatry”? An evaluation of focus groups and depth interviews. Qual. Mark. Res. Int. J. 9, 26–37. https://doi.org/10.1108/13522750610640530

Bernaras, A., Laresgoiti, I., Bartolome, N., Corera, J., 1996. An ontology for fault diagnosis in electrical networks, in: Proceedings of International Conference on Intelligent System Application to Power Systems. Presented at the Proceedings of International Conference on Intelligent System Application to Power Systems, pp. 199–203. https://doi.org/10.1109/ISAP.1996.501068

Bianchi, T., Santos, D.S., Felizardo, K.R., 2015. Quality Attributes of Systems-of-Systems: A Systematic Literature Review, in: 2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems. Presented at the 2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems, pp. 23–30. https://doi.org/10.1109/SESoS.2015.12

Bieberstein, N., Laird, R.G., Jones, K., Mitra, T., 2008. Executing SOA: A Practical Guide for the Service-oriented Architect. IBM Press/Pearson plc.

Bischof, S., Karapantelakis, A., Nechifor, C.-S., Sheth, A.P., Mileo, A., Barnaghi, P., 2014. Semantic modelling of smart city data.

Boardman, J., Sauser, B., 2006. System of Systems - the meaning of of, in: 2006 IEEE/SMC International Conference on System of Systems Engineering. Presented at the 2006 IEEE/SMC International Conference on System of Systems Engineering, pp. 6 pp.-. https://doi.org/10.1109/SYSOSE.2006.1652284

Breitman, K.K., Leite, J.C.S. do P., 2003. Ontology as a requirements engineering product, in: Proceedings. 11th IEEE International Requirements Engineering Conference, 2003. Presented at the Proceedings. 11th IEEE International Requirements Engineering Conference, 2003., pp. 309–319. https://doi.org/10.1109/ICRE.2003.1232775

Cadavid, H., Andrikopoulos, V., Avgeriou, P., 2020. Architecting systems of systems: A tertiary study. Inf. Softw. Technol. 118, 106202. https://doi.org/10.1016/j.infsof.2019.106202

Calida, B.Y., Jaradat, R.M., Abutabenjeh, S., Keating, C.B., 2016. Governance in systems of systems: a systems-based model. Int. J. Syst. Syst. Eng. 7, 235–257. https://doi.org/10.1504/IJSSE.2016.080313

Cambridge Dictionary, 2018. . Camb. Dict. Engl. Dict. Transl. Thesaurus.

Campbell, S., Roland, M., Wilkin, D., 2001. Improving the quality of care through clinical governance. BMJ 322, 1580–1582. https://doi.org/10.1136/bmj.322.7302.1580

Castillo-Barrera, F.-E., Durán-Limón, H.A., Médina-Ramírez, C., Rodriguez-Rocha, B., 2013. A method for building ontology-based electronic document management systems for quality standards—the case study of the ISO/TS 16949:2002 automotive standard. Appl. Intell. 38, 99–113. https://doi.org/10.1007/s10489-012-0360-1

Ceccarelli, A., Bondavalli, A., Froemel, B., Hoeftberger, O., Kopetz, H., 2016. Basic Concepts on Systems of Systems, in: Bondavalli, A., Bouchenak, S., Kopetz, H. (Eds.), Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 1–39. https://doi.org/10.1007/978-3-319-47590-5_1

Chairman, Joint Chiefs of Staff Instruction 3170.01E (No. CJCSI 3170.01E), 2005. , Joint Capabilities Integration and Development System. DoD, Washington, D.C.

Chin, K.S., Yau, P.E., Wah, S.K., Khiang, P.C., 2013. FRAMEWORK FOR MANAGING SYSTEM-OF-SYSTEMS ILITIES. DSTA Horiz. 14, 10.

Curtis, B., Kellner, M.I., Over, J., 1992. Process modeling. Commun. ACM 35, 75–90. https://doi.org/10.1145/130994.130998

Dahmann, D.J.S., Baldwin, K.J., Jr, G.R., 2009. Systems of Systems and Net-Centric Enterprise Systems.

Dahmann, J., 2014. 1.4.3 System of Systems Pain Points. INCOSE Int. Symp. 24, 108–121. https://doi.org/10.1002/j.2334-5837.2014.tb03138.x

Dahmann, J.S., Baldwin, K.J., 2008. Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering, in: 2008 2nd Annual IEEE Systems Conference. Presented at the 2008 2nd Annual IEEE Systems Conference, pp. 1–7. https://doi.org/10.1109/SYSTEMS.2008.4518994

Dahmen-Lhuissier, S., 2019. ETSI - standards for interconnection & interoperability, safety, reliability, environmental care, regulators, legislators, support

[WWW Document]. ETSI. URL https://www.etsi.org/standards/why-standards (accessed 10.7.19).

Damianou, N., Dulay, N., Lupu, E., Sloman, M., 2001. The Ponder Policy Specification Language, in: Sloman, M., Lupu, E.C., Lobo, J. (Eds.), Policies for Distributed Systems and Networks, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 18–38.

Defense acquisition guidebook, 2011. . Washington, DC: Department of Defense.

Dillon, T., Potdar, V., Singh, J., Talevski, A., 2011. Cyber-physical systems: Providing Quality of Service (QoS) in a heterogeneous systems-of-systems environment, in: 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011). Presented at the 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), pp. 330–335. https://doi.org/10.1109/DEST.2011.5936595

Dobson, Glen, Lock, R., Sommerville, I., 2005. QoSOnt: a QoS ontology for service-centric systems, in: 31st EUROMICRO Conference on Software Engineering and Advanced Applications. Presented at the 31st EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 80–87. https://doi.org/10.1109/EUROMICRO.2005.49

Dobson, G., Lock, R., Sommerville, I., 2005. Quality of service requirements specification using an ontology. Proc. 5th Annu. DIRC Res. Conf. 128.

Fan, S., Hua, Z., Storey, V.C., Zhao, J.L., 2016. A process ontology based approach to easing semantic ambiguity in business process modeling. Data Knowl. Eng. 102, 57–77. https://doi.org/10.1016/j.datak.2016.01.001

Fernández-López, M., 1999. Overview of methodologies for building ontologies, in: Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolving Methods Lessons Learned and Future Trends. Presented at the Workshop Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends de la conferencia International Joint Conference for Artificial Intelligence (IJCAI'99), Facultad de Informática (UPM), Stockholm, Sweden.

FIPA Quality of Service Ontology Specification [WWW Document], 2002. URL http://www.fipa.org/specs/fipa00094/XC00094.html (accessed 11.11.18).

Flegkas, P., Russo, A., Charalambides, M., Lupu, E.C., Pavlou, G., Bandara, A.K., Dulay, N., Rubio-Loyola, J., Sloman, M., 2005. Policy Conflict Analysis for Quality of

Service Management, in: Policies for Distributed Systems and Networks, IEEE International Workshop on(POLICY). pp. 99–108. https://doi.org/10.1109/POLICY.2005.23

Flick, U., 2018. The SAGE Handbook of Qualitative Data Collection. SAGE Publications Ltd, 1 Oliver’s Yard, 55 City Road London EC1Y 1SP. https://doi.org/10.4135/9781526416070

Gandhi, S.J., Gorod, A., Sauser, B., 2012. A systemic approach to managing risks of SoS. IEEE Aerosp. Electron. Syst. Mag. 27, 23–27. https://doi.org/10.1109/MAES.2012.6226691

Gansler, J.S., Lucyshyn, W., Rigilano, J., 2012. Implementing System-of-Systems Governance. MARYLAND UNIV COLLEGE PARK CENTER FOR PUBLIC POLICY AND PRIVATE ENTERPRISE.

Garcia, D.Z.G., Toledo, M.B.F. de, 2008. Ontology-Based Security Policies for Supporting the Management of Web Service Business Processes, in: 2008 IEEE International Conference on Semantic Computing. Presented at the 2008 IEEE International Conference on Semantic Computing, pp. 331–338. https://doi.org/10.1109/ICSC.2008.35

GEIASTD0009: Reliability Program Standard for Systems Design, Development, and Manufacturing - SAE International [WWW Document], 2008. URL https://www.sae.org/standards/content/geiastd0009/ (accessed 10.7.19).

Gheorghe, A.V., Vamanu, D.V., Katina, P.F., Pulfer, R., 2018. System of Systems Governance, in: Gheorghe, A.V., Vamanu, D.V., Katina, P.F., Pulfer, R. (Eds.), Critical Infrastructures, Key Resources, Key Assets: Risk, Vulnerability, Resilience, Fragility, and Perception Governance, Topics in Safety, Risk, Reliability and Quality. Springer International Publishing, Cham, pp. 93–130. https://doi.org/10.1007/978-3-319-69224-1_4

Gobo, G., 2004. Sampling, representativeness and generalizability, in: Qualitative Research Practice. SAGE Publications Ltd, London, pp. 405–426.

Gómez-Pérez, A., 2001. Evaluation of ontologies. Int. J. Intell. Syst. 16, 391–409. https://doi.org/10.1002/1098-111X(200103)16:3<391::AID-INT1014>3.0.CO;2-2

Gorod, A., Merchant, S., Hallo, L., 2018. Toward Systemic Governance of Cancer Treatment as a System of Systems, in: 2018 13th Annual Conference on System of Systems Engineering (SoSE). Presented at the 2018 13th Annual Conference on System of Systems

Engineering (SoSE), IEEE, Paris, pp. 556–560. https://doi.org/10.1109/SYSOSE.2018.8428792

Gorod, A., White, B.E., Ireland, V., Gandhi, S.J., Sauser, B., 2014. Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering. CRC Press.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5, 199–220. https://doi.org/10.1006/knac.1993.1008

Grüninger, M., Fox, M.S., 1995. Methodology for the Design and Evaluation of Ontologies.

Guarino, N., 1998. Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy, 1st ed. IOS Press, Amsterdam, The Netherlands, The Netherlands.

Guide to the Systems Engineering Body of Knowledge (SEBoK) v1.4 [WWW Document], 2015. URL https://www.feacinstitute.org/resource/ea-zone/item/guide-to-the-systems-engineering-body-of-knowledge-sebok-v1-4 (accessed 8.25.18).

Hall-May, M., Kelly, T., 2006. Towards Conflict Detection and Resolution of Safety Policies.

Hendrikse, J.W., Hendrikse, L., 2003. Business Governance Handbook: Principles and Practice. Juta and Company Ltd.

HermiT 1.3.8 [WWW Document], 2013. . Httpwwwhermit-Reason.

Hevner, A., 2007. A Three Cycle View of Design Science Research. Scand. J. Inf. Syst. 19.

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. MIS Q 28, 75–105.

Holley, K., Arsanjani, A., 2010a. 100 SOA Questions: Asked and Answered, 01 edition. ed. Prentice Hall, Upper Saddle River, NJ.

Holley, K., Arsanjani, A., 2010b. 100 SOA Questions: Asked and Answered, 1st ed. Prentice Hall Press, Upper Saddle River, NJ, USA.

Holley, K., Palistrant, J., Graham, S., 2006. Effective SOA governance., IBM White Paper. IBM Corporation.

Howarth, T., Watson, P., 2011. Construction Quality Management: Principles and Practice, 1 edition. ed. Routledge, London.

INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th ed, 2015. . John Wiley & Sons Inc, Hoboken, NJ, USA.

Iqbal, R., Murad, M.A.A., Mustapha, A., Sharef, N.M., 2013. An Analysis of Ontology Engineering Methodologies: A Literature Review. Res. J. Appl. Sci. Eng. Technol. 6, 2993–3000. https://doi.org/10.19026/rjaset.6.3684

ISACA, 2012. COBIT 5: A Business Framework for the Governance and Management of Enterprise IT. ISACA.

ISO 9001:2015 - Quality management systems -- Requirements [WWW Document], 2015. URL https://www.iso.org/standard/62085.html (accessed 9.7.18).

ISO 9241-210:2010 - Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems [WWW Document], 2010. URL https://www.iso.org/standard/52075.html (accessed 9.11.18).

ISO/IEC 17065:2012 - Conformity assessment Requirements for bodies certifying products, processes and services [WWW Document], 2012. URL https://www.iso.org/standard/46568.html (accessed 9.7.18).

ISO/IEC 21827:2008 - Information technology -- Security techniques -- Systems Security Engineering -- Capability Maturity Model® (SSE-CMM®) [WWW Document], 2008. URL https://www.iso.org/standard/44716.html (accessed 9.7.18).

ISO/IEC 27001:2013 [WWW Document], 2013. . ISO. URL http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/45/54534.html (accessed 10.7.19).

ISO/IEC Guide 65:1996 - General requirements for bodies operating product certification systems [WWW Document], 1996. URL https://www.iso.org/standard/26796.html (accessed 9.7.18).

J. Black, P. Koopman, 2009. System safety as an emergent property in composite systems, in: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks. Presented at the 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, pp. 369–378. https://doi.org/10.1109/DSN.2009.5270316

Jamshidi, M., 2008. System of systems engineering - New challenges for the 21st century. IEEE Aerosp. Electron. Syst. Mag. 23, 4–19. https://doi.org/10.1109/MAES.2008.4523909

JCI Accreditation Standards for Hospitals, 2017. . 54321.

John, B.E., Packer, H., 1995. Learning and Using the Cognitive Walkthrough Method: A Case Study Approach, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’95. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 429–436. https://doi.org/10.1145/223904.223962

Juristo, N., Morant, J.L., 1998. Common framework for the evaluation process of KBS and conventional software. Knowl.-Based Syst. 11, 145–159. https://doi.org/10.1016/S0950-7051(98)00047-1

Kaiya, H., Saeki, M., 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation, in: 14th IEEE International Requirements Engineering Conference (RE’06). Presented at the 14th IEEE International Requirements Engineering Conference (RE’06), pp. 189–198. https://doi.org/10.1109/RE.2006.72

Kaiya, H., Saeki, M., 2005. Ontology based requirements analysis: lightweight semantic processing approach, in: Fifth International Conference on Quality Software (QSIC’05). Presented at the Fifth International Conference on Quality Software (QSIC’05), pp. 223–230. https://doi.org/10.1109/QSIC.2005.46

Kassab, M., Ormandjieva, O., Daneva, M., 2009. An Ontology Based Approach to Non-functional Requirements Conceptualization, in: 2009 Fourth International Conference on Software Engineering Advances. Presented at the 2009 Fourth International Conference on Software Engineering Advances, pp. 299–308. https://doi.org/10.1109/ICSEA.2009.50

Keating, C.B., 2015. Complex system governance: Theory to practice challenges for system of systems engineering, in: 2015 10th System of Systems Engineering Conference (SoSE). Presented at the 2015 10th System of Systems Engineering Conference (SoSE), pp. 226–231. https://doi.org/10.1109/SYSOSE.2015.7151955

Keating, C.B., Katina, P.F., 2019. Complex system governance: Concept, utility, and challenges. Syst. Res. Behav. Sci. 36, 687–705. https://doi.org/10.1002/sres.2621

Kossmann, M., Wong, R., Odeh, M., Gillies, A., 2008. Ontology-driven Requirements Engineering: Building the OntoREM Meta Model, in: 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications. Presented at the 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications, pp. 1–6. https://doi.org/10.1109/ICTTA.2008.4530315

Kotonya, G., Sommerville, I., 1998. Requirements Engineering: Processes and Techniques, 1st ed. Wiley Publishing.

Lee, K., Jeon, J., Lee, W., Jeong, S.-H., Park, S.-W., 2003. QoS for Web Services: Requirements and Possible Approaches.

Lee, S., Shim, J., Wu, C., 2002. A meta model approach using UML for task assignment policy in software process, in: Ninth Asia-Pacific Software Engineering Conference, 2002. Presented at the Ninth Asia-Pacific Software Engineering Conference, 2002., pp. 376–382. https://doi.org/10.1109/APSEC.2002.1183007

Maier, M.W., 1998. Architecting principles for systems-of-systems. Syst. Eng. 1, 267–284. https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D

Mansour, A., 2016. King Hussein Cancer Center expansion to house state-of-the-art facilities.

Mansouri, M., Gorod, A., Wakeman, T.H., Sauser, B., 2009. A systems approach to governance in Maritime Transportation System of Systems, in: 2009 IEEE International Conference on System of Systems Engineering (SoSE). Presented at the 2009 IEEE International Conference on System of Systems Engineering (SoSE), pp. 1–6.

Mansouri, M., Mostashari, A., 2010. A systemic approach to governance in extended enterprise systems, in: 2010 IEEE International Systems Conference. Presented at the 2010 IEEE International Systems Conference, pp. 311–316. https://doi.org/10.1109/SYSTEMS.2010.5482432

March, S.T., Smith, G.F., 1995. Design and natural science research on information technology. Decis. Support Syst. 15, 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

March, S.T., Storey, V.C., 2008. Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. MIS Q. 32, 725–730. https://doi.org/10.2307/25148869

Martins, F.M., Trienekens, J., Omta, O., 2017. Differences in quality governance: the case of the Brazilian pork chain. Br. Food J. https://doi.org/10.1108/BFJ-09-2016-0418

Maximilien, E.M., Singh, M.P., 2004. A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Comput. 8, 84–93.

Musa, N., Ibrahim, D.H.A., Bolhassan, N.A., Abdullah, J., Kulathuramaiyer, N., Khairuddin, M.N., 2014. An IT governance framework for achieving the development of academic programme in higher institutions: A case of Universiti Malaysia Sarawak

(UNIMAS), in: The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M). Presented at the The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), pp. 1–6. https://doi.org/10.1109/ICT4M.2014.7020673

National Academies of Sciences, E., 2009. Communicating the Value of Research: Contractor's Final Report. https://doi.org/10.17226/23034

Natschlager, C., 2011. Towards a BPMN 2.0 Ontology, in: Dijkman, R., Hofstetter, J., Koehler, J. (Eds.), Business Process Model and Notation. Springer Berlin Heidelberg, pp. 1–15.

Nicola, A.D., Lezoche, M., Missikoff, M., 2007. An Ontological Approach to Business Process Modeling, in: IICAI.

Nicolae, O., Cosulschi, M., Giurca, A., Wagner, G., 2009. Towards a BPMN Semantics Using UML Models, in: Ardagna, D., Mecella, M., Yang, J. (Eds.), Business Process Management Workshops, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 585–596.

Nielsen, C.B., Larsen, P.G., Fitzgerald, J., Woodcock, J., Peleska, J., 2015. Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Comput Surv 48, 18:1–18:41. https://doi.org/10.1145/2794381

Noy, N.F., McGuinness, D.L., 2001. Ontology Development 101: AGuide to Creating Your First Ontology 28.

Nunamaker, J.F., Chen, M., 1990. Systems development in information systems research, in: Twenty-Third Annual Hawaii International Conference on System Sciences. Presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, pp. 631–640 vol.3. https://doi.org/10.1109/HICSS.1990.205401

Nunamaker, J.F., Jr., Chen, M., Purdin, T.D.M., 1990. Systems Development in Information Systems Research. J Manage Inf Syst 7, 89–106. https://doi.org/10.1080/07421222.1990.11517898

Odeh, Y., Tbaishat, D., Al-Okaily, A., Khudirat, S., Al-Smadi, O., Hejazi, A., Sharma, S., Tbakhi, A., Odeh, M., 2018. Informing Business Process Models Adherence to Protocols via Business Process Modelling: The Case of Cell Therapy and Applied Genomics in Cancer Care, in: 2018 1st International Conference on Cancer Care Informatics (CCI). Presented at the 2018 1st International Conference on Cancer Care Informatics (CCI), pp. 82–99. https://doi.org/10.1109/CANCERCARE.2018.8618205

Oliver, D., Lainhart, J., 2012. COBIT 5: Adding Value Through Effective Geit. EDPACS 46, 1–12. https://doi.org/10.1080/07366981.2012.706472

Ould, 1995. Business Processes: Modelling and Analysis for Re-engineering, First Edition edition. ed. John Wiley & Sons, Chichester ; New York.

Ould, M.A., 2005. Business Process Management: A Rigorous Approach. British Computer Society, Swindon, UK.

Palmieri, C., Bird, E., Simcock, R., 2013. ABC of Cancer Care. John Wiley & Sons.

Pant, K., Juric, M., 2008. Business Process Driven SOA Using BPMN and BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture. Packt Publishing.

Patton, M.Q., 2015. Qualitative research & evaluation methods: integrating theory and practice, Fourth. ed. SAGE, Los Angeles.

Patton, M.Q., 1990. Qualitative evaluation and research methods. SAGE Publications, inc.

Payne, V.L., Metzler, D.P., 2005. Hospital care watch (HCW): an ontology and rule-based intelligent patient management assistant, in: 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05). Presented at the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05), pp. 479–484. https://doi.org/10.1109/CBMS.2005.64

Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J., 2006. The Design Science Research Process: A Model for Producing and Presenting Information Systems Research, in: In: 1st International Conference on Design Science in Information Systems and Technology (DESRIST. pp. 83–106.

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24, 45–77. https://doi.org/10.2753/MIS0742-1222240302

Phan, T., Han, J., Schneider, J.-G., Wilson, K., 2008. Quality-Driven Business Policy Specification and Refinement for Service-Oriented Systems, in: Bouguettaya, A., Krueger, I., Margaria, T. (Eds.), Service-Oriented Computing – ICSOC 2008, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 5–21.

Protege [WWW Document], 2013. . https://protege.stanford.edu.

Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., 2018a. OntoSoS.QR: Semantic Representation of Quality Requirements metamodel for Systems of Systems, in: 2018 International Arab Conference on Information Technology (ACIT). Presented at the 2018 International Arab Conference on Information Technology (ACIT), Lebanon, pp. 1–7. https://doi.org/10.1109/ACIT.2018.8672693

Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., Tbakhi, A., Mansour, A., 2018b. A Semantically-Enriched Quality Governance Framework for Systems of Systems applied to Cancer Care, in: 1st International Conference on Cancer Care Informatics. Presented at the 2018 1st International Conference on Cancer Care Informatics, Amman, Jordan, pp. 15–24. https://doi.org/10.1109/CANCERCARE.2018.8618263

Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., Traish, I., Tbakhi, A., 2017. OntoSoS.QM.Gov: A Quality Governance Framework for Systems of Systems, in: Eighth International Conference on Complex Systems Design & Management, CSD&M. Presented at the Complex Systems Design & Management, CESAM Community, Paris, pp. 159–172.

Quality governance in the NHS - A guide for provider boards, 2011. National Quality Board, UK.

Ran, S., 2003. A Framework for discovering web services with desired quality of services attributes.

Robinson, W.N., 2006. A requirements monitoring framework for enterprise systems. Requir. Eng. 11, 17–41. https://doi.org/10.1007/s00766-005-0016-3

Rospocher, M., Ghidini, C., Serafini, L., 2014. An ontology for the Business Process Modelling Notation, in: FOIS. https://doi.org/10.3233/978-1-61499-438-1-133

Sabata, B., Chatterjee, S., Davis, M., Sydir, J.J., Lawrence, T.F., 1997. Taxonomy for QoS specifications, in: Proceedings Third International Workshop on Object-Oriented Real-Time Dependable Systems. Presented at the Proceedings Third International Workshop on Object-Oriented Real-Time Dependable Systems, pp. 100–107. https://doi.org/10.1109/WORDS.1997.609931

Samad, T., Annaswamy, A.M., 2011. The Impact of Control Technology. IEEE Control Systems Society.

Schneider, J., Ebringer, T., Rogers, T., Phan, T., Han, J., 2008. A Survey of Policy-Based Management Approaches for Service Oriented Systems, in: 2008 19th Australian

Software Engineering Conference(ASWEC). pp. 392–401. https://doi.org/10.1109/ASWEC.2008.56

Sciore, E., Siegel, M., Rosenthal, A., 1994. Using semantic values to facilitate interoperability among heterogeneous information systems. ACM Trans. Database Syst. TODS 19, 254–290. https://doi.org/10.1145/176567.176570

Sheth, A.P., Larson, J.A., 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv. CSUR 22, 183–236.

Siegel, J., 2014. Model Driven Architecture (MDA): MDA Guide Revised (MDA Guide revision 2.0) (No. Report ID: Version 2). Object Management Group.

Simon, H.A., 1996. The Sciences of the Artificial. MIT Press.

Sledge, C.A., 2010. Reports from the Field on System of Systems Interoperability Challenges and Promising Approaches (No. CMU/SEI-2010-TR-013). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Sloane, E., Gehlot, V., Way, T., Beck, R., 2008. System of Systems Issues for the 2008 U. S. National Healthcare Information Network Remote Patient Monitoring Requirements, in: 2008 2nd Annual IEEE Systems Conference. Presented at the 2008 2nd Annual IEEE Systems Conference, pp. 1–5. https://doi.org/10.1109/SYSTEMS.2008.4519018

Smith, M.K., Welty, C., McGuinness, D.L., 2004. OWL Web Ontology Language Guide [WWW Document]. W3C Recomm. URL https://www.w3.org/TR/owl-guide/ (accessed 10.16.18).

Snir, Y., Ramberg, Y., Strassner, J., Cohen, R., Moore, B., 2003. Policy Quality of Service (QoS) Information Model.

Sommerville, I., 2011. Software Engineering, 9th ed. Pearson, Boston.

Sommerville, I., 2010. Software Engineering: United States Edition, 9 edition. ed. Pearson, Boston.

Sommerville, I., 2001. Software engineering. Addison-Wesley.

Sommerville, I., 2000. Software engineering, 6th ed. ed, International computer science series. Addison-Wesley, Harlow, England ; New York.

Susanti, F., Sembiring, J., 2011. The mapping of interconnected SOA governance and ITIL v3.0, in: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics. Presented at the Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp. 1–5. https://doi.org/10.1109/ICEEI.2011.6021574

Systems Engineering Guide for Systems of Systems, V 1.0, 2008. . Washington, DC.

Systems of Systems (SoS) - SEBoK [WWW Document], 2017. . Guide Syst. Eng. Body Knowl. SEBoK. URL https://www.sebokwiki.org/wiki/Systems_of_Systems_(SoS) (accessed 1.16.19).

Tondello, G.F., Siqueira, F., 2008. The QoS-MO Ontology for Semantic QoS Modeling, in: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08. ACM, New York, NY, USA, pp. 2336–2340. https://doi.org/10.1145/1363686.1364239

Tran, V.X., Tsuji, H., Masuda, R., 2009. A new QoS ontology and its QoS-based ranking algorithm for Web services. Simul. Model. Pract. Theory, Dependable Service-Orientated Computing Systems 17, 1378–1398. https://doi.org/10.1016/j.simpat.2009.06.010

Tsang, E.W.K., 2014. Generalizing from Research Findings: The Merits of Case Studies. Int. J. Manag. Rev. 16, 369–383. https://doi.org/10.1111/ijmr.12024

Uschold, M., Gruninger, M., 2004. Ontologies and Semantics for Seamless Connectivity. SIGMOD Rec 33, 58–64. https://doi.org/10.1145/1041410.1041420

Uszok, A., Bradshaw, J.M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., Aitken, S., 2004. KAoS policy management for semantic Web services. IEEE Intell. Syst. 19, 32–41. https://doi.org/10.1109/MIS.2004.31

Vaishnavi, V.K., Kuechler, W., 2015. Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 2Nd Edition, 2nd ed. CRC Press, Inc., Boca Raton, FL, USA.

Vaishnavi, V.K., Kuechler, W., Jr., 2007. Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 1st ed. Auerbach Publications, Boston, MA, USA.

Valverde, R., Toleman, M., 2007. Ontological Evaluation of Business Models: Comparing Traditional and Component-Based Paradigms in Information Systems Re-

Engineering, in: Sharman, R., Kishore, R., Ramesh, R. (Eds.), Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems, Integrated Series in Information Systems. Springer US, Boston, MA, pp. 49–77. https://doi.org/10.1007/978-0-387-37022-4_3

Van Arnum, P., 2004. Information Technology Insights:Cyber Security Ushers IT Into Corporate Governance Practices. Chemical Market Reporter.

Vierhauser, M., Rabiser, R., Grünbacher, P., 2016. Requirements monitoring frameworks: A systematic review. Inf. Softw. Technol. 80, 89–109. https://doi.org/10.1016/j.infsof.2016.08.005

Wagner, S., 2013. Software Product Quality Control. Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38571-1

Yang, S., Guo, J., Wei, R., 2017. Semantic interoperability with heterogeneous information systems on the internet through automatic tabular document exchange. Inf. Syst. 69, 195–217. https://doi.org/10.1016/j.is.2016.10.010

Yin, R.K., 2003. Case Study Research: Design and Methods. SAGE.

Yousef, R., Odeh, M., Coward, D., Sharieh, A., 2009a. BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures, in: 2009 Second International Conference on the Applications of Digital Information and Web Technologies. Presented at the 2009 Second International Conference on the Applications of Digital Information and Web Technologies, pp. 50–55. https://doi.org/10.1109/ICADIWT.2009.5273939

Yousef, R., Odeh, M., Coward, D., Sharieh, A., 2009b. Translating RAD business process models into BPMN, in: 2009 Second International Conference on the Applications of Digital Information and Web Technologies. Presented at the 2009 Second International Conference on the Applications of Digital Information and Web Technologies, pp. 75–83. https://doi.org/10.1109/ICADIWT.2009.5273913

Yousef, R.M., 2010. BPAOntoSOA: a semantically enriched framework for deriving SOA candidate software services from Riva-based business process architecture. University of the West of England, Bristol.

Zhou, C., Chia, L.-T., Lee, B.-S., 2005. Web Services Discovery with DAML-QoS Ontology. Int. J. Web Serv. Res. IJWSR 2, 43–66. https://doi.org/10.4018/jwsr.2005040103

Appendices

Appendix I: Research Publications

This section lists the research communication.

I.1. Journal and Conferences Publications

This research has resulted in the following publications so far:

- Qaddoumi, E., Odeh, M., Khan, Z. and Kossmann, M., 2018, November. OntoSoS. QR: Semantic representation of quality requirements metamodel for systems of systems. In *2018 International Arab Conference on Information Technology (ACIT)* (pp. 1-7). IEEE.
- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., Tbakhi, A. and Mansour, A., 2018, November. A semantically-enriched quality governance framework for systems of systems applied to Cancer Care. In *2018 1st International Conference on Cancer Care Informatics (CCI)* (pp. 15-24). IEEE.
- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., Traish, I. and Tbakhi, A., 2017, December. OntoSoS. QM. Gov: A quality governance framework for systems of systems. In *Eighth International Conference on Complex Systems Design & Management, CSD&M* (pp. 159-172).
- Kossmann, M, Samhan, A, Odeh, M, Qaddoumi, E, Tbakhi, A, Watts, S. Extending the scope of configuration management for the development and life cycle support of systems of systems—An ontology-driven framework applied to the Enceladus Submarine Exploration Lander. *Systems Engineering*. 2020; 23: 366– 391. https://doi.org/10.1002/sys.21532

I.2. Presentations and Posters

- Qaddoumi, E. (2019). OntoSoS.QGov: A Semantically-Enriched Quality Governance Framework for Systems of Systems applied to Cancer Care [power point presentation]. Presented at Software Engineering Research Group(SERG) Monthly seminar, UWE, Bristol, UK, 3 April.
- Qaddoumi, E. (2017). OntoSoS.QM.Gov.CC**:** Ontology-based metamodels of Systems of Systems for Quality Management Governance applied to Cancer Care [power point presentation]. Presented at King Hussain Cancer Canter (KHCC), Amman, Jordan, 27 August

- Qaddoumi, E. (2017). A Semantically Enriched Quality Governance Framework for Systems of Systems [power point presentation]. Presented at Software Engineering Research Group(SERG) Monthly seminar, UWE, Bristol, UK, 10 May.
- Qaddoumi, E. (2016) Quality Governance Framework for Systems of Systems Applied to Airbus Avionics [poster]. In: *EU Ashley Project Public Forum. 2016*, Bristol, UK, 25 October.
- Qaddoumi, E. (2016). Quality Governance Framework for Systems of Systems Applied to Airbus Avionics [power point presentation]. Presented at EU Ashley Project Public Forum, Bristol, UK, 25 October.
- Qaddoumi, E. (2016). Quality Governance for Cancer Care in Systems of Systems Context Applied to KHCC [power point presentation]. Presented at King Hussain Cancer Canter (KHCC), Amman, Jordan, 26 July

I.3. In-Preparation Publications

- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., [Expected 2021]. A Layered framework for System of Systems Quality Governance. Information and Software Technology Journal. [In preparation]
- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., [Expected 2021]. OntoSoS.Process: Semantic Representation of a Process model applied to the process of handling outpatient's reception at Cell Therapy and Applied Genomics-related systems. [In preparation]
- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., [Expected 2021]. OntoSoS.Stand: Semantic Representation of a Quality Standard model for Systems of Systems applied to JCIAS: Joint Commission International Accreditation Standards. [In preparation]
- Qaddoumi, E., Odeh, M., Khan, Z., Kossmann, M., [Expected 2021]. OntoSoS.QPolicy: Semantic Representation of a Quality Policy model for Systems of Systems applied to Cancer Care, in: 2nd International Conference on Cancer Care Informatics. [In preparation]

Appendix II: Examples of Models Representation

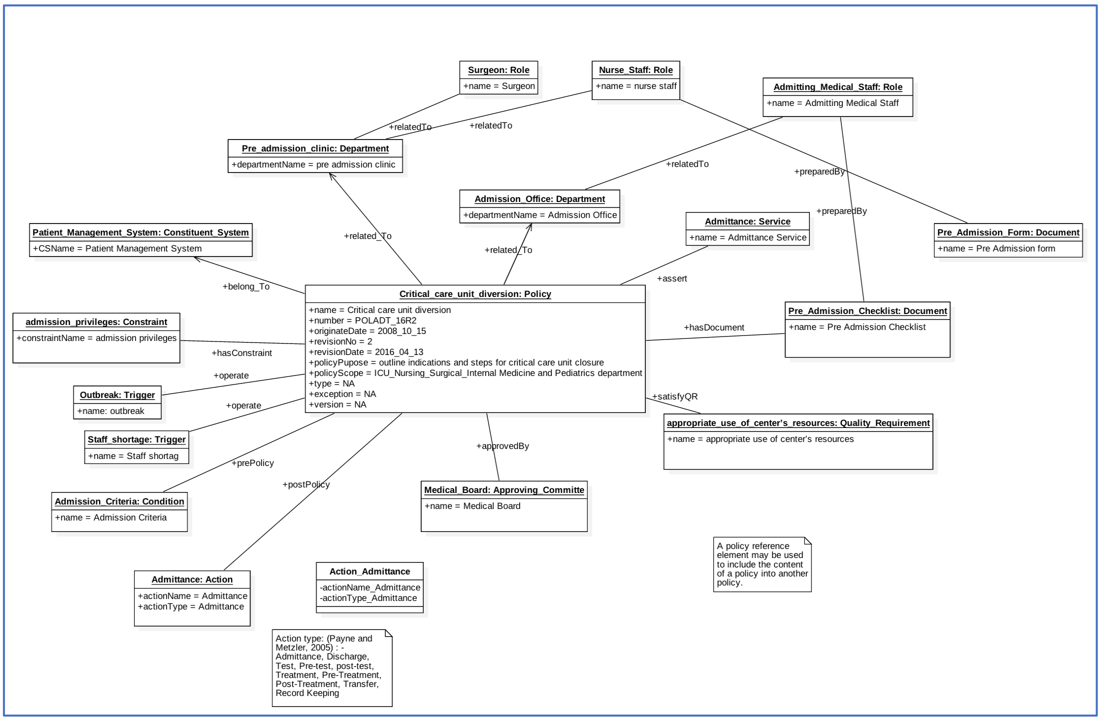

Surgeon: Role
+name = Surgeon
Nurse_Staff: Role
+name = nurse staff
Admitting_Medical_Staff: Role
+name = Admitting Medical Staff
+relatedTo
+relatedTo
Pre_admission_clinic: Department
+departmentName = pre admission clinic
+relatedTo
+preparedBy
Admission_Office: Department
+departmentName = Admission Office
+preparedBy
Admittance: Service
+name = Admittance Service
Pre_Admission_Form: Document
+name = Pre Admission form
Patient_Management_System: Constituent_System
+CSName = Patient Management System
+related_To
+related_To
+assert
+belong_To
Critical_care_unit_diversion: Policy
+name = Critical care unit diversion
+number = POLADT_16R2
+originateDate = 2008_10_15
+revisionNo = 2
+revisionDate = 2016_04_13
+policyPupose = outline indications and steps for critical care unit closure
+policyScope = ICU_Nursing_Surgical_Internal Medicine and Pediatrics department
+type = NA
+exception = NA
+version = NA
Pre_Admission_Checklist: Document
+name = Pre Admission Checklist
+hasDocument
admission_privileges: Constraint
+constraintName = admission privileges
+hasConstraint
+operate
Outbreak: Trigger
+name: outbreak
+operate
+satisfyQR
appropriate_use_of_center's_resources: Quality_Requirement
+name = appropriate use of center's resources
Staff_shortage: Trigger
+name = Staff shortag
+approvedBy
+prePolicy
+postPolicy
Medical_Board: Approving_Committe
+name = Medical Board
Admission_Criteria: Condition
+name = Admission Criteria
A policy reference element may be used to include the content of a policy into another policy.
Admittance: Action
+actionName = Admittance
+actionType = Admittance
Action_Admittance
-actionName_Admittance
-actionType_Admittance
Action type: (Payne and Metzler, 2005) : - Admittance, Discharge, Test, Pre-test, post-test, Treatment, Pre-Treatment, Post-Treatment, Transfer, Record Keeping

Figure 1: An Example of Policy Object Model- Critical Care Unit Diversion Policy

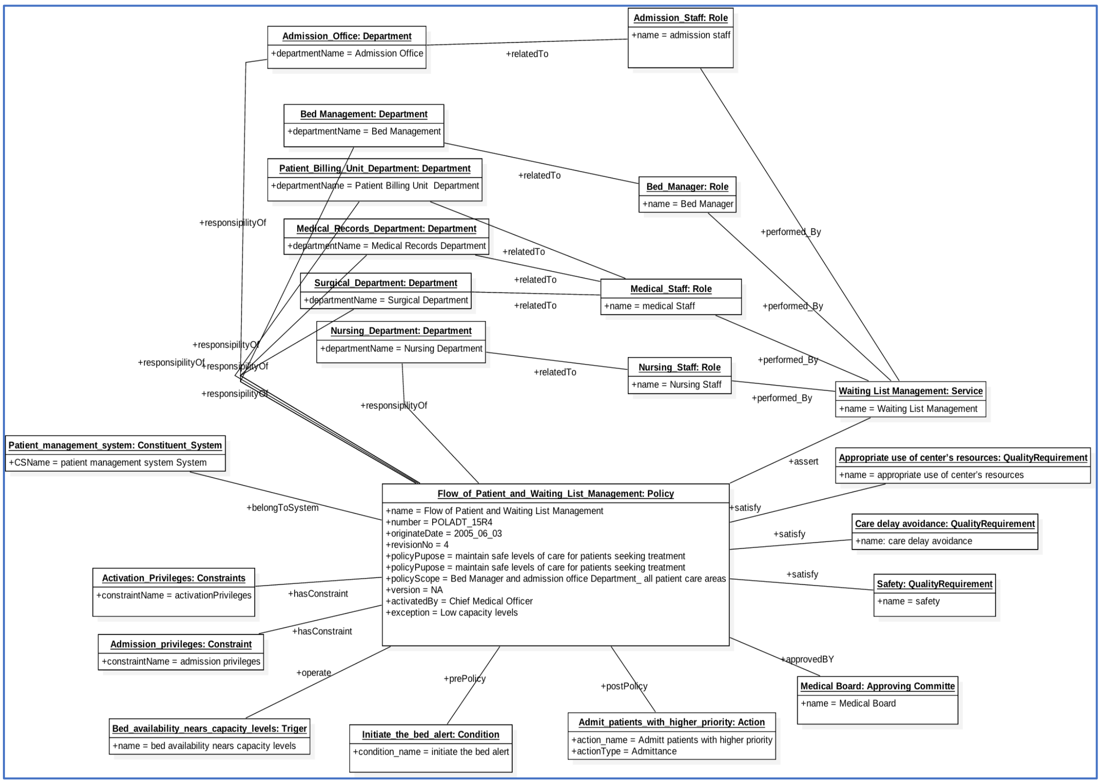

Admission_Staff: Role
+name = admission staff
Admission_Office: Department
+departmentName = Admission Office
+relatedTo
Bed Management: Department
+departmentName = Bed Management
Patient_Billing/Unit_Department: Department
+departmentName = Patient Billing Unit Department
+relatedTo
Bed_Manager: Role
+name = Bed Manager
+responsipilityOf
Medical_Records_Department: Department
+departmentName = Medical Records Department
+performed_By
+relatedTo
Surgical_Department: Department
+departmentName = Surgical Department
+relatedTo
Medical_Staff: Role
+name = medical Staff
+relatedTo
+performed_By
Nursing_Department: Department
+departmentName = Nursing Department
+responsipilityOf
+responsipilityOf
+responsipilityOf
+performed_By
Nursing_Staff: Role
+name = Nursing Staff
+relatedTo
+responsipilityOf
+performed_By
Waiting List Management: Service
+name = Waiting List Management
+responsipilityOf
Patient_management_system: Constituent_System
+CSName = patient management system System
+assert
Appropriate use of center's resources: QualityRequirement
+name = appropriate use of center's resources
Flow_of_Patient_and_Waiting_List_Management: Policy
+name = Flow of Patient and Waiting List Management
+number = POLADT_15R4
+originateDate = 2005_06_03
+revisionNo = 4
+policyPupose = maintain safe levels of care for patients seeking treatment
+policyPupose = maintain safe levels of care for patients seeking treatment
+policyScope = Bed Manager and admission office Department_ all patient care areas
+version = NA
+activatedBy = Chief Medical Officer
+exception = Low capacity levels
+belongToSystem
+satisfy
+satisfy
Care delay avoidance: QualityRequirement
+name: care delay avoidance
+satisfy
Activation_Privileges: Constraints
+constraintName = activationPrivileges
+hasConstraint
Safety: QualityRequirement
+name = safety
+hasConstraint
Admission_privileges: Constraint
+constraintName = admission privileges
+approvedBY
+operate
+prePolicy
+postPolicy
Medical Board: Approving Committe
+name = Medical Board
Admit_patients_with_higher_priority: Action
+action_name = Admitt patients with higher priority
+actionType = Admittance
Bed_availability_nears_capacity_levels: Triger
+name = bed availability nears capacity levels
Initiate_the_bed_alert: Condition
+condition_name = initiate the bed alert

Figure 2: An Example of Policy Object Model- Flow of Patient and Waiting list Management Policy

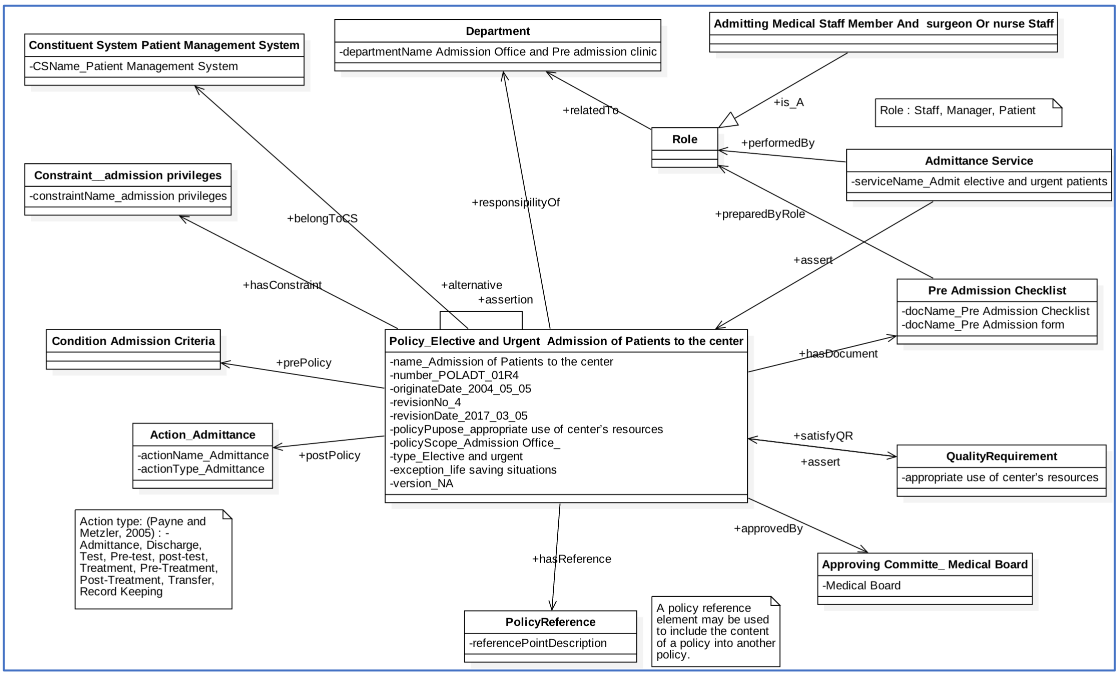

Constituent System Patient Management System
-CSName_Patient Management System
Department
-departmentName Admission Office and Pre admission clinic
Admitting Medical Staff Member And surgeon Or nurse Staff
+relatedTo
+is_A
Role : Staff, Manager, Patient
Role
+performedBy
Admittance Service
-serviceName_Admit elective and urgent patients
Constraint__admission privileges
-constraintName_admission privileges
+responsibilityOf
+belongToCS
+preparedByRole
+assert
+hasConstraint
+alternative
+assertion
Pre Admission Checklist
-docName_Pre Admission Checklist
-docName_Pre Admission form
Condition Admission Criteria
Policy_Elective and Urgent Admission of Patients to the center
-name_Admission of Patients to the center
-number_POLADT_01R4
-originateDate_2004_05_05
-revisionNo_4
-revisionDate_2017_03_05
-policyPupose_appropriate use of center's resources
-policyScope_Admission Office_
-type_Elective and urgent
-exception_life saving situations
-version_NA
+prePolicy
+hasDocument
Action_Admittance
-actionName_Admittance
-actionType_Admittance
+postPolicy
+satisfyQR
+assert
QualityRequirement
-appropriate use of center's resources
Action type: (Payne and Metzler, 2005) : - Admittance, Discharge, Test, Pre-test, post-test, Treatment, Pre-Treatment, Post-Treatment, Transfer, Record Keeping
+approvedBy
+hasReference
Approving Committe_ Medical Board
-Medical Board
PolicyReference
-referencePointDescription
A policy reference element may be used to include the content of a policy into another policy.

Figure 3: An Example of Policy Object Model- Elective and Urgent Admission of Patients to the Centre Policy

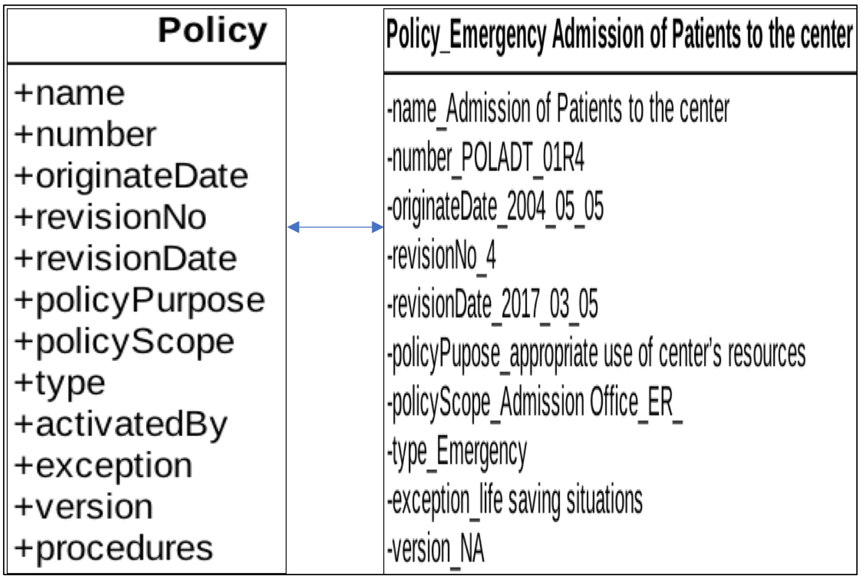

Policy
+name
+number
+originateDate
+revisionNo
+revisionDate
+policyPurpose
+policyScope
+type
+activatedBy
+exception
+version
+procedures
Policy_Emergency Admission of Patients to the center
-name_Admission of Patients to the center
-number_POLADT_01R4
-originateDate_2004_05_05
-revisionNo_4
-revisionDate_2017_03_05
-policyPupose_appropriate use of center's resources
-policyScope_Admission Office_ER_
-type_Emergency
-exception_life saving situations
-version_NA

Figure 4: A Snapshot of the values of the attributes for the policy of "Emergency admission of patients to the main centre"

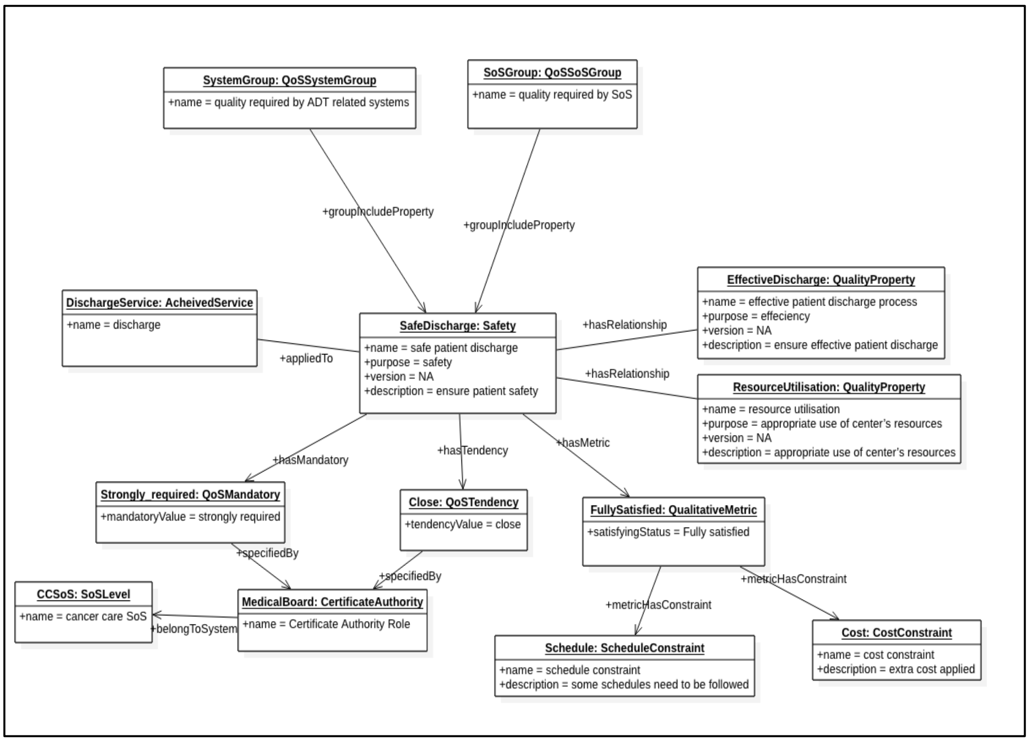

SystemGroup: QoSSystemGroup
+name = quality required by ADT related systems
SoSGroup: QoSSoSGroup
+name = quality required by SoS
+groupIncludeProperty
+groupIncludeProperty
DischargeService: AcheivedService
+name = discharge
+appliedTo
SafeDischarge: Safety
+name = safe patient discharge
+purpose = safety
+version = NA
+description = ensure patient safety
+hasRelationship
EffectiveDischarge: QualityProperty
+name = effective patient discharge process
+purpose = effeciency
+version = NA
+description = ensure effective patient discharge
+hasRelationship
ResourceUtilisation: QualityProperty
+name = resource utilisation
+purpose = appropriate use of center's resources
+version = NA
+description = appropriate use of center's resources
+hasMandatory
+hasTendency
+hasMetric
Strongly_required: QoSMandatory
+mandatoryValue = strongly required
Close: QoSTendency
+tendencyValue = close
FullySatisfied: QualitativeMetric
+satisfyingStatus = Fully satisfied
+specifiedBy
+specifiedBy
+metricHasConstraint
+metricHasConstraint
CCSoS: SoSLevel
+name = cancer care SoS
+belongToSystem
MedicalBoard: CertificateAuthority
+name = Certificate Authority Role
Schedule: ScheduleConstraint
+name = schedule constraint
+description = some schedules need to be followed
Cost: CostConstraint
+name = cost constraint
+description = extra cost applied

Figure 5: An Example of Quality Requirements Object Model- Safety Quality Requirement

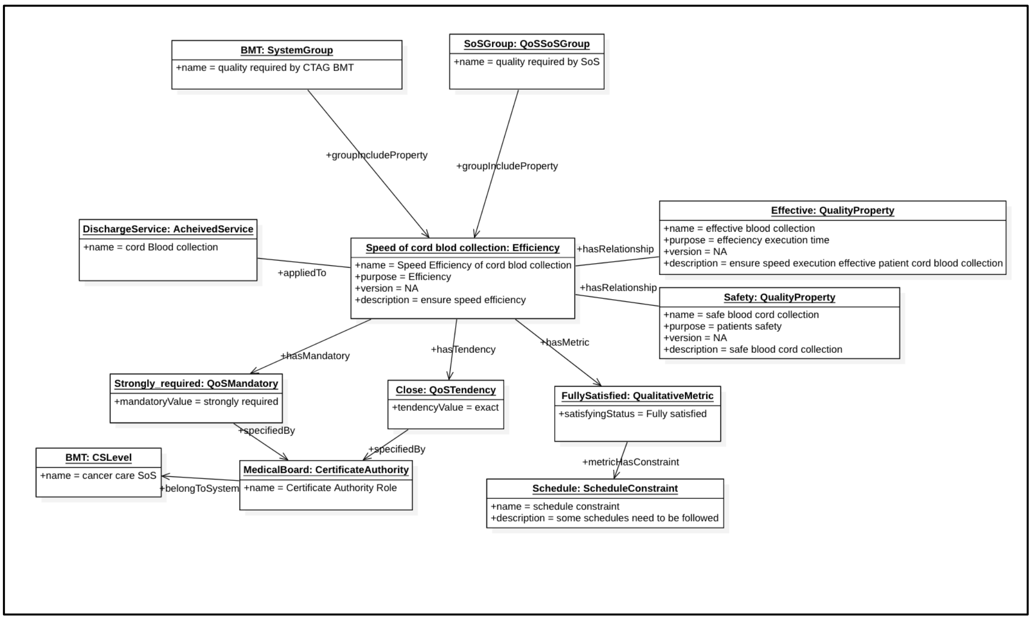

BMT: SystemGroup
+name = quality required by CTAG BMT
SoSGroup: QoSSoSGroup
+name = quality required by SoS
+groupIncludeProperty
+groupIncludeProperty
DischargeService: AcheivedService
+name = cord Blood collection
+appliedTo
Speed of cord blod collection: Efficiency
+name = Speed Efficiency of cord blod collection
+purpose = Efficiency
+version = NA
+description = ensure speed efficiency
+hasRelationship
Effective: QualityProperty
+name = effective blood collection
+purpose = effeciency execution time
+version = NA
+description = ensure speed execution effective patient cord blood collection
+hasRelationship
Safety: QualityProperty
+name = safe blood cord collection
+purpose = patients safety
+version = NA
+description = safe blood cord collection
+hasMandatory
+hasTendency
+hasMetric
Strongly_required: QoSMandatory
+mandatoryValue = strongly required
Close: QoSTendency
+tendencyValue = exact
FullySatisfied: QualitativeMetric
+satisfyingStatus = Fully satisfied
+specifiedBy
+specifiedBy
+metricHasConstraint
BMT: CSLevel
+name = cancer care SoS
+belongToSystem
MedicalBoard: CertificateAuthority
+name = Certificate Authority Role
Schedule: ScheduleConstraint
+name = schedule constraint
+description = some schedules need to be followed

Figure 6: An Example of Quality Requirements Object Model- Efficiency Quality Requirement

Appendix III: Evaluation-related Documents

III.1. Participant Information Sheet

Participant Information Sheet

Faculty of Environment and Technology,
Software Engineering Research Group (SERG).

Research Title: 'A Semantically- Enriched Quality Governance Framework in the Systems of Systems (SoS) context applied to cancer care'.

Dear Participant, you are being invited to take part in a research study. Before you decide whether to participate, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully. Ask questions if anything you read is not clear or if you would like more information and decide whether you wish to take part. If you choose not to participate nothing will change and you will not be affected in any way. If you change your mind, you will have the right to withdraw at any time within two weeks from the date of submitting your answers. Take time to decide whether or not you wish to take part. Thank you for reading.

What is the purpose of this study?

The purpose of this study is to evaluate the design and the development of a Systems of systems quality governance framework and its related models. In this research, a quality governance process that models and implements the interaction between four models will be developed and assessed. These models are: (1) standards, (2) policies, (3) processes and (4) quality requirements.

Why have I been chosen?

You have been chosen because you are allocated middle or top-level managerial role in your organisation.

Do I have to take part?

Please note that your participation in this project is completely voluntary, and there is no obligation to participate. It is up to you to decide whether or not to take part. If you choose to participate, you will be given this information sheet to keep and be asked to sign a consent form. If you decide to participate, you are still free to withdraw from this research and without giving a reason. If you decide to withdraw, you need to let me know within **one week** from the date of submitting your answers. You can withdraw and discontinue the participation and you will not be penalised, neither your manager/ supervisor will be informed. If you withdraw your consent, any recorded data will be discarded and not be used in the study. However, after one week from the date of submitting your answers, you are agreeing to participate and cannot withdraw after this point.

What will happen to me if I take part? What do I have to do?

If you decide to take part in this research, you will be given some time to read this information sheet, then you will be asked to sign the consent form. Afterwards, in order to assess the quality governance process and the overall quality governance framework proposed in this research, participants will be asked to answer different questions related to various elements of the developed model. It is expected that 3-5 semi-structured interviews and walkthrough sessions will be conducted with each participant to evaluate the research framework. The expected time of any interview and walkthrough session will be between 1-2 hours. The length of the session will be determined according to participants' availability/preference.

The provided feedback is highly valuable for the research and will allow to further develop and mature the framework of this research.

Will my taking part in this study be kept confidential? What will happen to the results of the research study?

Please note that to protect the participant's confidentiality, no personal information will be collected that would identify any of the participants, and the results of this study will be used only for scholarly purposes and the information provided will be used in this research as an input to process and generalise findings that may be used in publications, reports, web pages, and other research outputs. This information may only be shared amongst the members of the research team, the results cannot be used to identify any of the participants. Furthermore, all the collected data will be stored in a password protected electronic format. Also, all the hard copies of the collected data will be stored in a locked locker in UWE.

After completing this study, all hard copies will be discarded following UWE related procedures. Also, all the digital media will be distroyed after a period of 12 months following publication of the study results. Moreover, the findings of the research will be made available online and the KHCC participants will be informed through sending them a link.

Questions about the research or your rights as a participant

Please do not hesitate to contact the research team if you have any questions or concerns regarding your participation or the research.

Contact us at:
Eman Qaddoumi (PGR student): Eman3.Qaddoumi@uwe.ac.uk
Mohammed Odeh (DoS) : Mohammed.Odeh@uwe.ac.uk

THANK YOU FOR TAKING TIME TO READ THIS INFORMATION SHEET

III.2. Participant Consent Form

Participant Consent Form

PhD Study: ‘A Semantically- Enriched Quality Governance Framework in the System of Systems Context Applied to Cancer Care’

Please confirm that you understand and agree to the following:

- I am over the age of 18.
- I have read and understood the “Participant information sheet”.
- I have been given the opportunity to ask questions about the study.
- I have had the opportunity to clarify any aspects of the research and I have had the study explained to my satisfaction.
- I understand that by consenting to take part in this study, I can still withdraw within one week of submitting my answers without being obliged to give any reasons.
- I understand that after submitting the answers, I cannot withdraw my data.
- I understand that I will not be personally identified in any report, and the results communicated by this study cannot be used to identify me.
- I understand that this information will be used for the purpose set out in the information sheet and in the research- associated publications and presentations. My consent is conditional upon the university complying with the duties and obligation under the General Data Protection Regulation (GDPR).
- I understand that the information I provide will be used in this research as an input to process and generalise findings that may be used in publications, reports, web pages, and other research outputs.

- I confirm that I have read and understood the aforementioned agreement, and I agree to take part in the research study.

□ YES

□ No

Participant’s Signature: ______________________________ Date: __________________________

Researcher’s Signature: ______________________________ Date: _________________________

III.3. Manual Walkthrough to Validate Governance Models

Step 1: Manual “walkthrough” and semi-structured interview to validate the abstract level of the quality governance model.

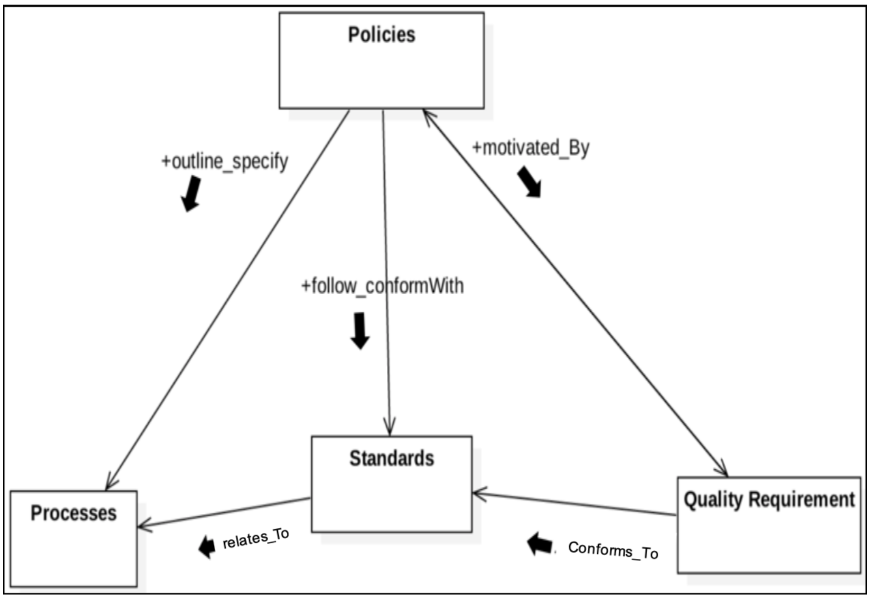


Figure 1: The Abstract level of the Quality Governance Model

- Please answer the following questions regarding the quality governance model:

Table 1: General questions to validate the quality governance model

	Questions	Answers	Further Suggestions (Remarks)
1	Do you agree that the formation of quality-related *Policies* motivated by the definitions of the *Quality Requirements*?		
2	Do you suggest any other relations between policies and quality requirements models?		
3	Do you agree that the *Policies* need to follow/conform with the *Standards*?		
4	Do you suggest any other relations between policies and standards models?		

5	Do you agree that the quality- related *Policies* need to outline/specify the *Processes (workflows)* followed in the organisation?		
6	Do you suggest any other relations between policies and processes models?		
7	Do you agree that the *Quality Requirements* conform to *Standards*?		
8	Do you suggest any other relations between quality requirements and standards models?		
9	Do you agree that the *Processes(workflows)* related to *Standards*?		
10	Do you suggest any other relations between *Processes(workflows)* and standards models?		
- Please write here any further comments:			

Appendix IV: Interaction Views

IV.1. Interaction Views

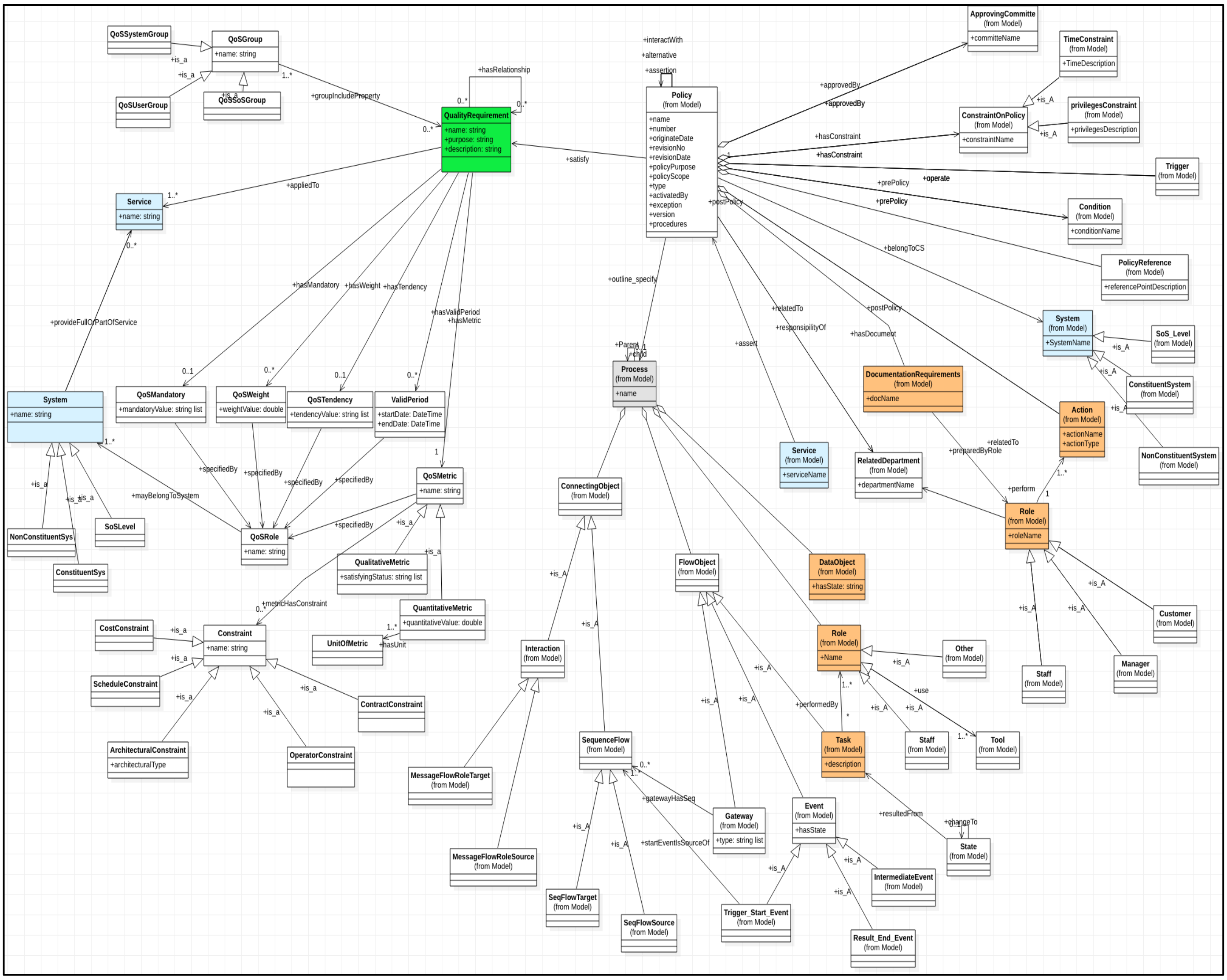


Figure 1: Interactions in Increment

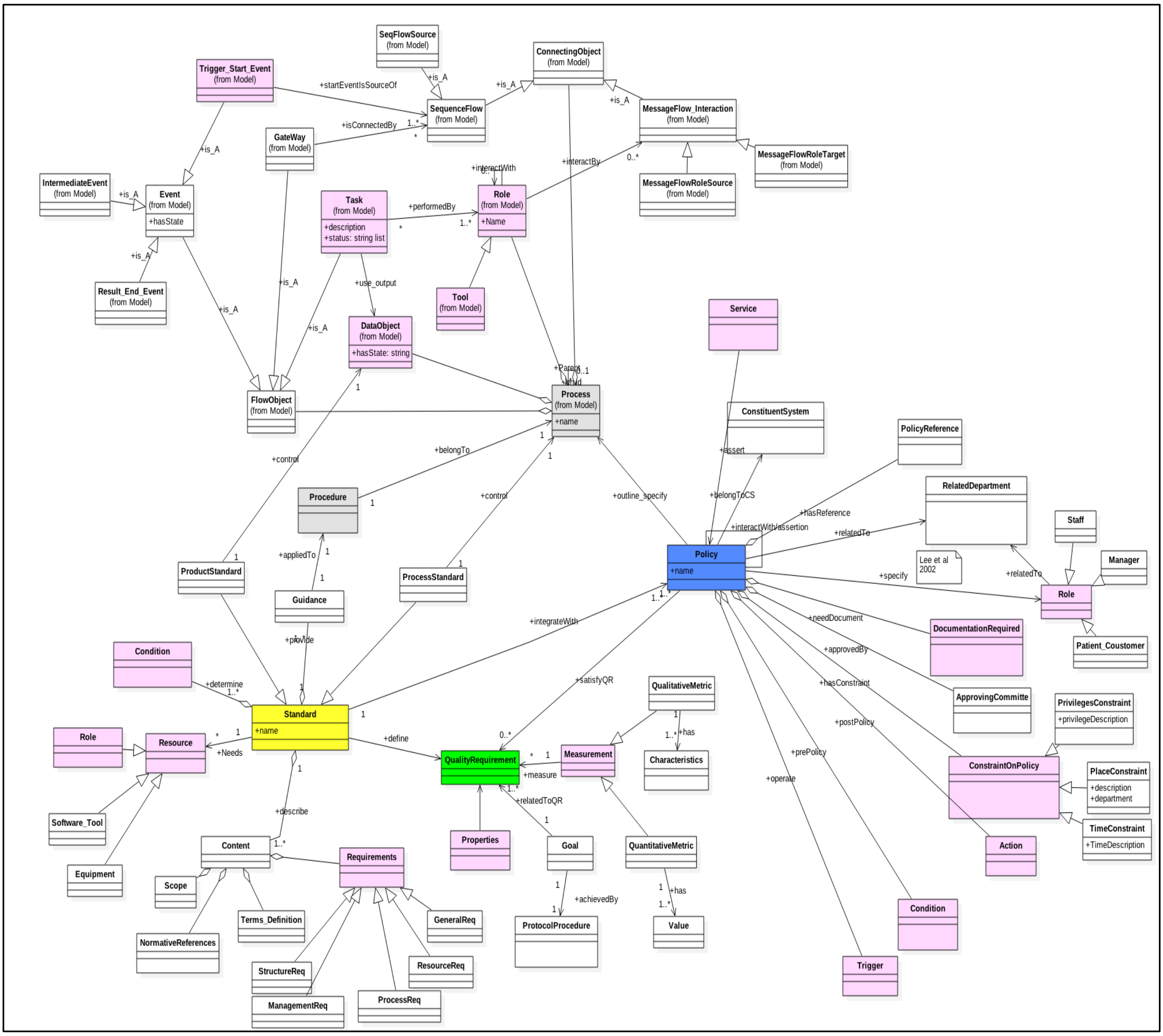


Figure 2: Interactions in Increment 4

IV.2. Interaction Sheet

Please check the following interaction to validate the interaction and to clarify related constituent systems

QR	QR at KHCC	Policies	Process	Standard
Safety	safety committee , international safety policies and goals, event	QM and Patient safety policies		
Health and safety risk mitigation	reportssafety	Discharge of Patients from the Centre POLADT-05R4	Discharge process	JCIAS: ACC.4
Health and safety risk mitigation	safety continuity of care	Discharge Planning Process POLADT13R4	Discharge process	JCIAS: ACC.5, AOP.1.1
Health and safety risk mitigation	safety continuity of care	Admission of Patients POLADT-01R4	Triage process to prioritize patients with immediate needs	JCIAS: ACC.1.1,
Health and safety risk mitigation	safety continuity of care	Admission of Patients POLADT-01R5	Admission process (Patient Reception Process)	JCIAS: ACC.1.1,
Health and safety risk mitigation	safety continuity of care	Flow of Patient and Waiting List Management POLADT-15R4	Admission process	JCIAS: ACC.5.2
Health and safety risk mitigation	ensure safety for both patient and doner	Guidelines for Processing Allogeneic Cellular Products IPPLAB-BMT-55-R3	CTAG BMT Handle precollection process	CAP Transfusion Medicine Checklist
Health and safety risk mitigation	ensure safety for both patient and doner	Patient File Preparation Procedure IPPCTAG-BMT-27-R5	CTAG BMT Handle precollection process	CAP Transfusion Medicine Checklist
Health and safety risk mitigation	ensure safety for both patient and doner	PatientDonor Preparation Procedure IPPCTAG-BMT-35-R5	CTAG BMT Handle precollection process	CAP Transfusion Medicine Checklist
Health and safety risk mitigation	ensure safety for both patient and doner	Review of Required Tests IPPCTAG-BMT-07-R4	CTAG BMT Handle precollection process	CAP Transfusion Medicine Checklist

Health and safety risk mitigation	ensure safety for both patient and doner	Critical care unit diversion POLADT 16R2	Transfer process	JCIAS: ACC.5.1
Economic risk mitigation	ensure financial coverage	Flow Cytometry charging and LIS IPPCTAG-FC.26/R1	flow cytometry specimens charging patients process	in-house
Environmental risk mitigation	ensure financial coverage	High Alert Medications Policy POLPT30R5	Alert Medications process	JCIAS: IPSG.3,
Environmental risk	ensure financial coverage	Medication Administration POLNUR-	Medications management	JCIAS: IPSG.3,
Environmental risk mitigation	ensure financial coverage	Nursing Care Pre Invasive /Surgical procedure POLNUR -05R6		JCIAS: IPSG.3.1,
Environmental risk mitigation	ensure financial coverage	Hand Hygiene Policy (POLINC -01)	Handle health care-associated infection process/ Handhygiene guidelines throughout	JCIAS: IPSG.5,
Environmental risk mitigation	ensure financial coverage	Fall Prevention and Management (POLNUR-15)	the hospitalHandle assessing all inpatients for fall risk process	JCIAS: IPSG.6,
Security		MRC policies (mainly)		
Confidentiality	Confidentiality	Confidentiality of Patient Information POLMRC -09R2	CTAG BMT2 Handle collection process	JCIAS: MOI.1.9
Confidentiality	Confidentiality	Confidentiality of Patient Information POLMRC -09R3	process of requesting patient's test results	JCIAS: MOI.1.10
Confidentiality	Confidentiality	Release of Confidential Patient's Information POLMRC -10R3	process of requesting patient's test results	JCIAS: MOI.1.11
Confidentiality	Confidentiality	Patient File Preparation Procedure IPPCTAG-BMT-27-R5	CTAG BMT2 Handle collection process	JCIAS: MOI.1.12
Confidentiality	Confidentiality	Patient File Preparation Procedure IPPCTAG-BMT-27-R6	process of requesting patient's test results	JCIAS: MOI.1.13
Integrity	Confidentiality	Access to Confidential & Restricted Information/Records POLMRC -12R4	process of requesting patient's files	JCIAS: MOI.1.14
Performance efficiency				

Resource utilization	Efficient use of centre resources	Discharge of Patients from the Centre POLADT-05R4	discharge process	JCIAS: MOI.1.16
Resource utilization	Efficient use of centre resources	Flow of Patient and Waiting List Management POLADT-15R04	admission process	JCIAS: MOI.1.17
Resource utilization	Efficient use of centre resources	Emergent and elective admission POLADT-01R4	admission process	JCIAS: MOI.1.18
Resource utilization	Efficient use of centre resources	Critical care unit diversion POLADT -	admission process	JCIAS: MOI.1.19
Resource utilization	Efficient use of centre resources	Discharge Planning Process POLADT13R4	discharge process	JCIAS: MOI.1.20

Space	efficient use of spaces and rooms	Discharge Planning Process POLADT13R5	discharge process	JCIAS: MOI.1.21
Execution Time	efficiency of cord blod collection	Cord Blood Collection Procedure IPPCTAG-BMT-15-R4	CTAG BMT2 Handle collection process	CAP
Execution Time	efficiency of cord blod collection	PatientDonor Preparation Procedure IPPCTAG-BMT-35-R5	CTAG BMT2 Handle collection process	
Response Time	speed of operation/ deliver medical records to care providers in a timely manner	Requesting Medical Records POLMRC 07R3 Cytogenetics quality management and	process of requesting patient's files Cyto1 Handle bone morrow	in-house
Throughput	Throughput	quality control plan IPPLAB-CYG.11/R6Cytogenetics quality management and	and peripheral blood analysis process	CAP (GLP)
Throughput	Throughput	quality control plan IPPLABCYG.11/R6	Cyto2 Handle solid tissues by FISH technology process	CAP (GLP)
Interface/Interoperability: Usability				
Usability	usability of VISTA	Flow Cytometry charging and LIS IPPCTAG-FC.26/R1	flow cytometry specimens charging patients process	in-house
Usability	usability of ATS	Flow Cytometry test samples management IPPCTAG-FC.04/R6	flow cytometry specimens charging patients process	in-house
Reliability	bed alert to ask Riyad			

Availability	continuity of care, waiting list, bed availability	Discharge Planning Process POLADT13R4	process of requesting patient's f	ACC
Availavility	continuity of care, waiting list, bed availability	Requesting Medical Records POLMRC 07R3	process of requesting patient's f	In-house
Fault tolerance	VISTA operates as intended despite the presence of software faults	Requesting Medical Records POLMRC 07R4	process of requesting patient's f	In-house
Recoverability	VISTA recover the data directly affected and re-establish the desired state of the system	Requesting Medical Records POLMRC 07R5	process of requesting patient's f	In-house
Accuracy : Functional correctness	Improve accuracy of patient identification	Patient Identification Policy (POLPIC01)	Patient identification process	JCIAS: IPSG.1
Accuracy	Improve accuracy of patient identification	Medication Administration Policy (POLNUR-11)	Patient identification process	JCIAS: IPSG.2
Accuracy	Improve accuracy of patient identification	Blood Product Administration & Monitoring (POLNUR-10)	Patient identification process	JCIAS: IPSG.3
Accuracy	Improve accuracy of patient identification	Patient Transportation Policy (POLNUR 22)	-Patient identification process	JCIAS: IPSG.4
Accuracy	Improve accuracy of patient identification	Radiation Oncology - Patient Identification (POLRAD -06)	Patient identification process	JCIAS: IPSG.5
Accuracy	Improve accuracy of patient identification	Patient movement and Care in Radiology (POLDGR -97)	Patient identification process	JCIAS: IPSG.6
Accuracy	Improve accuracy of patient identification	Radiology request form(POLDGR-67)	Patient identification process	JCIAS: IPSG.7
Accuracy	Improve accuracy of patient identification	DPLM Labeling Procedure IPPLABGEN.33	CTAG BMT2 Handle collection process	JCIAS: IPSG.8
Accuracy	Improve accuracy of patient identification	DPLM Labeling Procedure IPPLABGEN.34	Patient identification process	JCIAS: IPSG.9
Accuracy	Improve accuracy of patient identification	Handling Laboratory Verbal Requests and Results IPPLAB-GEN.24	Patient identification process	JCIAS: IPSG.10

Accuracy	Improve accuracy of patient identification	Reporting of Lab Results IPPLAB-GEN.32	Patient identification process	JCIAS: IPSG.11
Accuracy	Effective and accurate communication	Laboratory Critical Results Policy (IPLAB-GEN.12)	process of verbal and/or telephone communication	JCIAS: IPSG.2,
Accuracy	Effective and accurate communication	Examination Reporting Policy (POLDGR-55)	among caregiversprocess of verbal and/or telephone communication	JCIAS: IPSG.2,
Accuracy	Effective and accurate communication	Handling Laboratory Verbal Requests and Results IPPLAB-GEN.24	among caregiversprocess of verbal and/or telephone communication	JCIAS: IPSG.2,
Accuracy	Effective and accurate communication	Physician's Handover POLADT-21	among caregiversprocess of verbal and/or telephone communication	JCIAS: IPSG.2,
Accuracy	Effective and accurate communication	Shift Endorsements POLNUR 14	reporting process of critical results of diagnostic tests	JCIAS: IPSG.2.1,
Accuracy	Effective and accurate	Patient Transportation POLNUR-22R4	handover communication	JCIAS: IPSG.2.2,
Accuracy	Effective and accurate communication	Handover Guidelines for Clinical Nurse Coordinators POLNUR-98	handover communication process	JCIAS: IPSG.2.2,
Accuracy	prepare a specimens correctly	Flow Cytometry test samples management IPPCTAG-FC.04/R6	Outpatient sample reception process	In-house, Internal CTAG IPP
Accuracy	prepare a specimens correctly	Flow Cytometry test samples management IPPCTAG-FC.04/R7	Handle sample rejection process	In-house, Internal CTAG IPP
Accuracy	Improve accuracy	Correct Side, Site, Procedure and Patient Undergoing Surgery Policy (POLSRG-	Handle sample rejection process	JCIAS: IPSG.4,
Accuracy	Improve accuracy	01)Surgical / Invasive Procedure Time – Out Policy (POLNUR - 281)	Handle sample rejection process	JCIAS: IPSG.4,

Accuracy	Improve accuracy	Nursing Care Pre Invasive/Surgical procedure POLNUR-05	Handle sample rejection process	JCIAS: IPSG.4,
Accuracy	minimise medical records errors	Monitoring of Medical Records Completion policy POLMRC -06R3	monitor/review medical records process	JCIAS: ACC, Access to Care and continuity of Care