INCREASING THE LEVEL OF REUSE OF COMPLETE PROCESS MODELS IN A PROCESS MODELLING REPOSITORY

BY

Ross Shearer Veitch

Thesis submitted for the Degree of

Doctor of Philosophy

in Information Systems

October 2024

Plagiarism Declaration

"This thesis/dissertation represents my own work, both in concept and execution, and has been submitted to the Turnitin module. I confirm that my supervisor has seen my report. Any concerns revealed by such have been resolved with my supervisor."

Name: Ross Shearer Veitch

Student number: VTCROS002

Signed by candidate

Signed by candidate

Date: 15 October 2024

DEDICATION

To my children, David and Emma, thank you for the inspiration and support.

To my late father, David, you departed this world too early. Thank you for choosing mom!

To my late mother, Kit, who raised my sister and me in a single-parent household, thank you for all the wisdom and support through the years.

And to everyone who assisted mom, you helped make my sister and me who we are today.

Acknowledgements

I would like to express my gratitude to my supervisor, Professor Lisa Seymour, for all her advice, guidance, patience, and input on this PhD journey. Without her, this journey would not have been completed.

I would also like to thank the faculty of Information Systems at UCT, specifically Professor Michael Kyobe, Professor Wallace Chigona, and Professor Ojelanki Ngwenyama, for providing the foundations and philosophy of knowledge and the PhD journey.

Thanks to the participants in this study for taking the time to sit through interviews and answer my questions. Without you, this study would not have been possible.

A special thanks to my family, David and Emma, and my late mother, who have all at various times had to make sacrifices for this study. Without your support and motivation, this would not have been possible.

Abstract

Title of Thesis: Increasing the Level of Reuse of Complete Process Models in a Process Modelling Repository

Business Process Modelling has become a common activity in organisations. However, as the number of process models increases, the number of duplicated models also increases because the same process is often carried out in different parts of the organisation. As a result, the level of process model reuse has been found to be surprisingly low, and this leads to wasted effort modelling processes which already exist. Given this situation, the motivation for this study was to develop a modelling method which increased the level of reuse of process models in a repository. We named this modelling method EPReS (Enterprise Process Reuse System). This study specifically focussed on improving factors which impacted increased process model reuse. Incorporated in this study is also the first method to objectively measure the Level of Reuse of process models in a repository. Finally, this study describes the system dynamics that are setup within an organisation when process models are not reused.

A pragmatic research paradigm was adopted to answer the research questions. Using a Design Science Research strategy, EPReS was designed and tested in a real-world setting in a large South African financial services organisation. A mixed-methods approach was used, consisting of semi-structured interviews conducted with Process Modellers and a quantitative analysis of the process repository. EPReS incorporates reuse concepts from the software development domain.

This study makes three contributions to knowledge. Firstly, this study develops a new modelling method which increases the level of reuse of process models in a process repository. Secondly, the study develops a measure which can be used to determine the level of process model reuse in a process repository. Finally, it explains the consequences for the organisation, using system dynamics, of not reusing process models, and illustrates how vicious cycles are established, which result in continually increasing costs. This study has made practical contributions as it also offers the vendors of process modelling tools opportunities to further enhance their products by incorporating elements of EPReS to facilitate the increased reuse of complete process models by users.

This study has limitations. This research project was carried out in a single organisation and data collection from semi-structured interviews was limited to those Process Modellers who were using the new modelling method. It is possible that other Process Modellers could identify additional factors impacting the consequences of not reusing complete process models. Also, the new

modelling method designed is dependent on the underlying architecture of the modelling tool in use and therefore, this method will not be suitable for every available process modelling tool. Future studies may improve the method further or better quantify the consequences of not reusing process models.

Definitions, Acronyms, and Abbreviations

Abbreviation	Meaning
Activity	The symbol used in a model to represent an action or a set of actions executed.
Agile	A project management methodology where project deliverables are broken down into deliverables that can be completed within a short period (known as a Sprint). A Sprint typically lasts two calendar weeks.
ABU	Architecture Business Unit
BA	Business Architecture
BAG	Business Architecture Guild
BPM	Business Process Modelling
Capability	A capability defines "what the business does" as opposed to "how the business does it."
Control-Flow Model	Describes a process model which can represent the sequence of activities required to achieve a particular outcome. Examples include the EPC (and row/column variations), BPMN Process, and BPMN Collaboration Diagrams.
Current Account	A type of bank account. Also known as a Checking account.
Dev	Development Environment
EA	Enterprise Architecture
EPReS	Enterprise Process Reuse System (the new modelling method)
ERD	Entity Relationship Diagram
FAD	Function Allocation Diagram
FEAPO	The Federation of Enterprise Architecture Professional Organizations
Function	The object type used in ARIS to represent an Activity or another process model. A Function can decompose into a more detailed process model. In such an instance, the Function is the "parent object" of the process model.
GK	Gatekeeper
GUID	Global Unique Identifier

Abbreviation	Meaning
L1, L2, L3, L4, L5	Refers to the Architectural level of a model or Function. L1=Level 1, L2=Level2, etc. A Level 2 model/Function is more granular than a L1 model/Function.
L2 Function	Refers to a Function positioned at Level 2 in the process architecture. A L2 Function is the parent object of an end-to-end L3 process model (Scenario).
L3 Function	Refers to a Function positioned at Level 3 in the process architecture. A Level 3 Function is the name of an instance of a Capability.
L4 Function	Refers to a Function positioned at Level 4 in the process architecture. A Level 4 Function is the parent object of a L5 process model, which is the most granular process model.
Level of Reuse	A measure (percentage) of the number of control-flow models in the repository which are referenced by 2 or more other control-flow models in the repository.
LOR	Level of reuse
PA	Process Architecture
Parent Function	A Function that decomposes into a process model is known as the "Parent Function" of that model. Architecturally, if a process model is at Level X, then the parent Function will be at Level X-1.
PM	Process modeller
PPM	Process Publication Model
Prod	Production Environment
PSD	Process Selection Diagram
Publication	Publication refers to the set of models and objects which were published.
Publish	To publish a project is the act of taking the process models (and objects) created or updated for a project and moving these models (and objects) from the Development Environment to the Production Environment. This includes ensuring the correct model and object management required when updating an existing process model or modelling a new process.
QC Report	Quality Control Report

Abbreviation	Meaning
Scenario	A Scenario is the end-to-end process from a customer perspective. It will consist of a L2 Function with a L3 process model assigned to it. The object is known as the Scenario object, and the model is known as the Scenario model.
Script	A set of instructions written in Javascript, which the modelling tool executes. In ARIS, the repository can be manipulated using scripts. It can be compared to a macro in Microsoft Excel.
Top-Down Method	The new modelling method developed by this research project became known as the "Top-Down" method within the business unit because it forced process modellers to start modelling at L2/L3 in the process architecture. This is the same as EPReS.

Publications

During the course of this study, two papers were written, which were accepted and presented at international conferences. These papers, as listed below, presented findings and contributions related to this study:

Veitch, R., & Seymour, L. F. (2017). Business Process Model Reuse In A Multi-Channel / Multi-Product Environment–Problem Identification And Tentative Design. *MCIS 2017 Proceedings. 25.*

Veitch, R., & Seymour, L. F. (2019). Measuring Business Process Model Reuse in a Process Repository. In Di Francescomarino C., R. Dijkman, & U. Zdun (Eds.), *Business Process Management Workshops. BPM 2019. Lecture Notes in Business Information Processing, vol 362*. Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-030-37453-2_58

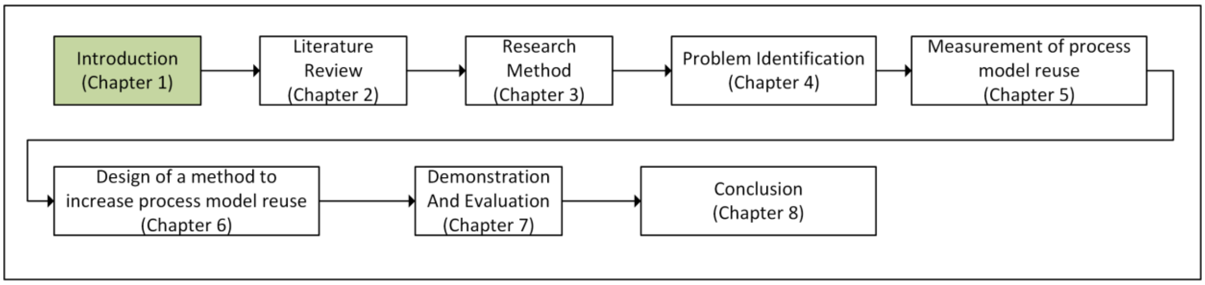
Table of Contents

List of Tables

List of Figures

List of Equations

1 Chapter 1 Introduction



This chapter explains the background to this research project and then elaborates on the problem statement, research questions, methods used, and finally, the outline of this thesis.

Businesses face pressures on many fronts today. From increasing regulatory requirements, increasing competition, and rapidly evolving technologies to growing customer demands for tailored products and improved service, a business must attempt to balance these demands while continuing to remain profitable. At the heart of delivering anything (a physical product or a service) to the customer lies the business process by which this is achieved.

Possibly the most often quoted definition of a business process is “a structured, measured set of activities designed to produce a specific output for a particular customer or market” (Davenport, 1993, p.5). Process modelling has been defined differently by different authors (Kesari et al., 2003; Kim, 1995; Rosemann et al., 2009; Sayuri et al., 2011). However, the common theme in all these definitions is that a graphical representation of the business process is produced. Process models are built using modelling tools, a modelling language, and modelling conventions (known as methods). The look and feel of the process model depend on the modelling language used and the modelling method adopted. Several studies have compared modelling techniques (Aguilar-Saven, 2004; Recker et al., 2009). The selection of the tool, technique, and method is important as not all modelling techniques are suitable for all purposes (Macintosh, 1993; Recker et al., 2009), and not all tools support all techniques. In short, the selection of the tool, technique, and method is dependent on the purpose of the model being constructed (Macintosh, 1993; Recker, 2006; Recker et al., 2009).

BPM has become an important practice that is common to most enterprises because a process model can be used for different purposes in different areas of an organisation. Some of the uses of process models listed in the literature include communication between stakeholders, input to training material, requirements definitions, identification of SOA services and business process management. (Alotaibi, 2016; Davies et al., 2006; Kesari et al., 2003; Krogstie et al., 2006; Rosemann et al., 2009). Furthermore, building process models is an essential step in all process automation initiatives (Radgui et al., 2013; van der Aalst, 2013). Among the benefits of process modelling cited in the literature are facilitating process improvement, improving understanding of business processes, improving communication between stakeholders, improving process analysis (Dalal et al., 2004; Indulska, Green, et al., 2009), managing complexity and documenting procedures (van der Aalst, 2013). In one study, the top ten perceived business process modelling benefits were listed as process improvement, understanding, communication, model-driven process execution, process performance measurement, process analysis, knowledge management, reuse, process simulation and change management (Indulska et al., 2009)

1.1 Problem Statement

While the concept of reuse in BPM is not new, research relating to process model reuse has focussed on topics such as the identification of similar models, locating the models to reuse, separating business rules from the process models, variation management, fragments, and process templates. Although the benefits of process model reuse have been theorised about in the literature, little empirical evidence exists to support these theoretical benefits (Fellmann et al., 2014). Furthermore, research also shows that approximately 10% of process modellers (PMs) reuse complete process models in practice (Fellmann et al., 2014; Koschmider et al., 2014), and it can therefore be argued that with such a small fraction of PMs reusing complete process models, it is not surprising that the expected benefits have not been seen. Recent literature confirms that the problem of reuse in business process modeling has not been adequately solved (Cui, 2017; Deng et al., 2017; François et al., 2023; François & Plattfaut, 2024; Khider et al., 2020; Polyvyanyy et al., 2020; Savary-Leblanc et al., 2024; Yongchareon et al., 2020)

The research presented in this thesis is titled “Increasing the level of reuse of complete process models in a process modelling repository”. The study focuses on developing a new modelling method which increases the level of reuse of process models in a process repository.

1.2 Research Questions and Objectives

The objective of this research project is to address the unresolved issue of complete process model reuse in a process repository and to design a new modelling method which increases the level of reuse of complete process models, i.e., to design a new IS artefact. Accordingly, the main research question is:

“***How can the level of complete process model reuse be increased?***”

While there is literature which explains why PMs do or do not reuse process models, and there is also literature expounding on the benefits of reuse, no research could be found which describes the consequences of not reusing process models and the impact on the business (Aldin & de Cesare, 2011; Fellmann et al., 2014; Koschmider et al., 2014). This resulted in the research question:

“***Why does a lack of process model reuse impact business costs?***”

This research question is exploratory and was answered by taking a system dynamics approach using published research and analysing interviews conducted with PMs in the research organisation.

To determine whether the proposed modelling method was successful, it would be necessary to quantitatively measure the level of model reuse in the repository. No quantitative measure of process model reuse was found in the literature, and this study developed the first measure in this regard because it would be useful to be able to quantitatively demonstrate an improvement. As will be elaborated later in this thesis, this is a quantitative measure which can be derived by extracting the appropriate data from a process repository and applying a formula. This gave rise to the research question:

“***How can the level of reuse of complete process models be measured?***”

This research question was answered using a DSR approach, which designed a new method (algorithm) to quantitatively measure the level of reuse of process models in a process repository.

1.3 Methods Used

“***How can the level of complete process model reuse be increased?***”

This, the main research question, was answered by adopting a pragmatic philosophy and using a DSR approach to design a new modelling method which increased the level of reuse of process models in a process repository. A pragmatic philosophy is appropriate in this context because we are designing an artifact which must result in action and useful consequences (Goldkuhl, 2012).

“***Why does a lack of process model reuse impact business costs?***”

This research question was answered by using deductive reasoning and qualitative analysis of interviews conducted with PMs. A causal-loop-diagram was constructed to illustrate the resulting system dynamics.

“***How can the level of reuse of complete process models be measured?***”

This question was also answered by adopting a pragmatic philosophy and a DSR approach to design a method to measure the level of reuse of process models. It should be noted that this DSR cycle formed part of the “Design” stage when answering the main research question. Again, a pragmatic philosophy was appropriate because we were designing a new IS artifact (a new method) to measure the reuse of process models.

1.4 Thesis Structure

This thesis is structured to present the results of a Design Science Research Project, as depicted in Figure 1.1.

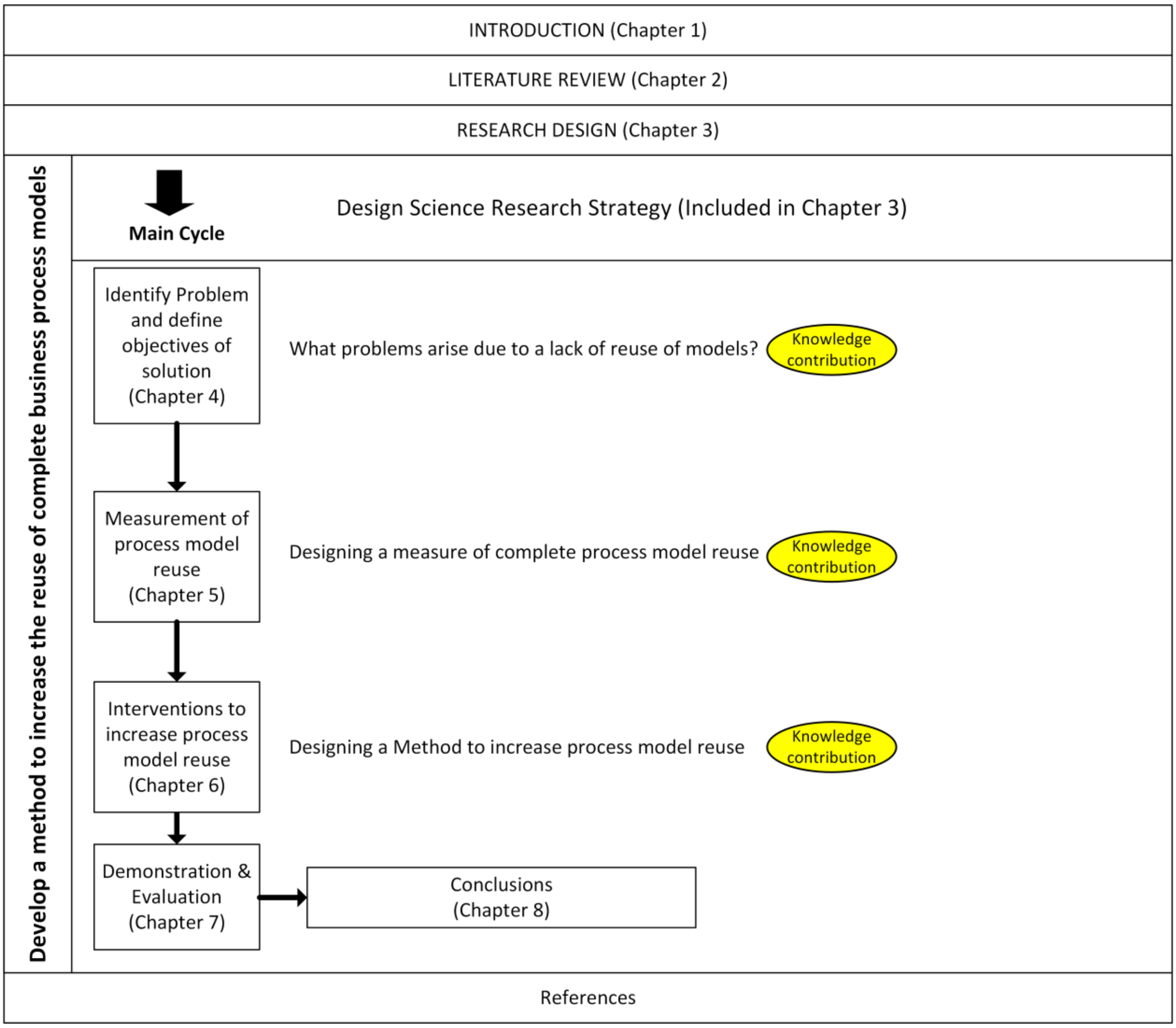


Figure 1.1: Thesis Structure

Chapter 1: Introduction: This chapter provides an overview of the research project, the problem statement, the research questions, the research method, and the structure of this thesis.

Chapter 2: Literature Review: This chapter provides an extensive review of the literature pertaining to process model reuse, software reuse, Business Architecture (BA), and Process Architecture (PA). Concepts such as capabilities and SOA, which are used in software reuse, are also reviewed. Theoretical models describing reuse in the process modelling domain are discussed.

Chapter 3: Research Method: This chapter explains how the research questions were answered using a pragmatic philosophy and a combination of qualitative and quantitative methods.

Chapter 4: Identify Problem and Define Objectives: This chapter describes the motivation for the research project by identifying the impact of not reusing process models. A causal-loop diagram is developed to describe the impact on business costs of not reusing process models.

Chapter 5: Measurement of process model reuse: In this chapter, a quantitative method to measure the level of reuse of process models is developed. This method is used in the demonstration stage

(Chapter 7) to quantitatively compare the level of reuse using both the “legacy” modelling method and the “new” modelling method.

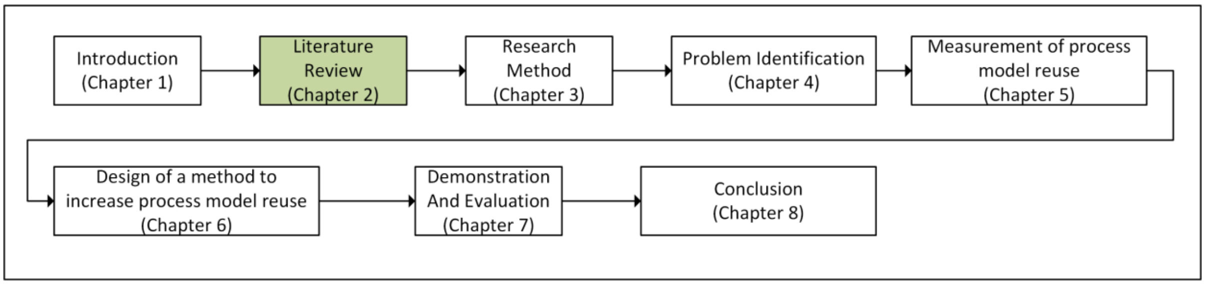
Chapter 6: Design of a method to increase process model reuse:

In this chapter, the elements of the new modelling method are justified and described individually. These are then integrated, and the full method is presented in an integrated manner.

Chapter 7: Demonstration and Evaluation: In this chapter, the improvement in the level of reuse of process models is demonstrated using the method described in Chapter 5. The new modelling method is evaluated by analysing interviews conducted with PMs.

Chapter 8: Conclusion: This chapter discusses the contributions of this research with respect to the three research questions which constitute the contribution of this research project. It considers the theoretical contribution, the practical contribution, and the limitations of this study. Finally, it offers some thoughts on potential future directions for research in the domain of business process modelling.

2 Chapter 2 Literature Review



In the first part of this literature review, the concept of reuse and its application in BPM are discussed. The theory of reuse and how to measure reuse are also discussed. The second part of this review relates to the domains of Enterprise Architecture (EA) and BA, which this research project draws on when developing the proposed artifact.

In order to identify previous research relating to process model reuse, an initial literature search was conducted in 2017 in four steps. 1) Searching Google Scholar using relevant search strings. 2) Identify and review relevant articles. 3) Review relevant articles which had cited those found in step 2. 4) Relevant references in any of the articles reviewed were also noted. The review was repeated in 2024 to identify any further relevant research which had been conducted since the initial search.

The initial search string used was "business process model reuse", "business process patterns", and appropriate permutations of these phrases.

2.1 Business Process Modelling

Process modelling tools are the software tools that are used to construct process models. These tools use two broad approaches. The process models are stored either as separate files (e.g., Microsoft Visio) or in a central database (process repository) that is accessible to all users. Storing each process model as a separate file is not a practical option for large process modelling endeavours, and most large organisations usually try to build a centralized process repository (Recker, 2006). Examples of process modelling tools in use are ARIS Business Architect and iGrafx. ERP systems (e.g., SAP, Oracle) and Business Process Management Suites (e.g., IBM, Software AG, Tibco Software) also include process modelling components (Hallerbach et al., 2010a; Recker, 2006).

A process modelling technique (also known as a language or notation) specifies the symbols, relationships and types of diagrams that will be used to represent the business processes. These languages were developed for different purposes, and therefore, the technique that is selected should be based on the purpose of the process modelling. Examples of common notations in use include UML, DFD, Petri Nets, Flow Charts, RAD, EPC, BPMN and BPEL (Nagm-Aldeen et al., 2015; Recker, 2006)

2.2 Reuse

The concept of reuse is familiar to all of us. We might have the recipe for our favourite cake so that we do not have to reinvent the recipe each time we wish to bake it. In that context, we have reused the recipe. It saves time and cost because the errors and wastage incurred when first developing the recipe are not repeated. This pattern repeats itself in many aspects of our lives, although the settings might vary.

Four types of model reuse were identified (Nolte et al., 2016):

- Initial Use: the original creation of the model.
- Revision: An update to the original model.
- Continued use: the sustained use of the model in support of one particular work task (e.g., continued use in a process improvement project).
- Repeated use: the repeated use of the model across multiple tasks (e.g., multiple projects).

A similar definition of reuse is “the post-creation usage of process models, thus using them for a different purpose, by a different user, or at a different point in time” (Nolte et al., 2013).

In the context of this study, we are interested in the “Repeated use” of a process model across multiple tasks and projects at different points in time.

2.3 Theories for Process Model Reuse

The factors driving the reuse of business process models have been investigated (Nolte et al., 2013). The factors influencing the “intention to reuse” of the PM were investigated as these were likely to translate into actual reuse. They wrote that “variance in an individual’s intention to reuse a process model is dependent on (a) factors describing properties of the process model considered for re-use, and (b) characteristics of the individual process model user. These relationships are being moderated by (c) organisational factors that determine the extent of social and normative pressure on re-use behaviour, and (d) attributes technologies that provide access to a model”. This model is depicted in Figure 2.1.

Process model factors refer to issues relating to the process model that will influence the intention to reuse. Such factors include the perceived accuracy (credibility) of the model as well as the ease with which the model can be understood. Individual factors relate to issues such as the PM's experience and motivation to reuse. Technological factors are those which could be provided by the modelling tool in use. While making the existing models easy to find and access does not guarantee reuse, difficulty in finding models will ensure minimal reuse. Organisational factors are those factors which impact how PMs operate within their area. If a superior is not seen to reuse models, reuse is less likely to be practised by the employee. If process model reuse is an accepted culture within the organisation, then a new PM is likely to follow suit.

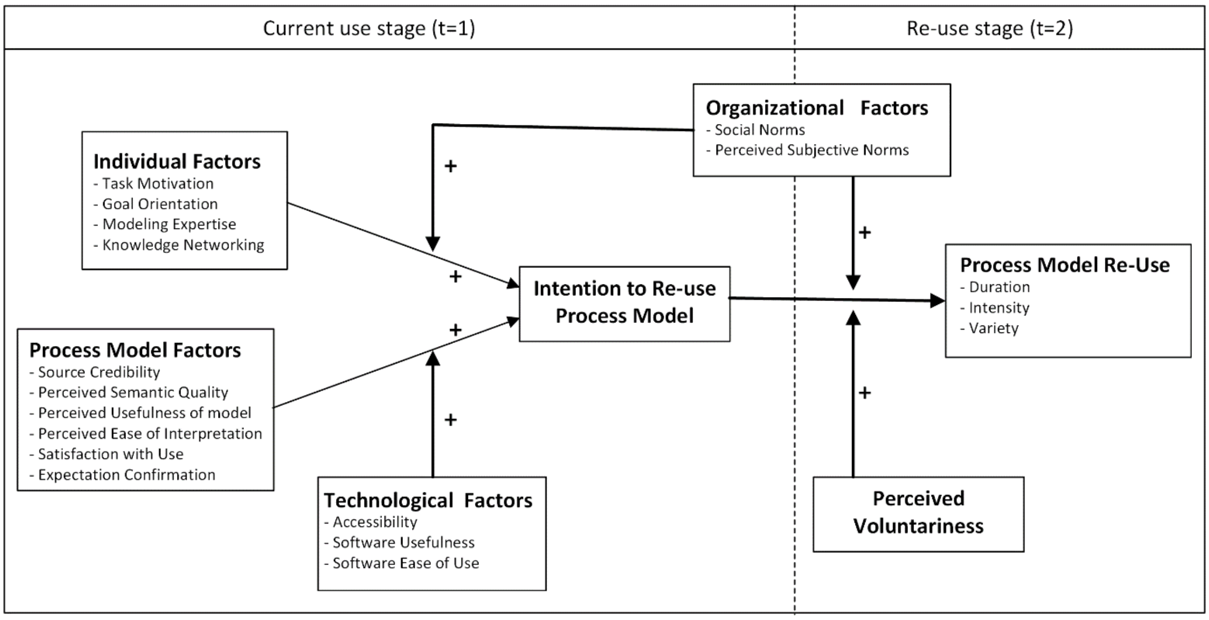


Figure 2.1: A Two-Stage Model of Process Model Re-Use (Nolte et al., 2013).

Based on this model, it is possible to identify the types of actions which would be expected to increase the level of reuse in an organisation. However, no practical guidance is offered to the PM on how to increase reuse.

The process of process model reuse has also been investigated (Erol, 2018). The model in Figure 2.2 can be summarized as “Either find and adapt an existing model or create a new model”. Again, no practical guidance is offered to the PM.

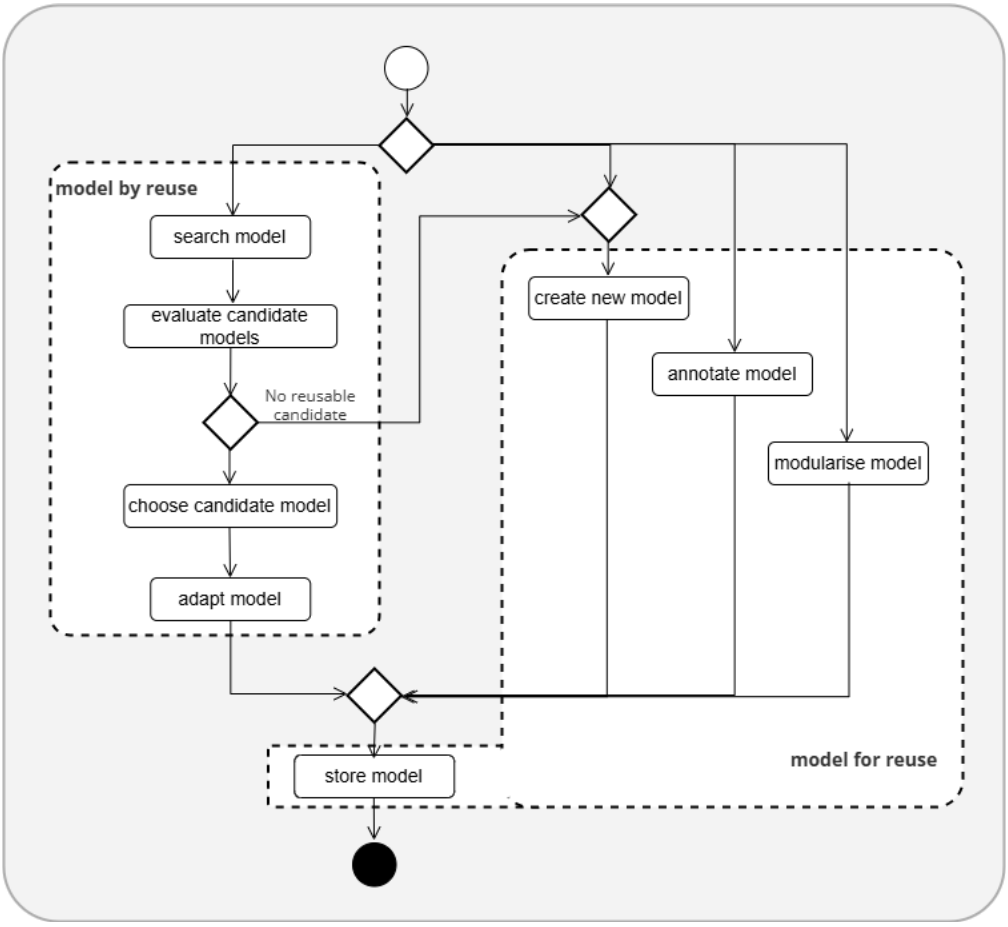


Figure 2.2: Process Model for business process model reuse (Erol, 2016)

These models of process reuse describe the factors influencing reuse and the steps that lead to reuse, and although they make intuitive sense, no practical advice or solutions are offered regarding what an organisation and PMs should do to increase the level of reuse. Some potential approaches were identified and are now described.

2.4 Approaches to process model reuse

The approaches identified in the literature to increase process model reuse can be categorized into seven high-level approaches, which are now described:

- Identification of similar models
- Locating the models to reuse
- Separating Business Rules from the process models
- Variation Management
- Fragments
- Process Templates
- Recommendation Systems

2.4.1 Identification of similar models

Those organisations that have modelled processes for a long time will already have a repository containing possibly thousands of models with such models frequently sharing redundant logic (Dumas, García-Bañuelos, & Dijkman, 2009). In this case, the challenge becomes one of identifying similar (or identical) process models and then rationalizing such models.

In a survey of process similarity measures (Becker & Laue, 2011), eight properties that a similarity measure should have (e.g. it should be computationally quick to calculate) were proposed, and twenty-three process similarity measures that had been reported in the literature were assessed and compared. This was done by starting with a base model and then applying different changes to the base model to create variants of the base model. The base model and these variants were used in the calculation of the similarity measures. It was shown that different similarity measures might rank the similarity of the processes differently and that no similarity measure fulfilled all eight desirable characteristics that were originally identified. Therefore, the similarity measure selected should be based on the use case.

These similarity measures offer an opportunity to identify similar/identical models in the repository, and this information can serve as a basis for removing redundant process models from the repository. In the theoretical model of reuse, this is a “Technological Factor” as it improves the “Accessibility” of the remaining process models in the repository.

2.4.2 Locating the models to reuse

The ease with which a PM is able to find the appropriate process model in a repository is an issue in process modelling. Business process models are usually modelled in a hierarchy, whether this hierarchy is built from the top-down or from the bottom-up. Navigating process models through a hierarchy can be difficult, and it is easy for the PM to lose sight of exactly where he / she is both in the hierarchy and in the actual process (Figl, Koschmider, & Kriglstein, et al., 2013); François & Plattfaut, 2024; Polyvyanyy et al., 2020). These difficulties will adversely impact the “Intention to reuse process model” parameter (through “Accessibility” in the “Technological Factors”) in the theoretical model, which is directly linked to the actual level of reuse of the process model.

Managing, finding, and reusing process models has also been investigated (Dijkman, Rosa, & Reijers, 2012). These cover areas of interest, such as identifying similar models, making recommendations to PMs regarding which process to use, and modelling techniques based on templates.

Three common approaches used to represent hierarchies are node-link, treemap and nested graph. The preferred method based on a survey of 14 experts was the node-link approach (Figl et al., 2013). An example of the node-link representation is shown in Figure 2.3.

This view of the process model hierarchy enables a PM to navigate through the hierarchy more quickly than would be the case having to open each process model and then navigate from that model down to the next level in the hierarchy. Building a node-link view of the process models in the process repository would assist PMs in finding the correct process model. However, none of the process modelling tools evaluated was able to create this view of the process models automatically (Figl et al., 2013).

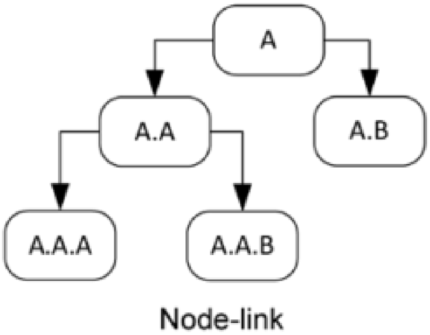


Figure 2.3: Node-link example (Figl et al., 2013)

2.4.3 Separating Business Rules from the process models

One of the reasons an organisation has many variations of a process is that business rules have been modelled in the process logic itself (Van Eijndhoven, Iacob, & Ponisio, 2008). The problem that arises is when another area in the organisation has a slightly different version of the business rule. It then becomes necessary to make a “copy” of the process and update the business rule at the appropriate place in the “new” process variant.

2.4.4 Variation Management

One approach is where the model is treated as an asset, and an “asset tree” is used as a mechanism to store the model versions and their variants (Narendra et al., 2012). The “Asset Capability and Analysis Model” (ACAM) is used to specify the capabilities of the models, and the “Asset Requirements and Constraints Model” (ARCM) is used to define the requirements. Algorithms that matched the capabilities to the requirements and the degree of match that was found were proposed (Narendra et al., 2012).

2.4.5 Fragments

Another line of research relates to identifying recurring fragments of a process model and the reuse of such fragments (Baumann et al., 2023; Belhajjame Khalid and Grigori, 2017; Deng et al., 2017; Eshuis, 2023; Goldstein & Gonzalez-Alvarez, 2021; (Markovic & Pereira, 2007; Smirnov et al., 2012). A fragment is a standalone piece of a business process that occurs frequently in the process repository. Fragments could be used to autocomplete a new process model, or they could be substituted into a placeholder in a process template.

2.4.6 The Use of Process Templates

The use of templates is a common practice in organisations to ensure a standard structure and "look and feel" for documents. They also ensure that important sections that should be in the document are not accidentally omitted. Similarly, process templates offer another approach that can be used to increase the reuse of process models. Process templates can be used in business process modelling in two fundamentally different manners. Firstly, the process template can be used to generate a new process in the repository based on the template (Cui, 2017; (Kumar & Yao, 2012), or the template can be used with configuration data which defines the instantiation of a process using the template (Eshuis, 2023; (Tran, Coulette, & Vu, et al., 2011). However, although using the template to generate a new process in the repository ensures consistency among the processes that were generated using the same template, it still results in additional process models in the repository, which require maintenance. Changes required to one of the models that were based on the template may also have to be made manually to the other models that were also based on the same template. Worse still, a change made to a model based on the template but not needed in the other models will mean that the updated model is now inconsistent with the original template that was used. Secondly, Reference models (e.g. SAP R3 and ITIL) are another type of template that can be used. Formal approaches to customizing a reference model to ensure the integrity of the result have been proposed (Soffer et al., 2007); Yongchareon et al., 2020). Nevertheless, reference models must be used with care as the reference model may not be applicable to a given situation (Recker, 2006).

2.4.7 Recommendation Systems

Streaming services such as Spotify and Netflix are based on recommendation algorithms where the system recommends items which might be of interest to the user. This has increased research into the use of recommender systems in BPM where the system identifies one or more models which might be of interest and presents them to the modeller. The modeller is then able to select one of the recommended models or create a new model if none of the recommended models are suitable. Recommendation systems can be Content-based, Collaboration-based, Knowledge-based, Context-based, Socially-based, Demographically-based, or Hybrid-based (Almonte et al., 2021, 2022; Deng et al., 2017; Goldstein & Gonzalez-Alvarez, 2021; Khider et al., 2020; Sola et al., 2021; Sola & Stuckenschmidt, 2021; Trabelsi et al., 2021).

2.4.8 Summary

In this section different approaches to process reuse were discussed. However, the problem of reuse has not been adequately solved as calls for improved support of reuse in process modelling continue to be made (François et al., 2023; François & Plattfaut, 2024; Polyvyanyy et al., 2020; Savary-Leblanc et al., 2024).

2.5 Software Development and Business Process Modelling (BPM)

Diagrams are used to represent conceptual solutions to problems, and this is particularly prevalent in the fields of software development and BPM (Ferrucci et al., 2002). Visual programming languages (VPLs) have been developed, which take a graphical representation of program logic and can directly generate the corresponding textual version, which can be compiled into an application and executed (Ferrucci et al., 2002). BPM notations such as EPC and BPMN are commonly used to represent business process control-flow logic, while Business Process Execution Language (BPEL), which can be converted directly into an executable application by many workflow systems, is essentially BPMN with additional constraints enforced (Ouyang et al., 2008). Accordingly, business process models can be considered as a programming abstraction for business users where the commands are not executable but syntactical in nature (Fantinato et al., 2012; Zhao et al., 2006).

2.6 Measurement of Process Model Reuse

Complicating enterprise process modelling is the proliferation of new channels. Organizations are now having to design similar processes covering multiple channels (consider bricks and mortar, call centres, the internet, mobile devices, email, social media, instant messaging, and more) while ensuring consistency, as customers are expecting seamless, improved, and consistent experiences when dealing with the organisation (Seck & Philippe, 2011). In practice, these organizations will typically document their business processes in a process repository. Organizations wish to reuse these models where possible to improve the economic value of these models, and the question of how to achieve and measure process model reuse then arises. A possible measure of process model reuse is the amount of reuse of models in a repository by other process models in the same repository. Measuring the level of process model reuse by other models in the process repository would be an important indicator of model reuse. For example, it enables the level of process model reuse to be measured directly and automated, the modelling practices within an organization to be managed, and even the direct comparison of modelling methods regarding the reuse of process models. While process model reuse is a frequent topic of research, no research could be found related to measuring the reuse of models within a process repository by other models within the repository.

The value of BPM increases with process model reuse. Previous research into process model reuse has focused on behavioural aspects of reuse, such as the intention to reuse, the repeated reuse of a process model over time, and the identification of elements of process models which could be reused. However, process model reuse can also be considered from the perspective of the reuse of process models by other process models in the same repository. Such a measure would be a direct measure of whether PMs are creating bespoke versions of existing process models or are indeed reusing existing process models and reaping some of the purported benefits of reuse. Furthermore, it would provide a measure of reuse which can be automated. Organizations that operate in a multi-channel, multi-product environment have business processes that frequently share functionality (consider authentication, for example) and that may be used in different organizational units.

2.6.1 Process Model Reuse

Much research has been conducted into aspects of the reuse of process models: reuse of process fragments, similarity searches, reference models, management of process variants, and patterns. Several literature surveys relating to reuse in business process modelling have also been published (Fantinato et al., 2012; Koschmider et al., 2014).

Typical benefits of process model reuse are claimed to be a reduction in modelling time, an increase in productivity, leveraging of existing resources and skills, and an increase in the quality of the process models (Deng et al., 2017; (Fellmann et al., 2014; François & Plattfaut, 2023; Khider et al., 2020; Koschmider et al., 2014). The use of the word “claimed” is deliberate because although these benefits are stated in most studies about the reuse of process models, validation of these claims is almost completely missing (Fellmann et al., 2014). One research study showed that only 10.5% of respondents reused complete process models, 55.1% reused individual process elements and the search for reusable model assets was conducted manually by 40.5% of respondents (Koschmider et al., 2014). It could be argued that the positive effects of process model reuse cannot be realized because process model reuse is not being practised sufficiently (Koschmider et al., 2014). Related to this dearth of empirical data on process model reuse by other models is the question of how process model reuse within a repository should be measured.

It was also surprising to discover that although measuring software reuse quantitatively was well established (Poulin, 1997), no corresponding metrics for process model reuse could be found. Without a quantitative method of measuring process model reuse, it would not be possible to compare modelling methods quantitatively regarding the reuse of process models. Accordingly, we identified a need for a quantitative measure of the level of reuse of process models in a process repository. We will refer to this as the “Level of Reuse” (LOR). Software reuse, which has been extensively researched, provides the foundation for the development of a method to measure the reuse of process models.

2.6.2 Software Reuse

The expected benefits of software reuse are well documented: improvements in productivity and quality, savings during the maintenance phase due to fewer errors (Poulin, 1997), quicker time to market for the product (Frakes et al., 1996; Poulin, 1997) and improved interoperability of components (Frakes et al., 1996; Poulin, 1997). A high correlation between reuse rate, reduced development time, and a decrease in the number of errors has also been identified (Frakes et al., 1996). However, reuse does not come for free (Fenton & Bieman, 2015), and for software reuse to make economic sense, the quantified benefits of reuse need to exceed the cost of developing for reuse. While this has resulted in various economic models which can be used to quantify the benefits of reuse (Frakes et al., 1996; Poulin, 1997), all economic models depend on some measure of reuse as an input to the model. Measurement models for software reuse are commonly based on counting either lines of code (excluding comment lines), data items (e.g., business objects) or Function points (Daneva, 1999).

Reuse Level. The de facto metric for measuring software reuse is the reuse % or level of reuse (Poulin, 1997), which measures how much of the product can be attributed to reuse (Frakes et al., 1996). Their measurement of software reuse is based on the assumption that the software is composed of items at different levels of abstraction. An item could be a module (.c file), which, in turn, is composed of functions that contain lines of code. A lower-level item is used by a higher-level item. The equations of Terry (Terry, 1993) for software reuse can be written as:

$$\text{External Reuse Level} = EU / T \qquad \text{(Eq. 1)}$$

$$\text{Internal Reuse Level} = IU / T \qquad \text{(Eq. 2)}$$

$$\text{Total Reuse Level} = \text{Internal Reuse Level} + \text{External Reuse Level} = (IU + EU) / T \qquad \text{(Eq. 3)}$$

Where:

EU = number of external lower-level items that are used more than ETL

IU = number of internal lower-level items that are used more than ITL

T = total number of lower-level items in the higher-level item, both internal and external

ITL = internal threshold level, the minimum number of times an internal item must be used before it is counted as reused

ETL = external threshold level, the minimum number of times an external item must be used before it is counted as reused

This researcher agrees with Eqn. (1), Eqn. (2), and Eqn. (3). The inclusion of ITL and ETL means that the equations provide for variations of when an item is counted as reused. There may be instances when an item must be reused five times before it is counted as reused. In this situation, that item is still only a single item and counted once. Therefore, it is important to recognize that these counts are of unique items and not references. In other words, an item is only counted once, even if it is reused ten times. In the context of this research project, an item is a process model. This research project will count a model as reused if it is used two or more times (ITL=2). Then to determine the reuse level, one must count the number of items reused (IU) divided by the total number of items (T). This research is dealing with the models in a repository which are internal to the repository. Accordingly, Eqn. (1) will be ignored, and then Eqn. (2) becomes the same as Eqn. (3). This discussion is further elaborated in Chapter 5 where the application of these equations to a process model repository is described.

2.7 Process Modelling and Reuse Challenges

While BPM is the foundation of process improvement projects, and many large enterprises have embarked on extensive process modelling projects, questions have been raised about whether the purported benefits are realised in practice. The top 10 business process modelling issues identified in a study are shown in Table 2.1 (Indulska, Recker, et al., 2009).

Table 2.1: Overall top 10 business process modelling issues (Indulska, Recker, et al., 2009)

Rank	Issue	Description
1	Standardization	Issues related to the standardization of modelling notations, tools, and methodologies.
2	Value of process modelling	Issues related to the value proposition of process modelling to the business.
3	Model-driven process execution	Issues related to the model-driven development of executable process code and the life-cycle of process modelling to execution.
4	Model management	Issues related to the management of process models such as publication, version, variant or release management.
5	Modelling level of detail	Issues related to the definition, identification, or modelling of adequate levels of process abstraction.
6	Methodology	Issues related to the process of process modelling.
7	Governance	Issues related to the governance of process modelling efforts on projects.
8	Buy-in	Issues related to the acquisition or ongoing assurance of buy-in and commitment from process modelling sponsors.
9	Business-IT divide	Issues related to the use of process modelling in IT versus business scenarios, application areas, or communities.
10	Process orientation	Issues related to the development or education of a process-aware perspective in relevant stakeholders or organizational units.

Many of the issues listed in Table 2.1 are related. Addressing one (or not) improves (or worsens) another. For example, a lack of agreement within an enterprise regarding “Modelling level of detail” (Issue #5) could lead to two models being created for the same process with different levels of detail. This, in turn, will aggravate “Model management” (Issue #4), which is likely to negatively impact “Value of process modelling” (Issue #2). This is just one demonstration of the linkages between the issues listed in Table 2.1. In Section 2.4 various approaches to process model reuse were described. In spite of the continued research into process model reuse, one of the challenges faced by modellers is that process modeling tools and repositories do not provide adequate support for process model reuse (Almonte et al., 2021; Costa et al., 2020; Cui, 2017; François et al., 2023; Khider et al., 2020; Makni et al., 2018; Zhu et al., 2024).

The most frequently mentioned positive benefit of process model reuse is the reduction in time required to construct a process model (Fellmann et al., 2014) and in this manner, process model reuse increases the value of process modelling (by reducing the costs), which is identified as the second most important process modelling issue in Table 2.1.

2.8 Multi-channel, multi-product environments

As technology evolves, enterprises are expected to offer their products and services through an ever-increasing number of channels. The practicalities of this are further complicated by the number of products and services that need to be offered. Taking a bank, for example, multiple products and services (current accounts, savings accounts, etc.) must be offered through a number of channels (physical branches, call centres, internet banking, mobile applications, etc.) to multiple client types (business client, retail client etc.). A European bank identified fourteen channels and customer touchpoints (Menrad, 2020).The internal structure of the organisation also complicates matters as the responsibility for the product design, technical solution, and the operational servicing of the customer is usually the responsibility of different parts of the organisation (Beerepoot et al., 2023). However, the processes executed in each area will likely share pieces of common logic. Depending on the process modelling approach, this shared logic may be modelled separately in each area (Branco, Xiong, Czarnecki, Kuster, & Volzer, 2014). Hallerbach et al. (2010) described a similar situation relating to car components where more than 20 variations of the same process were found based on product, supplier and the development phase of the component. This is depicted in Figure 2.4, where each column corresponds to an end-to-end process to open a current account for a client in two different channels. At this stage of the discussion, each channel is being treated separately. In a situation where multiple channels are available, Figure 2.4 shows that the initial part of such processes is channel-specific. For example, the initial part of the process is different when selling the account through a branch or a call centre; then, there is a portion of the process that is common to all scenarios, no matter which channel it was initiated on. Finally, there may be some administrative processes that are common to the product but may differ depending on whether the client is a business client or a retail client.

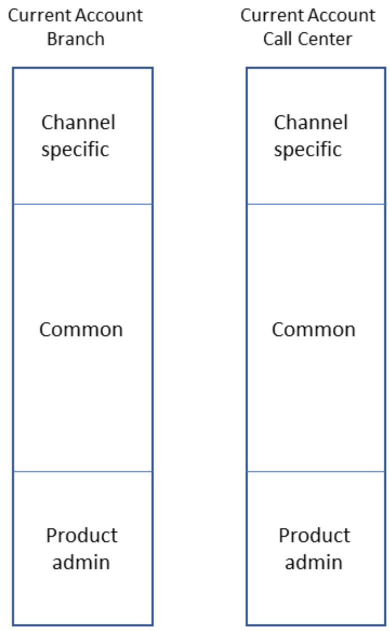


Figure 2.4: Commonalities and differences of a current account sale across channels

A client may wish to start in one channel and then switch to a different channel (e.g. possibly call the call centre) (Lockie, 2014; Verhoef, Kannan, & Inman, 2015). Due to the organisational structure issues referred to earlier, modelling this process flow becomes problematic because of the number of permutations that emerge. If there are four steps in a process and two possible channels for each step, then there are eight possible permutations of process flow available. The permutations become even worse when there are four or five channels in use. This will be referred to as the multi-channel / multi-product dilemma in this paper.

While the benefits (Görsch, 2002; Herhausen, Binder, Schoegel, & Herrmann, 2015; Lewis, Whysall, & Foster, 2014), integration requirements (Görsch, 2002; Lewis et al., 2014; Seck, 2013), and challenges (Lewis et al., 2014; Seck, 2013) of channel integration have been researched in the literature, no articles were found that addressed the multi-channel multi-product dilemma specifically. The mapping of business processes in a multi-channel environment is often carried out by different employees, in different parts of the organisation, for different projects, and over an extended period of time (Branco et al., 2014). As the number of process models in the repository increases over time, new issues begin to appear (Hallerbach et al., 2010). Multiple versions of the same model, similar logic appearing in multiple models (Branco et al., 2014; Cuesta, Ruesta, Tuesta, & Urbiola, 2015), difficulties in locating the correct version of a process model, and conflicting versions of a process model (Branco et al., 2014) are some of the issues that have been documented in the literature (Alotaibi, 2016; Hallerbach et al., 2010; Jonnavithula, Antunes, & Cranefield, 2015; Kumar & Yao, 2012; Reijers, Mans, & van der Toorn, 2009; Smirnov, Weidlich, Mendling, & Weske, 2012). One study found that only 10.2% of respondents reused complete process models

(Koschmider et al., 2014), and this could be laying the foundation for future model and repository management challenges.

Process model collections are frequently stored in process repositories. As enterprises embark on process modelling, and the number of process models increases, the size of the repository grows, and the management of the model collection becomes more important. Model management has been listed as the 4th most important issue in business process modelling (Indulska, Recker, et al., 2009).

2.9 Enterprise Architecture

The similarities between BPM and software development have been reported in the literature (Fantinato et al., 2012; Zhao et al., 2006). Using the concept of EA, this section explores Application Architecture and BA, and notes potential similarities across these two domains. In particular, this research project draws on the concept of a service-oriented-architecture (SOA), which is located in the Application Architecture sub-domain in the EA. A high-level view of an EA and relevant domains is shown in Figure 2.5.

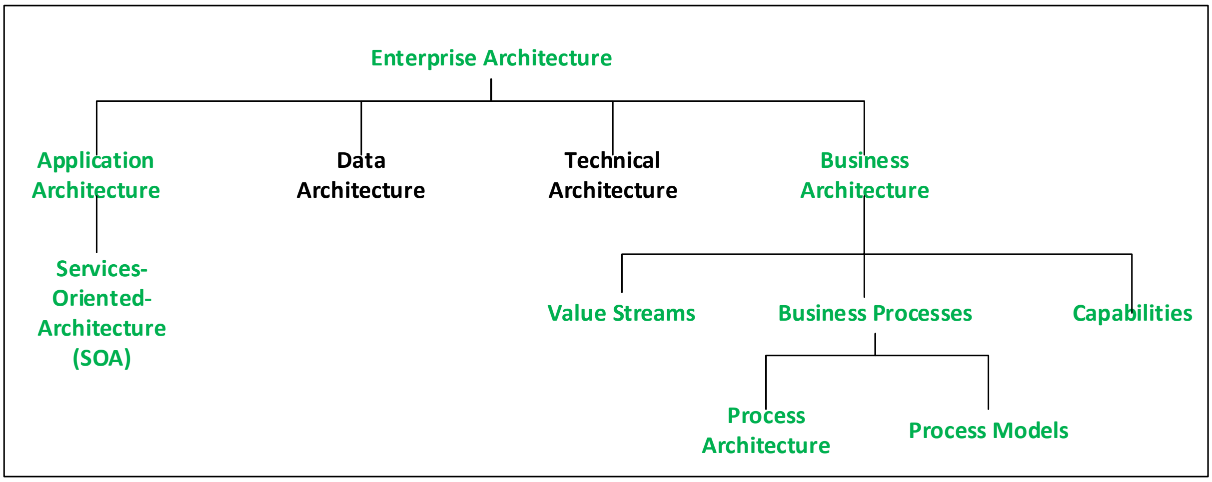


Figure 2.5: Enterprise Architecture: Positioning of Concepts and Tools

Successful enterprises consist of people, systems, suppliers, customers, and processes, all of which need to be coordinated to achieve success. A formal understanding of these elements and the relationships between them facilitates the alignment of the enterprise activities with the enterprise strategy (Whittle & Myrick, 2016). Accordingly, EA can be defined as “the holistic planning, analysis, design, and implementation for the development and execution of strategy by applying principles and practices to guide organizations through the integration and interoperation of all other architecture domains” (The Federation of Enterprise Architecture Professional Organizations (FEAPO), 2017). Within EA, FEAPO recognises four further sub-domains: 1) Application Architecture,

2) Business Architecture, 3) Data Architecture, and 4) Technical Architecture (The Federation of Enterprise Architecture Professional Organizations, 2017):

2.10 Business Architecture

To increase the level of reuse of process models, it is obvious that one of the key activities of a PM will be to locate the existing process model which is to be reused. The manner in which the process models are arranged in the process repository is therefore important, and in this section, we position the role of the BA and the PA.

The Federation of Enterprise Architecture Professional Organizations (FEAPO) Taxonomy Working Group (2017) defines BA as a representation of "*holistic, multidimensional business views of: capabilities, end-to-end value delivery, information, and organizational structure; and the relationships among these business views and strategies, products, policies, initiatives, and stakeholders*". The formalised structure and storage of the process models is an important aspect of any process modelling method as it facilitates an end-to-end view of longer processes which may have been decomposed into a series of shorter sub-processes (zur Muehlen et al., 2010). It also assists other users in locating these processes. From an EA perspective, the Business Architecture Guild (BAG) positions process modelling under the BA domain, and specifically under the Value Stream sub-domain, as depicted in Figure 2.6. Within the BA, further sub-domains are also recognised, as depicted in Figure 2.6.

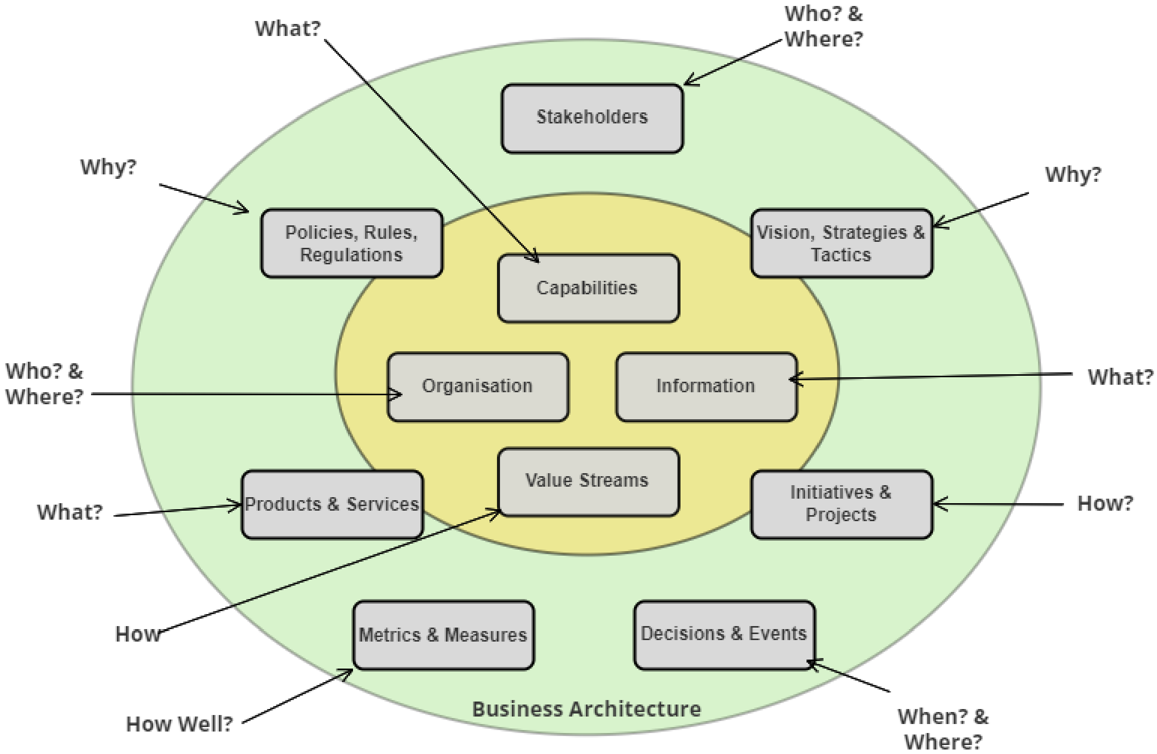


Figure 2.6: Business Architecture sub-domains (Business Architecture Guild, 2024)

The BA provides an abstract representation of the business and is used as:

- A communication tool
- An analytical tool
- A tool to navigate complexity, reduce risk, and align stakeholders within the business

The sub-domains of Value Streams, Process Architecture, and Capability Mapping provide the definitions and principles used to develop a process modeling method by providing guidance regarding end-to-end processes and the decomposition of such processes into sub-processes and components.

2.10.1 Value Streams

The Business Architecture Guild (2024) defines a Value Proposition as "An innovation, service, or feature intended to make a company, product, or service attractive to customers or related stakeholders" (p. 145). This value proposition is usually realised through a series of activities which lead to the desired value proposition. These intermediate activities are known as "Value Items". A Value Item is defined as "The judgement of worth, made by an individual or an organisation, attached to something tangible or intangible and attained in the course of a particular interaction with one or more parties" (Business Architecture Guild, 2024, p. 144). A Value Stream is a sequencing of Value Items to realise a desired Value Proposition as illustrated in Figure 2.7 (Business Architecture Guild, 2024).

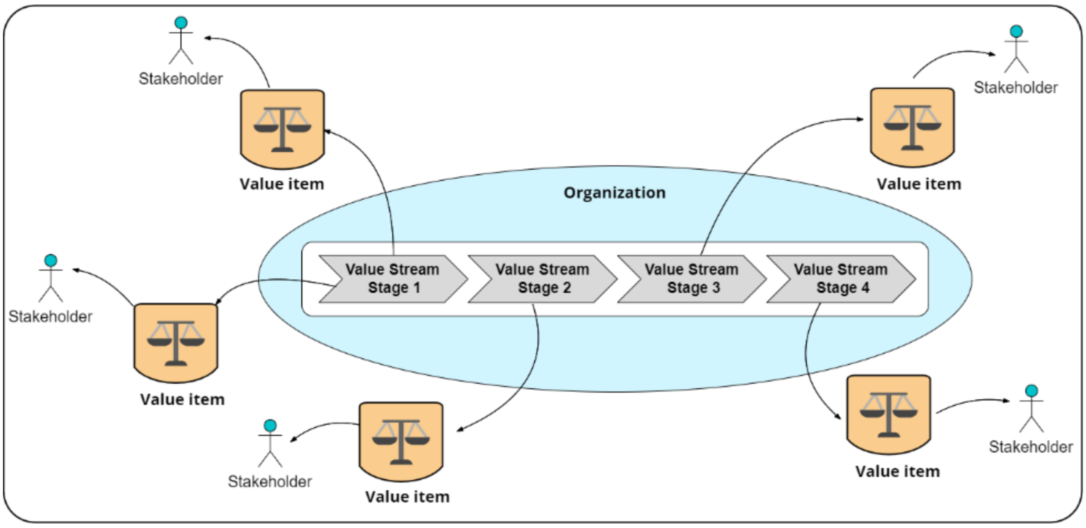


Figure 2.7: Stakeholder, Value Item, and Value Stream Stage Relationships (Business Architecture Guild, 2024)

The BAG differentiates between a Value Stream and a Business Process. The Value Stream reflects the business process but only contains the activities that deliver value from a customer's perspective. Activities relating to regulatory reporting, for example, would not appear in a value-stream as the customer will not view such activities to be of value. However, from a business process perspective, regulatory reporting is required and should be reflected in the business processes. Figure 2.8 illustrates the relationships between these elements.

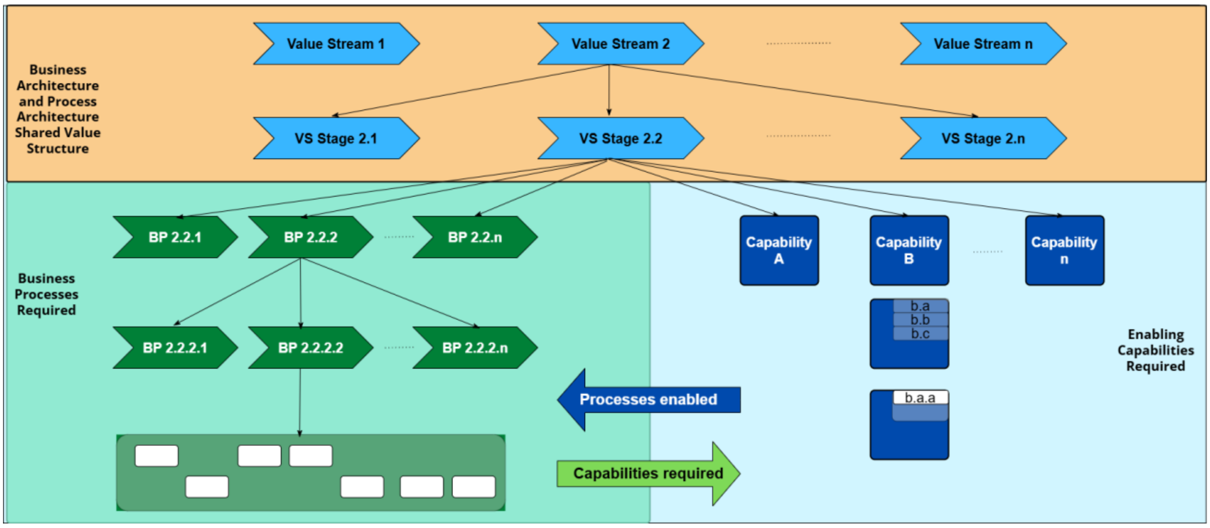


Figure 2.8: Illustration of Value Stream, Stages, and Capability (Burlton et al., 2019)

2.10.2 Process Architectures

The process models created in an organisation, either as part of a modelling initiative or during requirements definition, need to be stored in a systematic manner. The PA is the framework according to which these models will be stored, in the same way that the Dewey classification system is used for books in a library. Two categories of PAs can be identified (Malinova et al., 2013).

Decompositional PA: In a decompositional PA, the activities are decomposed into more granular sub-processes. The activities in the upper levels of these architectures are the most general but become more granular with each level of decomposition. Figure 2.9 and Figure 2.10 illustrate the concept of a decompositional PA.

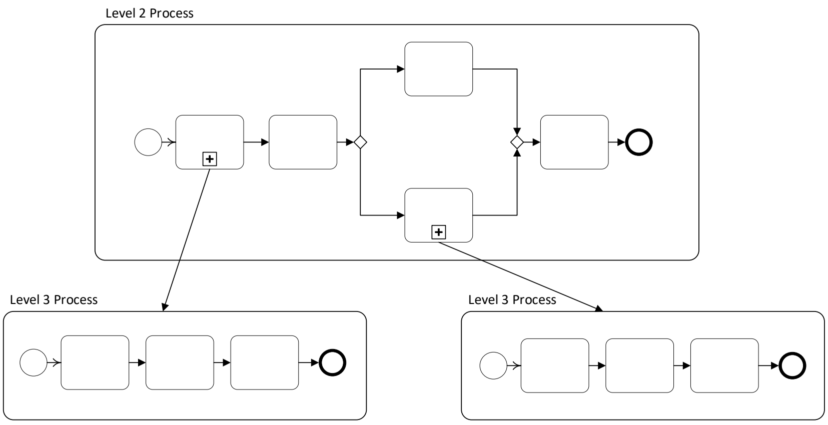


Figure 2.9: Decompositional Process Architecture (Malinova et al., 2013)

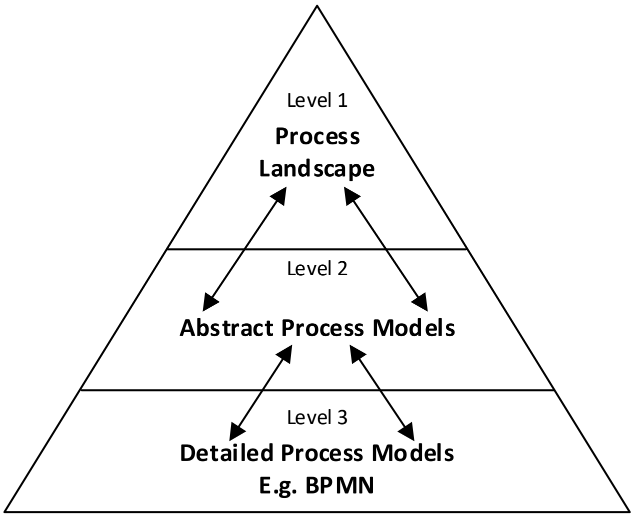


Figure 2.10: Levels of detail in a Process Architecture (Dumas et al., 2013)

While some companies use names to identify the different levels, simply numbering the levels is also a common practice. A restriction of this type of architecture is that a fine-grained process always has only one more coarsely-grained process associated with it (Malinova et al., 2013).

Service-oriented PA: In a service-oriented architecture, the processes are categorised into groups with a focus on process reuse across categories. In this approach, a fine-grained process can be associated with multiple other processes, regardless of the process level. However, it is still possible for each process in a category to consist of a hierarchal decomposition. Figure 2.11 illustrates the concept of a service-oriented PA.

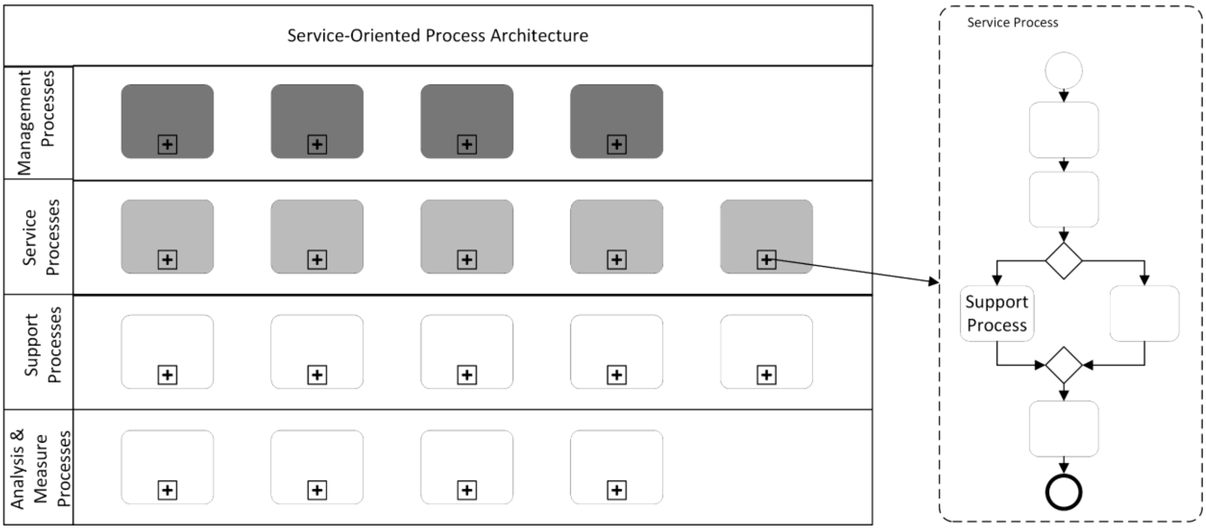


Figure 2.11: Service-Oriented Architecture (Malinova et al., 2013)

2.10.3 Capability Mapping

Successful organisations seek competitive advantages to maximise profits and stay in business. While Prahalad and Hamel argued that the source of competitive advantage of an organisation lies in the management's ability to identify its core competencies, Javidan viewed competencies as being the integration and coordination of capabilities (Javidan, 1998; Prahalad & Hamel, 2003). In a similar vein, Homann viewed the business as a network of capabilities (Homann, 2006). A capability defines "what the business does" as opposed to "how the business does it". Capability mapping is an analytical technique used for strategic decision-making by modularising processes into services and then determining which services can be outsourced (Beimborn et al., 2005). Furthermore, by associating processes with capabilities, different processes supporting the same capability can be identified, reducing redundancy where appropriate. While a capability map exposes all capabilities within the business, it does not necessarily fit with the org chart of the enterprise as it is a model of "what is done" – not who does it or how it is done. It has been found that the capabilities in all enterprises can be abstracted into five high-level capabilities consisting of 1) Product development, 2) Client interaction, 3) Fulfilment, 4) Plan and manage enterprise, and 5) Collaboration management. Capabilities are also hierarchical and can be decomposed, as shown in Table 2.2. Sales is a 2^{nd}-level capability under the 1^{st}-level capability "Client interaction", and "Advertising" is a 3^{rd}-level capability under the 2^{nd}-level capability "Sales". These capabilities can be decomposed further – there is no maximum number of levels permitted.

Table 2.2: Capability decomposition

Level	Capability
1	Client Interaction
2	Sales
3.	Advertising

2.10.4 Summary

In this section key elements of the Business Architecture have been reviewed. The relationships between Value Streams, processes in the Process Architecture, and Capabilities have been explained. While Value Streams can be viewed through two different lenses: business processes, and capabilities, we see that Processes are enabled by Capabilities while Capabilities require processes in order to be realised.

2.11 Application Architecture

The Application Architecture relates to the structure of, and relationships between the applications in an enterprise, and without deliberate planning, it is unlikely that all the applications in an enterprise will perform efficiently. The Federation of Enterprise Architecture Professional Organizations (FEAPO) Taxonomy Working Group (2017) defines the Application Architecture as:

“Application Architecture represents the specification and structural partitioning of technology-based automation into business logic, user experience, and data perspectives as an enabler of BA and strategy.”

The Web Services Architecture Working Group. (2004) defines software architecture as:

“The software architecture of a program or computing system is the structure or structures of the system. This structure includes software components, the externally visible properties of those components, the relationships among them, and the constraints on their use”.

In this section, the concept of a Service Oriented Architecture and some of the underlying principles drawn from the Application Architecture domain are introduced.

2.11.1 Service Oriented Architecture

A “service” is a well-understood concept frequently used by both individuals and organisations in society to refer to one or more tasks that are performed by the “service provider” for the “service consumer”. Governments provide services such as “policing” to their citizens (consumers), while access to the Internet is usually provided by an “Internet Service Provider” (ISP). Furthermore, services can be decomposed into finer-grained services. “Policing” can be decomposed into “Investigative Services” and “Protection Services”, which can, in turn, be further decomposed. Services such as “Document certification” and “New email address registration” could be examples of fine-grained services in the previous examples.

In application development, the concept of services emerged due to challenges associated with traditional software solutions, which were focused on automating a specific business process or processes (Erl, 2008). This approach, while building the specific solution efficiently, results in other issues such as:

- significant amounts of redundant logic
- continually rebuilding existing logic
- complex infrastructures and convoluted EA
- integration challenges because making these applications share data at a later point is difficult.

In application development, service orientation is a design paradigm where the fundamental unit of solution-oriented logic is the “service” (Erl, 2008). Key concepts in service orientation are listed in Table 2.3

Table 2.3: Key SOA Concepts

Concept	Definition
Service	1. An abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities" (Web Services Architecture Working Group.,(Haas & Brown, 2004) 2. "A physically independent program comprised of a set of capabilities. These capabilities can be invoked by external consumer programs" (Erl, 2008).
Service-Oriented-Architecture (SOA)	*A set of components which can be invoked and whose interface descriptions can be published and discovered* (Web Services Architecture Working Group.,(Haas & Brown, 2004).

Developing a solution to fulfil a business requirement entails identifying and building the required services and then assembling these services in a manner that meets the business requirement. The key element is that these services can then be reused in other solutions without having to duplicate existing logic. This concept is illustrated in Figure 2.12.

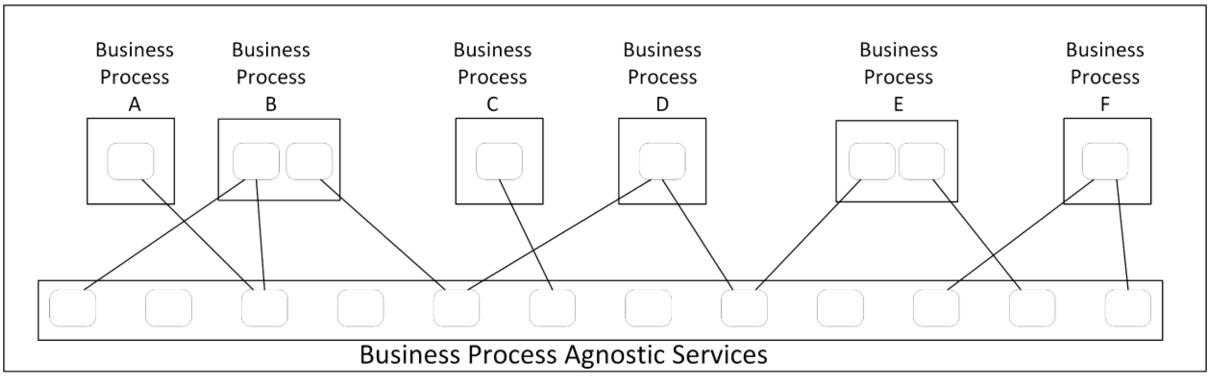


Figure 2.12: Reuse of services in different solutions (Erl, 2008)

The reuse of well-designed services results in the following benefits:

- Increased amounts of agnostic logic (logic independent of the language and platform on which it resides)
- Reduced amount of application logic
- Reduced volume of overall logic
- Increased agility
- Reduced development and maintenance costs

2.11.2 SOA Service Design Principles

Design principles have been historically encouraged in the IT world because they offer guidelines to ensure that when something is done, it is done in a consistent manner. As the knowledge and experience within a domain improves, principles will be adjusted and will frequently coalesce into a set of "Best Practise Principles". In SOA service design, eight key principles, listed in Table 2.4, have been identified (Erl, 2008).

No.	Principle	Summary
1	Services share standardized contracts.	A service contract describes the purpose and capabilities of a service. This is the information that is exposed to external consumers and includes a technical interface or one or more description documents. Different contract standards can exist, but services within the same service inventory are in compliance with the same contract design standards.
2	Services are loosely coupled.	Coupling can be thought of in terms of "dependency". Tightly coupled services cannot execute independently of each other, whereas loosely coupled services are able to execute independently of each other. Typically, if a service consumer is specifically designed to interact with a specific service, then a dependency has been created and it becomes more difficult to evolve the service due to the dependency of the consumer on that service. In contrast, loose coupling means that a service can be improved without impacting the service consumers as long as the interface (contract) remains unchanged.
3	Non-essential service information is abstracted.	In this context, abstraction means hiding any information from consumers that they do not require to use the service effectively. This means that the internal workings of the service can evolve and be improved as long as the contract and purpose of the service remain unchanged.
4	Services are reusable	Services are supposed to be reusable. This means that the intention is for the service to be used by multiple consumers in different solutions.
5	Services are autonomous	Something that is autonomous has the power to make its own decisions. In the context of a service, autonomy means that the

No.	Principle	Summary
		service is independent of its surroundings and is able to execute independently from outside influences.
6	Services minimise statefulness	The general condition of something can be considered its state. For example, a person can be dead or alive. In application development, the "state" often refers to the condition of an instance of a business object, say an account or a transaction at run-time. An account may be open, dormant, or active, for example. However, maintaining state information requires resources (memory, CPU, etc) and in applications with many concurrent users, maintaining this information for each user can impact the scalability of the solution. Minimising the statefulness of a service is therefore important for its scalability.
7	Services are discoverable	Services are discoverable resources within an enterprise and can be referenced when discovery queries are issued.
8	Services are composable	Services can be assembled into a composition through which they are coordinated to collectively solve a problem. Each service can participate as a building block in another higher-level service composition, thereby resulting in a hierarchal structure of services.

Table 2.4: Principles of Service Design (Erl, 2008)

2.11.3 Summary

In this section, the concepts of a Service and Service-Oriented Architecture were reviewed. In this context, a Service is a software component which is independent and performs a clearly defined task for the consumer. Services exist in an SOA and the use of Services results in the following benefits:

1) an increased amount of agnostic logic (logic independent of the language and platform on which it resides), 2) a reduced amount of application logic, 3) a reduced volume of overall logic, 4) increased agility, and 5) reduced development and maintenance costs. Eight service design principles were also identified.

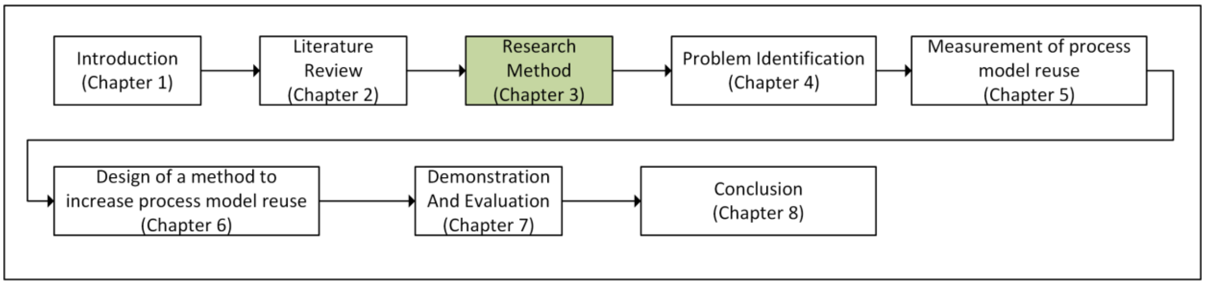
2.12 Literature Summary

Findings from the literature review indicate that different types of reuse have been identified, such as the creation of model variants, and the use of reference models. These methods can be grouped together under "reuse by adaptation". Another aspect of reuse in the process modelling domain is the reuse of fragments of a process. Research in this area has been focused on identifying fragments of process models that can be reused in different models, whether it is a single activity (with supporting information) or logic involving multiple activities in a process model that is then transported to another model.

Factors impacting the "intention" to reuse have been investigated and were classified into individual factors, process model factors, technological factors, and organisational factors (Nolte et al., 2013). However, while the benefits that should arise from reuse in process modelling have been described in the literature, the realisation of these benefits has been questioned (Fellmann et al., 2014). These benefits are described individually (e.g. improved model quality, improved efficiency), but no research was found in the literature which considered the relationships between the factors influencing process model reuse, and the corresponding impact on the costs to the organisation.

This researcher was surprised to find that while the measurement of software reuse has been extensively researched, and various methods to measure this have been proposed, no research could be found that described how to measure the reuse of process models quantitatively. Only one study was identified that described the reuse of process models based on interviews with PMs. Furthermore, only 10.2% of respondents indicated that they reused complete process models (Koschmider et al., 2014). No studies were found that described a LOR of complete process models. It should not be surprising, then, given the lack of a formal, quantitative measure of the LOR, that little practical advice was found in the literature which would increase the LOR of process models in a process repository.

3 Chapter 3 Research Method



This chapter demonstrates that the selected research method is appropriate and explain how it answers the research questions posed. The empirical situation, paradigmatic assumptions, research strategy, and intended theoretical contribution are presented. The concept of the research onion is used to describe the various components of the research and present a holistic overview of the methodology adopted (Saunders et al., 2023). The components of the research onion comprise the Research Philosophy, Approach to theory development, Methodological choice, Research strategies, Time Horizon, and Techniques and Procedures

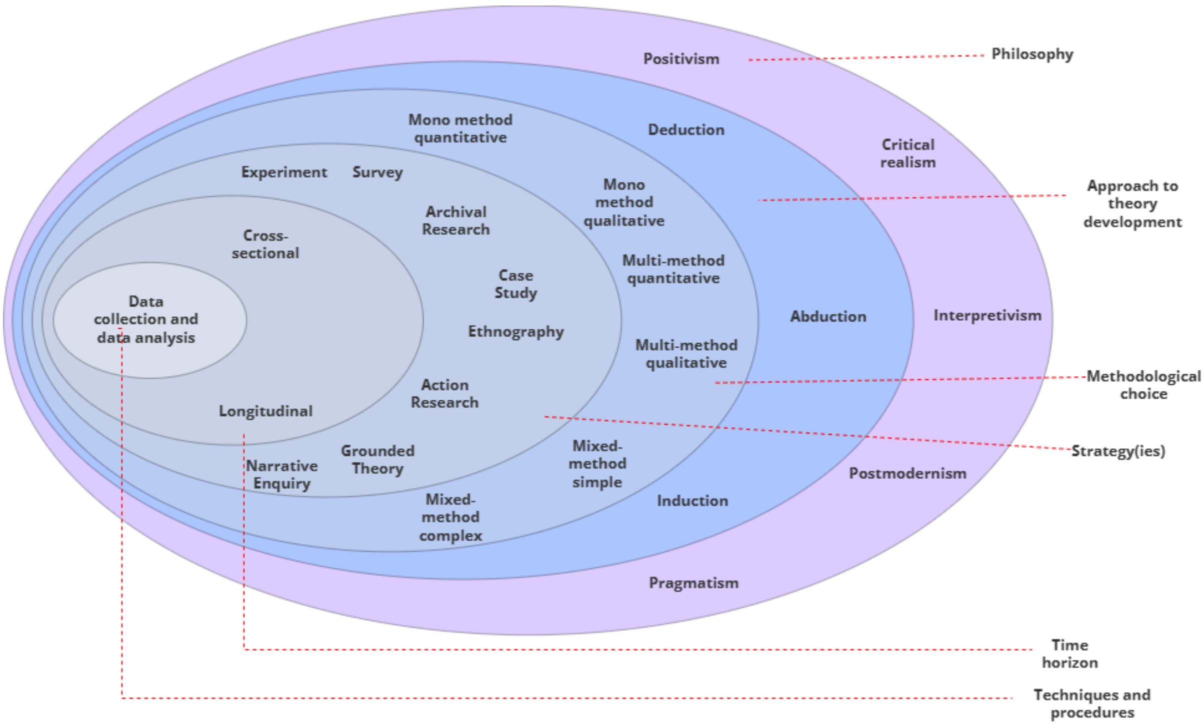


Figure 3.1: Research Onion (Saunders et al., 2023)

This research is focused on the development of a process modelling method which increases the level of reuse of complete process models. The main research question for this study was as follows:

How can the level of reuse of complete process models be increased?

3.1 Research Philosophy

The system of beliefs and assumptions about the development of knowledge is known as "research philosophy" (Saunders et al., 2023). These assumptions comprise ontological assumptions (about the realities encountered), epistemological assumptions (about human knowledge), and axiological assumptions (about the extent and ways our own values influence the research process). These assumptions underpin the entire research process: the methodology, the research strategy, data collection techniques, and the analysis procedures. An awareness of these assumptions will guide the researcher in the selection of appropriate data collection and analysis techniques, as well as the interpretation of the results. These concepts are defined below:

a) **Ontology**: Ontology relates to assumptions about how the world is made up. Does something exist objectively and have a reality external to social actors, or is reality a social construction created by social actors?
b) **Epistemology**: Epistemology relates to how knowledge is obtained. Can reality be measured, or must it be interpreted?
c) **Methodology**: Methodology relates to how one goes about finding out. It is the plan for conducting the research and was informed by the ontological and epistemological assumptions made.
d) **Methods**: Methods relate to the techniques that are used to collect and analyse the data. Again, the methods used are informed by the underlying ontological and epistemological assumptions made. Typically, these methods are quantitative or qualitative, although a combination of each can be used within the same study.

The philosophies relevant to this study are now discussed:

3.1.1 Positivism

Positivist research relates to the ontological assumption that reality is not socially constructed; it is real, external, and independent of the researcher (e.g. gravity). From an epistemological perspective, the phenomenon being researched is observable and scientifically measurable. Axiologically, the research is value-free; the researcher is detached and independent of the phenomenon being researched. The researcher adopts an objective stance as the results are true or false and not open to various interpretations. Methodologically, a positivist philosophy usually results in deductive, quantitative research.

3.1.2 Interpretivism

Interpretive research studies phenomena which are socially constructed (humans have created the reality) and may have multiple meanings. Such phenomena cannot be studied in the same manner as physical phenomena, and accordingly, different techniques are required. Epistemologically, the focus is on narratives, perceptions, and interpretation. Axiologically, the researcher is part of the phenomenon being studied and adopts a subjective stance. Methodologically, an interpretivist philosophy usually results in inductive, subjective research.

3.1.3 Pragmatism

Pragmatism as a philosophy originated in the late 19th century when Peirce and James argued that a philosophical question is of no interest if there are no practical consequences (Rylander, 2012). Pragmatism reconciles objectivism and subjectivism by considering theories and hypotheses in terms of how they result in action and practical consequences. For a pragmatist, knowledge is derived from enabling successful action (Saunders et al., 2023). Pragmatic research is a blend of positivism and interpretivism “concerned with action and change and the interplay between knowledge and action” (Goldkuhl, 2012). Ontologically, pragmatism views reality as the practical outcome of ideas. Epistemologically, in pragmatism, theory and knowledge are those which have practical use and enable successful action. Methodologically, multiple methods can be used: mixed, qualitative, or quantitative, as the emphasis is on practical solutions, and therefore, the appropriate method is the best method to answer the research question (Hanson et al., 2005).

3.1.4 Position of this research project

This research project is focused on the development of an IS artifact which addresses a real-world problem. Accordingly, a pragmatic philosophy is appropriate as this artifact must result in action and useful consequences (Goldkuhl, 2012). However, during the research project, other research questions arose for which both positivist and interpretivist philosophies were appropriate in answering those specific research questions.

A pragmatic ontology was adopted in this research project as the knowledge that is being sought is not positivistic (a single truth), nor is it constructivist (created by society or individuals). Instead, it is based on its usefulness. If the knowledge is not useful (i.e., in the context of this research project, if the resulting artefact is not useful), then it does not constitute knowledge (Hevner et al., 2004). Epistemologically, a pragmatic stance was adopted in this research project. The reality (knowledge) is neither measured directly nor interpreted. Instead, the best method is one that solves the identified problem (Hanson et al., 2005).

3.2 Research Strategy

Credible research requires a systematic investigation resulting in the development of theories, and solving problems, with a traceable path through the research process (Gough et al 2012, Dresch et al. 2015a). Correctly chosen methods, techniques, and scientific procedures provide the necessary rigor to ensure the validity of the results.

Design is a common aspect of our lives: new products, new ways of doing things, new technology, and new services all involve some sort of design process. Changing the way that something is done is a design process (Simon, 1996). Design Science Research (DSR) and Action Research (AR) are two research methods recognised in Academia which can be used to design artifacts or implement changes which result in knowledge, providing appropriate rigour is maintained through the research process. Due to the similarities between DSR and AR, a third methodology which combines both DSR and AR, known as Action Design Research (ADR) has also been proposed (Cole et al., 2005; Sein et al., 2011).

Design Science Research (DSR) is concerned with producing purposeful IT artifacts such as constructs, models, methods, and instantiations (Gleasure, 2015; Hevner et al., 2004; March & Smith, 1995; Vaishnavi & Kuechler, 2004). While the artefact designed might be useful, to qualify as scientific knowledge, rigour must also be demonstrated (Koppenhagen et al., 2012). Accordingly, to ensure rigour, a DSR contribution requires that the following criteria are met: (1) A problem is identified, (2) No suitable solution exists, (3) a new/improved artifact is designed, (4) rigorous evaluation of the utility of the artifact is conducted, (5) the contribution to IT knowledge is described, and (6) the implications are explained (March & Storey, 2008).

The aim of AR is to understand how and why a member, or a group behaves in a particular way in a given context (Coghlan, 2019). The researcher and members of the organisation work together to intervene in the situation being investigated. Table 3.1 compares the characteristics of DSR and AR (Collatto et al., 2018).

Action Design Research (ADR) brings together both DSR and AR. It involves researchers and participants jointly identifying a problem, designing an artifact to solve the problem, and evaluating the solution that addresses that problem in that context (Cole et al., 2005).

Table 3.1: Action Research vs Design Science Comparison (Collatto et al., 2018)

Characteristics	Action research (AR)	Design Science Research (DSR)
Epistemological paradigm	Traditional science (natural and social)	Design Science
Objectives that can be achieved	Solves or explains the problems of a system generating knowledge for practice and theory	Develops artifacts that offer satisfactory solutions to practical problems. Contributes in developing theories (mid-range theories)
	Explores, Describes, Explains and Predicts.	Designs and Prescribes
Main activities planned for a proper conduction of the research	Based on traditional science, the AR is conducted through exploration of the theme, data collection, action planning, implementation, evaluation and publicizing of data.	Problem identification, awareness of the problem, systematic literature review, identification of artifacts and existing classes of problems, artifact design, development of the artifact, evaluation of the artifact, making the learning explicit, generalisation of the artifact to a given class of problems, conclusions and reporting of results.
Main research results	Constructs, Hypotheses, Descriptions, Explanations, Actions	Artifacts (Constructs, Models, Methods, Instantiations, Design Propositions)
Generated knowledge	About how things are or how they behave	About how things should be
Role of the researcher	Multiple, depending on the type of action research	Designer and/or evaluator of the artifact
Collaboration between researcher-participant	Required	Not required
Empirical basis	Required	Not required
Implementation	Required	Not required
Evaluation of the results obtained by the research	Confrontation with theory	Applications, simulations, experiments using the artifact
Nature of data (collection/analysis)	May be qualitative or quantitative	May be qualitative and/or quantitative
Specificity of the research results	Specific situations	Generalisable to a certain class of problems.

DSR was selected as an appropriate methodology to conduct this research for the following reasons:

- The researcher identified the problem and designed the artifact. Although the researcher was working in the organisation, there was no collaboration identifying and developing the artifact.
- The result was an artifact (modeling method), not an action.
- The generated knowledge is an example of "how things should be".
- Evaluation of the artifact was based on using the artifact in the organisation.
- The research results are generalisable to a class of problems (business process modeling).
- The main objective of DSR is to design artifacts and prescribe solutions (Hevner, 2007; March & Smith, 1995).

The specific DSR methodology proposed by (Peffers et al., . (2007) was adopted for this research project. While this methodology provides for multiple entry points, it does not require that the stages be executed in exactly the sequence presented (Peffers et al., 2006; Venable et al., 2017). This approach comprises six stages, as shown in Figure 3.2.

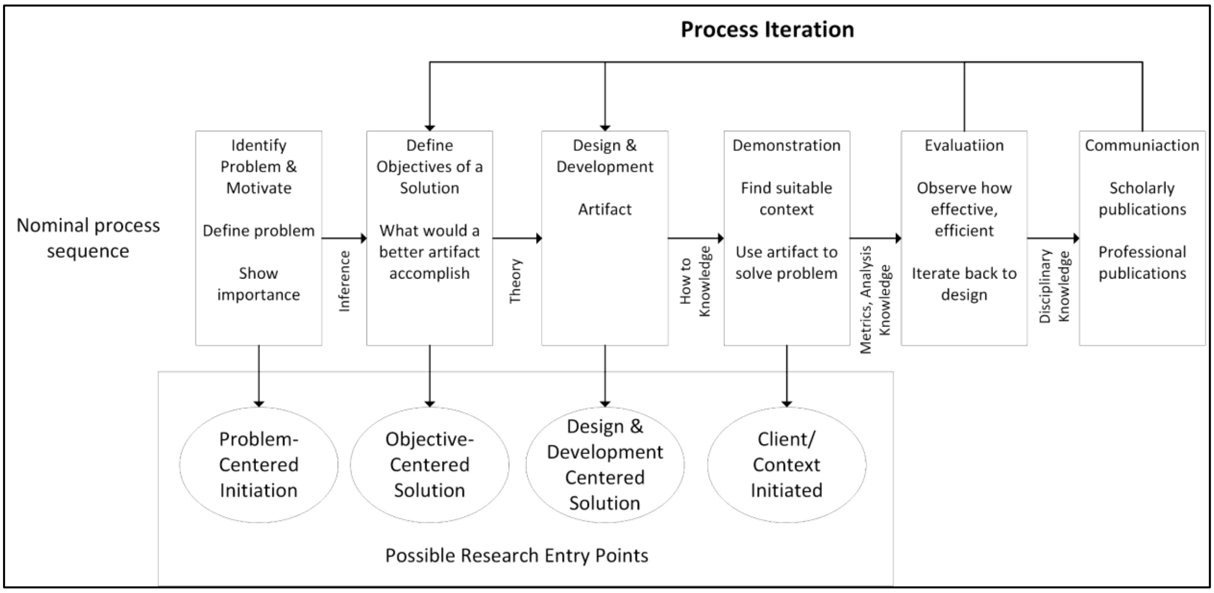


Figure 3.2: DSR approach proposed by Peffers et al. (2007)

A summary of each stage in the context of this research is shown in Table 3.2. Furthermore, each stage may consist of research questions, research data, research instruments, data collection, and data analysis. Throughout the rest of this dissertation, these are identified for each stage where appropriate.

Table 3.2: Research questions and objectives

Stage Name	Description	Research Question	Research Objectives
Problem identification and motivation	Illustrate the consequences of not reusing complete process models.	Why does a lack of process model reuse impact business costs? (Contribution)	Understand the consequences of not reusing process models.
Objectives of a solution	Increase process model reuse in process repositories.	What should the characteristics of a solution be that increase process model reuse?	Identify the characteristics expected of the proposed solution.
Design and development	Develop a method that increases process model reuse.	How can the level of reuse of complete process models be measured? (Contribution)	Develop a method to measure the level of reuse of process models in a repository.
		How can the level of complete process model reuse be increased? (Contribution)	Design a new modelling method which increases the level of reuse.
Demonstration	Test the method in the organization and prove that it works.	What is the level of reuse of complete process models prior to this research project?	Confirm the level of reuse quantitatively when modelling using the previous method.
		What is the level of reuse of complete process models when using EPReS?	Confirm the level of reuse quantitatively when modelling using EPReS.
Evaluation	Use the artefact in modelling projects and prove that it is practical	How effective and efficient is the proposed method?	Prove that the new modelling method is practical.
Communication	Communicate results appropriately	N/A	This thesis is the formal communication of the new modelling method. This research project has also resulted in the presentation of two papers at international conferences, as listed in the Publications' list.

3.3 Research Questions

Research questions are critical to all research projects as appropriately formulating the research question guides how it will be studied (Blaikie & Jan Priest, 2019). This research project identified research question for each stage of the DSR approach as indicated in Table 3.2. Three of the questions make contributions to IS Theory. In addition to these three contributions, the remaining research questions ensure rigour in this research process.

3.4 Approach to Theory Development

3.4.1 Theory development in IS

The output of academic research is "Theory" (Iivari, 2020). One definition of a theory is "a generalised body of knowledge, with a set of connected statements expressing general relationships among constructs that refer to entities of different types, both real-world and theoretical" (Gregor, 2009). A recommended approach to theory development is to begin with the research questions and then determine the appropriate type of theory (Gregor, 2006). Within the Information Systems domain, Gregor identified five types of theory, as depicted in Table 3.3.

Table 3.3: Types of Theory (Gregor, 2006)

Theory Type	Distinguishing Attributes
i. Analysis	Says what is. The theory does not extend beyond analysis and description. No causal relationships among the phenomena are specified, and no predictions are made.
ii. Explanation	Says what is, how, why, when, and where. The theory provides explanations but does not aim to predict with any precision. There are no testable propositions.
iii. Prediction	Says what is and what will be. The theory provides predictions and has testable propositions but does not have well-developed justificatory causal explanations.
iv. Explanation and Prediction (EP)	Says what is, how, why, when, where, and what will be. Provides predictions and has both testable propositions and causal explanations.
v. Design and Action	Says how to do something. The theory gives explicit prescriptions (e.g., methods, techniques, principles of form and function) for constructing an artifact.

The structure of a theory in the IS domain has also been studied, and seven components have been proposed, as elaborated in Table 3.4 (Gregor, 2006).

Table 3.4: Structural Components of a Theory (Gregor, 2006)

Theory Component	Definition
Components Common to All Theory	
Means of representation	The theory must be represented physically in some way: in words, mathematical terms, symbolic logic, diagrams, tables or graphically. Additional aids for representation could include pictures, models, or prototype systems.
Constructs	These refer to the phenomena of interest in the theory (Dubin's "units"). All of the primary constructs in the theory should be well-defined. Many different types of constructs are possible: for example, observational (real) terms, theoretical (nominal) terms and collective terms.
Statement of Relationship	These show relationships among the constructs. Again, these may be of many types: associative, compositional, unidirectional, bidirectional, conditional, or causal. The nature of the relationship specified depends on the purpose of the theory. Very simple relationships can be specified: for example, "x is a member of class A.".
Scope	The scope is specified by the degree of generality of the statements of relationships (signified by modal qualifiers such as "some," "many," "all," and "never") and statements of boundaries showing the limits of generalizations.
Components Contingent on Theory Purpose	
Causal explanations	The theory gives statements of relationships among phenomena that show causal reasoning (not covering law or probabilistic reasoning alone).
Testable propositions (hypotheses)	Statements of relationships between constructs are stated in such a form that they can be tested empirically.

Theory Component	Definition
Prescriptive statements	Statements in the theory specify how people can accomplish something in practice (e.g., construct an artifact or develop a strategy).

The application of these theory types and how their components relate to this research project are explained in Section 3.4.2.

3.4.2 This research project's contribution

This research project makes three contributions in the form of IS artifacts to IS knowledge. This section discusses the specific research questions making these contributions and positions the contributions in the context of IS Theory. Not all the answers to the research questions listed are intended to contribute as they form part of the DSR strategy adopted. Only those questions which contribute to IS Knowledge are discussed below and summarised in Table 3.5.

RQ1: How can the level of complete process model reuse be increased?

This research question is the focus and aims to prescribe a process modelling method which increases the level of reuse of complete process models in a process repository, and accordingly, the Theory for Design and Action is appropriate. The outcome was a new PA and modelling method. The criteria for a contribution to this type of theory are utility, novelty, and persuasiveness, which means that it is effective (March & Smith, 1995). Recent literature continues to call for improved methods and functionality which support the reuse of process models (François et al., 2023; François & Plattfaut, 2024; Polyvyanyy et al., 2020; Savary-Leblanc et al., 2024).

RQ2: Why does a lack of process model reuse impact business costs?

This research question aims to explain the consequences of not reusing process models, and accordingly, the Theory for Explaining (also referred to as the Theory for Understanding) is appropriate (Gregor, 2006). This theory is used to understand causal factors that contribute to a particular real-world situation. No research could be found in the literature relating to the system dynamics that develop when process models are not reused even though the benefits of reuse are mentioned (Baumann et al., 2023; Chen et al., 2020; François et al., 2023; François & Plattfaut, 2024; Khider et al., 2020; Yongchareon et al., 2020). The outcome was a causal-loop diagram illustrating the system dynamics that arise in an enterprise when process models are not reused. Criteria for a contribution of this type of theory is that it should be new and explain something that was not previously well understood (Gregor, 2006).

RQ3: How can the level of reuse of complete process models be measured?

This research question aims to prescribe a method for calculating the level of reuse of complete process models in a process repository, and accordingly, the Theory for Design and Action is appropriate. The outcome was an algorithm that can be used to determine the level of reuse of process models in a repository. It was found that while much research had been carried out measuring the level of reuse of software components, no such research could be found relating to process models. Criteria for a contribution for this type of theory are utility, novelty, and persuasiveness that it is effective (March & Smith, 1995).

Table 3.5: IS Theory constructs applied to this research project's contribution.

Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
Components Common to All Theory			
Methods of representation	Words, diagrams, tables		
Primary Constructs	Level of reuse, business processes, patterns, PA, and services.	Relationships between factors resulting from or impacting the level of reuse	Level of reuse, business processes, models, functions, object definitions, object occurrences
Statements of relationship	Provides a method (architecture, ERD, process) and explanations for how to design a method which increases the level of reuse of process models.	Causal Loop diagram indicating the system dynamic resulting from the lack of reuse of process models.	Provides a formula for the calculation of the level of reuse based on variables which can be quantitatively measured in a repository.

Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
Scope	Enterprises conducting BPM using a process repository	Enterprises conducting BPM using a process repository	Enterprises conducting BPM using a process repository
Components Contingent on Theory Purpose			
Causal Explanations	Underlying theories that explain the design include the Technology Acceptance Model when applied to BPM. Different factors impacting the PM's "inclination to reuse" are considered.	The statements of relationships include causal explanations.	None
Testable propositions	The claim is made that the design theory will assist other organisations to increase the level of reuse of process models.	The propositions (relationships between the elements) are tested and confirmed in interviews with participants.	The claim is made that the design theory will assist organisations in measuring the level of reuse of process models in their repository.
Prescriptive Statements	The design theory defines the meta-requirements and principles which can be used to increase the	Not present. The design theory is explanatory in nature.	The design theory specifies a formula and algorithm that can be used to measure the level of reuse of process models quantitatively.

Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
	level of reuse of process models.		

3.5 Empirical Situation

The researcher was contracted to the organisation where this research was conducted (and was previously employed by the same organisation) and, therefore, has insight into the process modelling work that has taken place over a period of approximately 20 years.

This research project was carried out in a large South African Financial Services organisation with more than 10 000 employees. The organisation has been documenting business processes using ARIS for over 20 years and has approximately 200 PMs active in the repository. The research project was cross-sectional, as a single snapshot of the process repository was analysed. The following resources were used to describe the empirical situation prior to developing a new modelling method:

- Historical documents, such as the scope of work for this project, and internal presentations relating to BPM within the business unit where this research was carried out.
- Process modelling guidelines issued by the “Process Value Center” (PVC) were reviewed.
- Analysis of the interviews conducted with PMs.

In this section, the empirical situation is described, and the issues that pertain to the reuse of complete process models are highlighted and linked to factors impacting process model reuse, which have been published in the literature. This research project then focuses on actions which will increase the level of reuse.

3.5.1 Process Modelling Tool

This research project was carried out using ARIS as the modelling tool. The key design aspects and concepts of the ARIS repository are described in Appendix 1.

3.5.2 Process Modelling Organisational Structure

The responsibilities for BPM are split across the different business areas within the bank. A Process Center of Excellence (PCoE) was responsible for defining and communicating the modelling method and managing the process repository overall. The BA was the responsibility of the Architecture Business Unit (ABU). This is summarised in Table 3.6.

Table 3.6: Process Modelling Organisational Structure

Business Unit	Responsibilities
Business Unit Teams	Definition of PA Levels 3,4 and 5 Creation and maintenance of models
Process Center of Excellence	Definition of the modelling method
Business Architecture	Definition of the PA Level 1 and Level 2

However, EPReS was tested in only one of those teams. In this research, we will refer to the team that tested this method as **Team X**. For the purposes of this research, we can combine all the other areas into one team, **Team Y**, which continued to use the previous modelling method. Each team stores their process models in their own area of the process repository in a logical data structure which is analogous to the Microsoft Windows folder structure. Within Team X, the models created using EPReS were stored in a structure separate from the area where the models created using the old method were stored. This enables the identification of three different populations of process models: 1) Models created by Team X using EPReS; 2) Models created by Team X using the old method; 3) Models created by Team Y using the old method. This is illustrated in Figure 3.3.

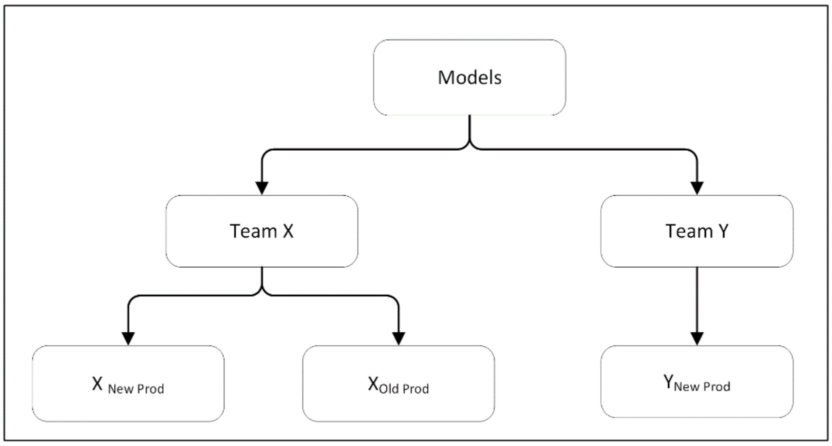


Figure 3.3: Repository Model Populations

3.5.3 Process Architecture

Levelling

The organisation uses a 5-level PA, Levels 4 and 5 being the levels at which the PMs create or update models, as illustrated in Figure 3.4. However, no guidelines existed to integrate the detailed process models (L4 and L5) into the higher levels. This resulted in each modelling area adopting its own L4/L5 architecture. The categorisation of the processes and the structure within the process repository was decided within each modelling team.

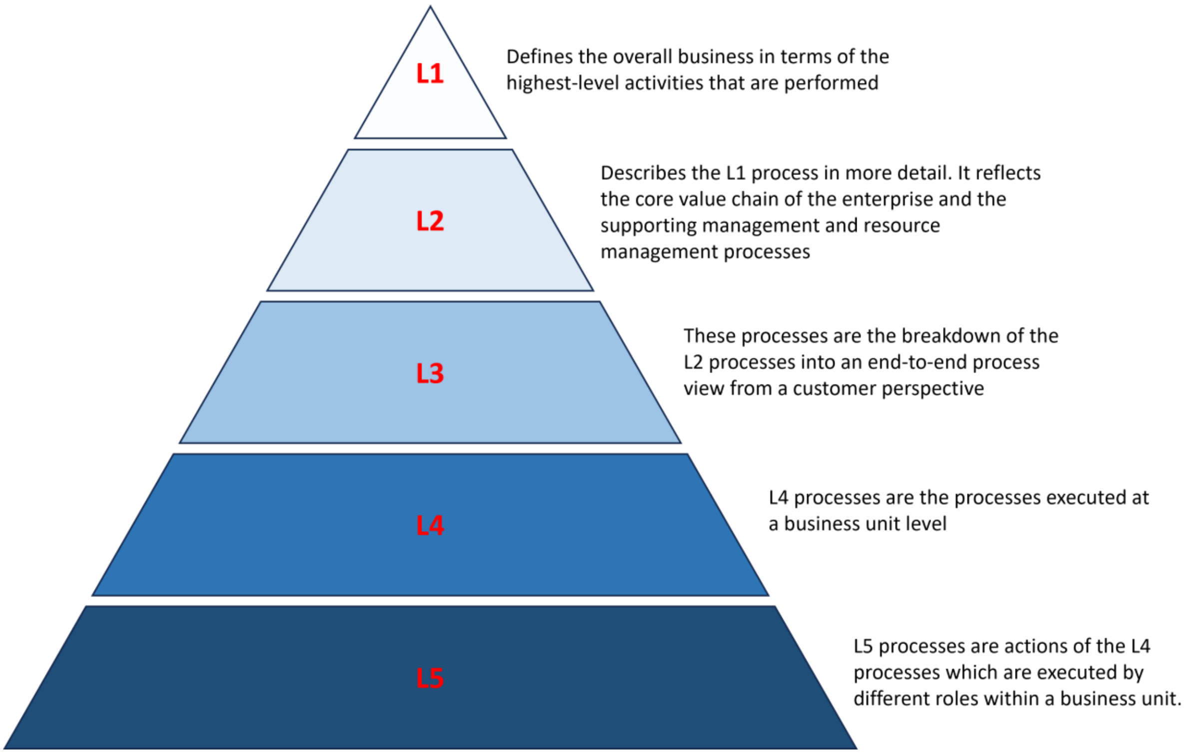


Figure 3.4: Process Architecture Levels

The Business Architecture Unit (BAU) was responsible for the definition of Levels 1 and 2, while the PMs were responsible for the creation and maintenance of the procedural models at Levels 4 and 5.

Model Types

The key model types used by the PMs are described in Table 3.7.

Table 3.7: Process model types used

Model Type	Description	Purpose	Comment
Value-Chain	Value-Added Chain Diagram	Value-Added Chain models illustrate both the sequence of processes and the hierarchical relationships between structurally relevant processes.	
Control-Flow Models	BPMN collaboration diagram (BPMN 2.0) Enterprise BPMN collaboration diagram EPC EPC (column display) EPC (row display)	Depicts the sequence of activities required to achieve the objective of that process.	
Other Model Types	Function Allocation Diagram (FAD)	The FAD is assigned to a Function object and depicts the relationships between the business objects which support the Function. For example, the FAD might reflect the role that carries out the Function, the data inputs required, and the data produced.	The FAD is an important model when trying to reuse complete process models. Multiple roles can often carry out a particular activity. If these roles are all reflected in the control-flow model, that model can become cluttered. Also, users often don't want to see all the other Roles on the model, and this can result in a new model being created to offer a suitable

Model Type	Description	Purpose	Comment
			customised view of the process. Removing this information from the control-flow model by placing it in a FAD simplifies the control-flow model and reduces the inclination of PMs to create customised process models, which are logical duplicates of existing models.

Model Content

The term "Fat model" is used to indicate that the model is enriched by depicting additional information relating to the activities on the model canvas itself. For example, the CRM application is drawn on the model and connected to the activity that it supports using a "supports" relationship. Other information that was permitted on the model canvas was input/output data, roles, screens, risks, etc. An example of such a model is shown in Figure 3.5**.**

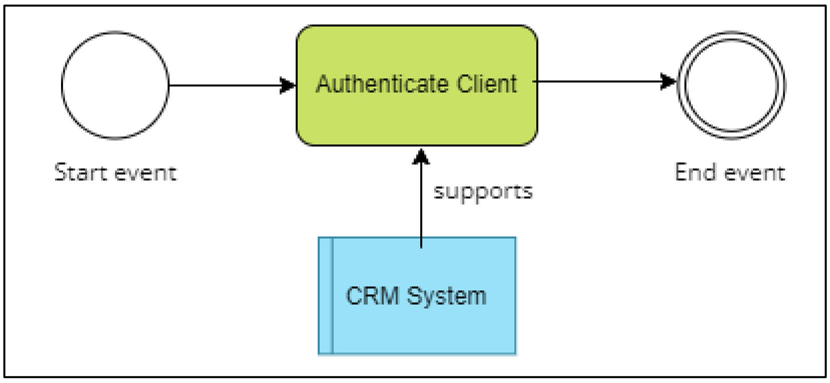


Figure 3.5: Example of a "Fat" model

Folder Structure

The folder structure within the repository was structured based on the business units rather than an overall BA. Each business unit was free to adopt any folder structure that they considered suitable within their space. Team X used a set of Production folders and a set of Development folders, as described in Table 3.8.

Table 3.8:Empirical Repository Folder Structure (Team X)

	Folder Example	Comment
Production (L4 Process)	/Main group/Process/Team X Process/L4 Production Processes/Financial Transaction Processing/L4 Initiate transaction	The folders "Financial Transaction Processing" and "Initiate transactions" were decided by Team X. They were not informed by the BA as defined by the ABU. From a hierarchical perspective, there is no link between Architecture L3 and L4.
Development (L4 Process)	/Main group/Process/Team X Process/L4 Development Processes/L5 Development Processes/ ModellerName	The Development folders are structured based on the PM name. A PM may work on multiple projects simultaneously and would then create further sub-folders based on the project name.
Production (L5 Process)	/Main group/Process/Team X/L4 Production Processes/Financial Transaction Processing/L4 Initiate transaction/L5 Initiate transaction	The L5 process models reside in a sub-folder of the L4 folder with a similar name
Development (L5 Process)	/Main group/Process/Team X Process/L4 Development Processes/L5 Development Processes/ModellerName	The Development folders are structured based on the PM name. A PM may work on multiple projects simultaneously and would then create further sub-folders based on the project name.

While the details of Table 3.8 are specific to Team X, it illustrates the following:

- Each modelling team and PM determined the repository folder structure.
- There was no link between the models created and the official BA.
- The folder structure in the examples listed can be up to 7 levels deep
- The L4 and L5 models are not linked to the high-level architecture. They start at L4.

3.5.4 **Roles**

The empirical BPM lifecycle consisted of two roles: PM and Gatekeeper (GK). The PM was provided with models (copies or blank models) by the GK, who was the interface between the repository and the PM.

Rules

- In Team X, the PM was not permitted to create new models. This rule was implemented to reduce the number of redundant/duplicate models in the repository.
- Certain model and object attributes are mandatory. For example, the Process Architecture Level attribute, which indicates the level in the PA hierarchy, must be populated in all control-flow models and Functions.

3.5.5 **Modelling Process description**

Prior to the development and implementation of EPReS, Team X had previously adopted an internal approach to control the creation of new models. While from a PM's perspective the process to create a new model was the same as that to update a model, from the GK's perspective, the two are different due to the model and object management required. An overview of the empirical process of updating a model is depicted in Figure 3.6, and the empirical process of creating a new model is depicted in Figure 3.8. The corresponding empirical model and object management carried out are depicted in Figure 3.7 and Figure 3.9.

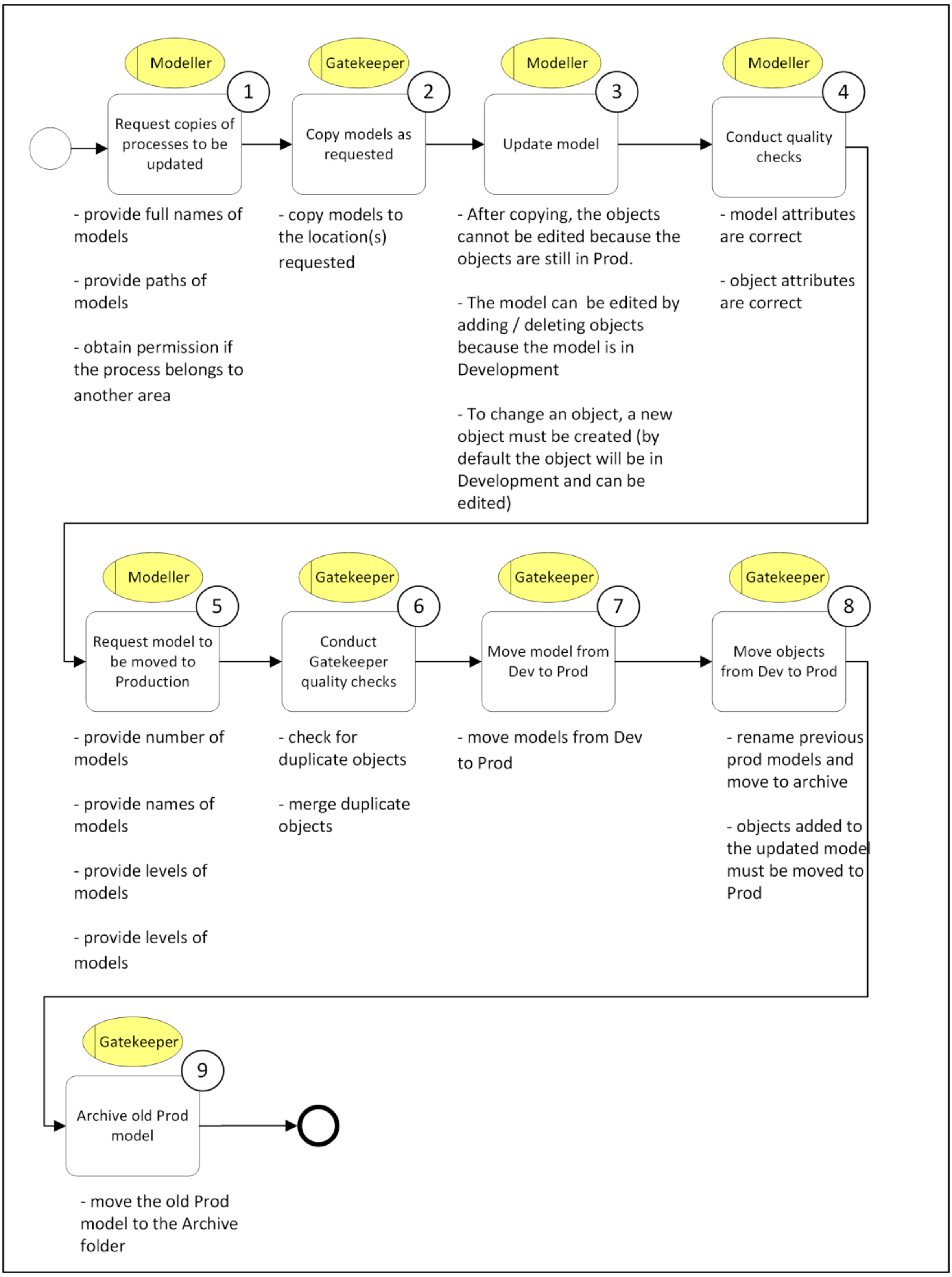


Figure 3.6: Overview of the empirical process to update a model

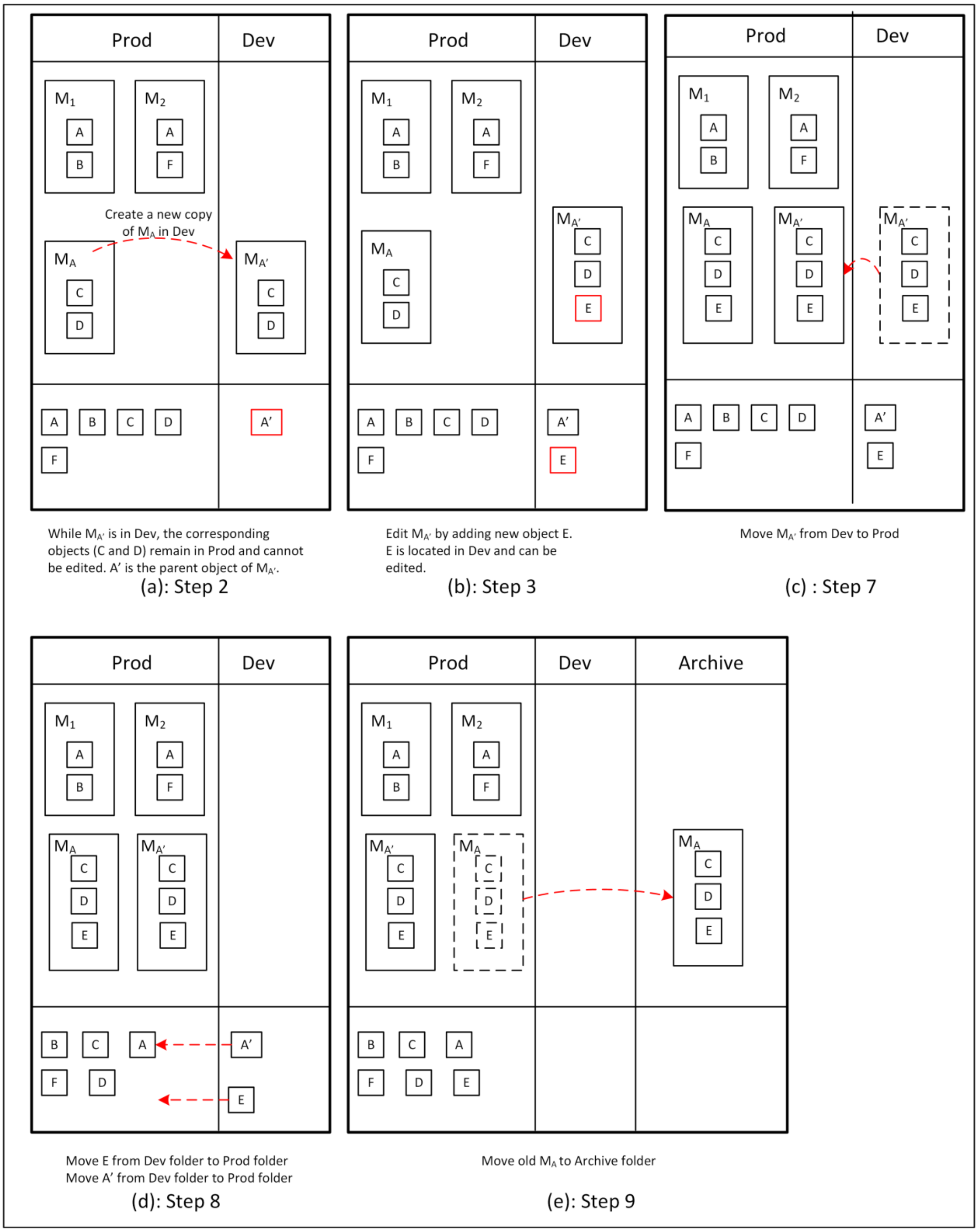


Figure 3.7: Empirical process to update a model: Model and object management.

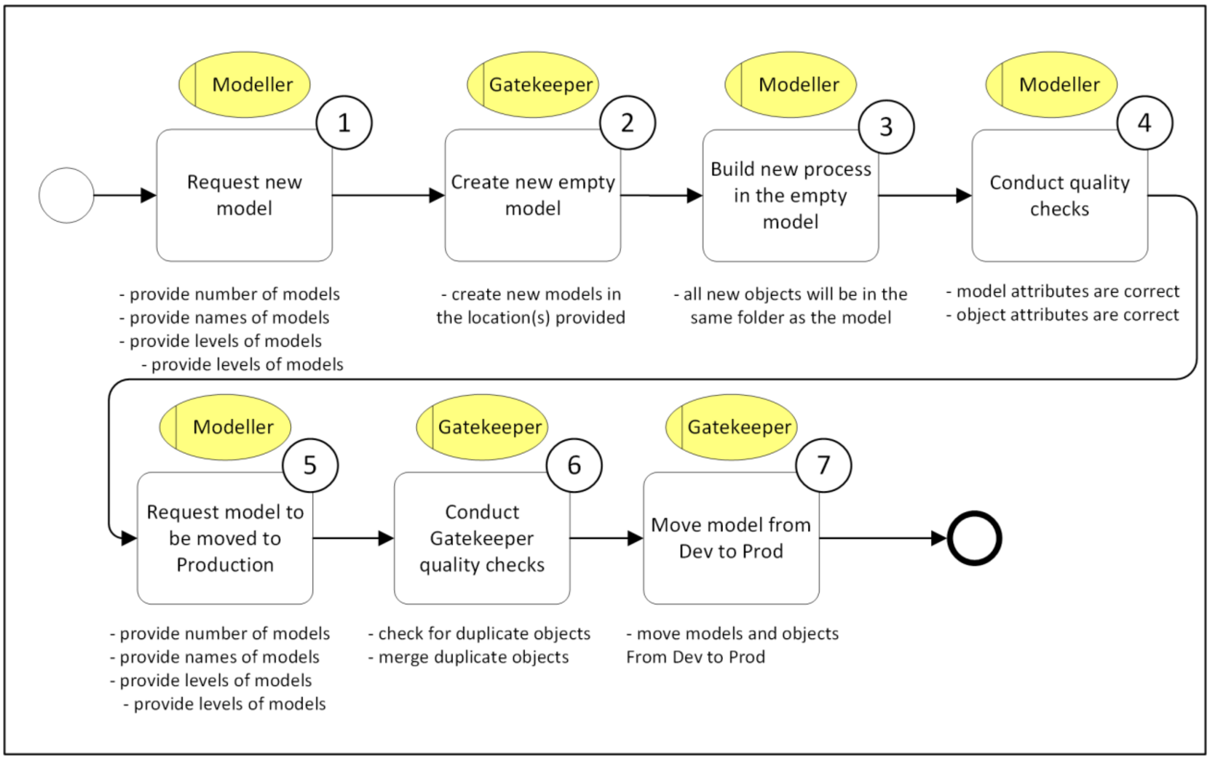


Figure 3.8: Empirical process to create a model.

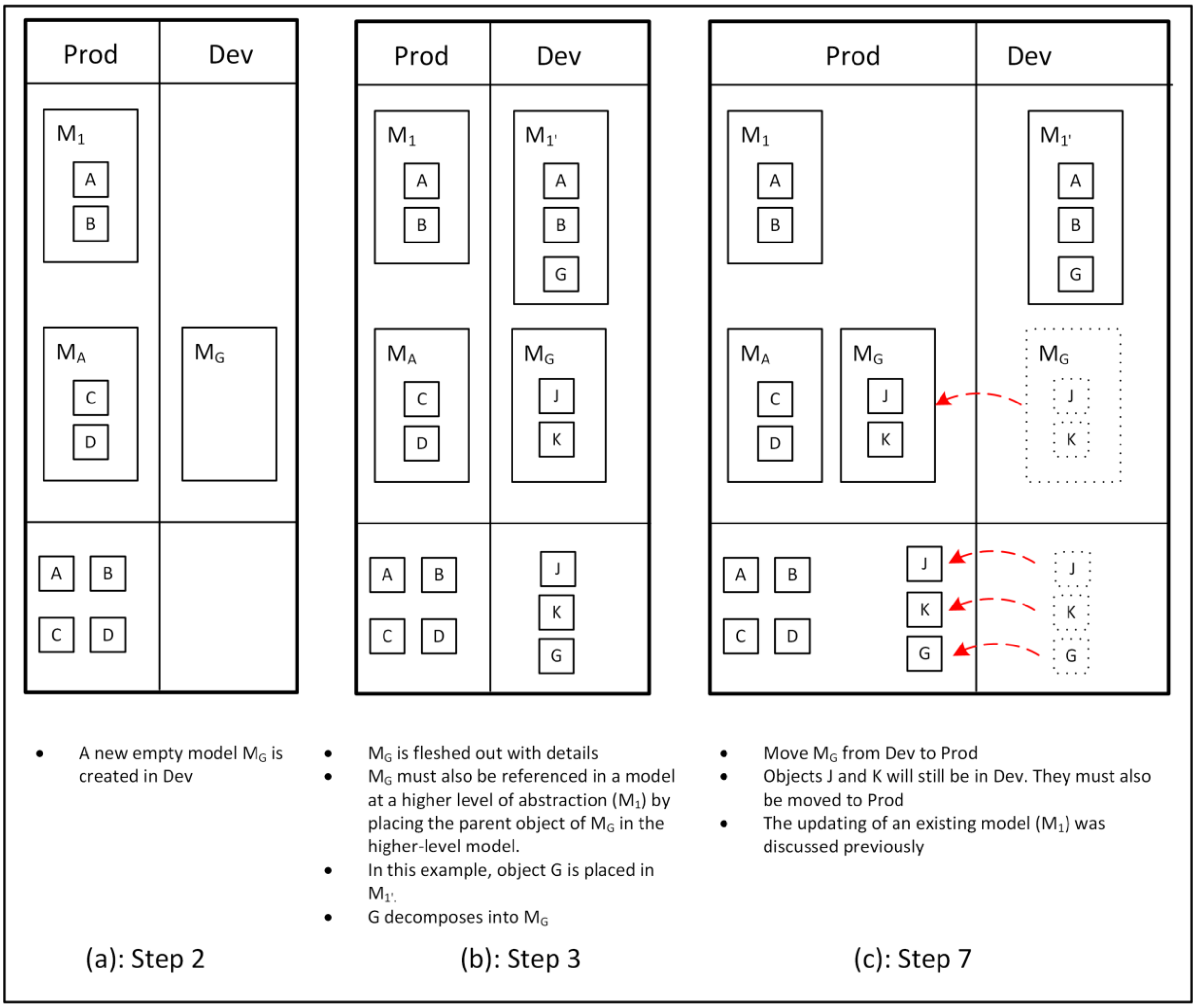


Figure 3.9: Empirical process to create a new model: Model and object management

3.5.6 <u>Multiple projects impacting the same process model</u>

The organisation also faced the situation where multiple projects running in parallel would impact the same process model. Furthermore, while there would be an overall implementation plan for these projects, it could not be guaranteed in which sequence the projects would finally be implemented. Sometimes, one of the projects would not be implemented at all. This forces the process models for each project to be updated separately (a copy of the model is required for each project), and then before each project is implemented, the process models must be compared to the latest Prod version of each model in case one of the other projects has been implemented first and made changes to the process model which have not yet been updated in the next project’s process models.

3.5.7 <u>Summary of empirical issues experienced</u>

A summary of the issues related to the empirical situation in Team X is described in Table 3.9

Table 3.9: Summary of Empirical Modelling Issues

Issue ID	Name	Description	Consequence	Factor Impacting reuse (Nolte et al., 2013)
1.	Governance – GK creates new or copies of models	Fully specify the source and target paths of models when requesting copies from Prod or when requesting models be published.	Tedious and error-prone because some folder structures are at least six levels deep. Models moved to the incorrect locations are difficult to find	**Software ease of use**
2.	Multiple active copies of the same models	Multiple projects can touch the same process models simultaneously	Multiple copies of the same process model are required, leading to complex model and object management when publishing process models.	**Perceived usefulness of a model**
3.	Model management	The process for creating new models and updating existing models is different when publishing such models.	Incorrect handling of these scenarios results in a corrupted repository.	**Software usefulness** **Software ease of use**
4.	Object management	Parent objects of models which have been updated must be "merged" with the original parent object of the model that was updated.	Incorrect handling of these scenarios results in a corrupted repository	**Software usefulness** **Software ease of use**

3.5.8 Ethics & Confidentiality

The ethics and confidentiality requirements relating to this research were of the utmost importance. To this end, the following was observed:

- Permission was obtained from the organisation to conduct this research (request for permission is contained in Appendix 5).
- The organisation remains anonymous.
- Interview participants remain anonymous.
- Participation in interviews was voluntary.
- All data gathered was kept in a location secured by a password.

The request for permission to conduct the research in the organisation is shown in Appendix 5. The interview cover letter is included in Appendix 6. The Informed Consent form is shown in Appendix 7.

3.6 Research data, research instruments, data collection, and data sampling risks

In this section, the research data, research instruments, data collection, and sampling risks are discussed. We first present an overview and then discuss how each was carried out in the different DSR stages. This is presented for each stage identified in the DSR cycle to provide a consolidated view of the research data, instruments, data collection, and sampling risks for each stage. A mixed-methods methodology was used to conduct the research. Mixed methods are usually necessary when more than one research question is being investigated, which, as previously explained, applies to this research project (Blaikie & Priest, 2019). Mixed methods are also considered to be appropriate for a pragmatic paradigm where answering the research question (i.e. solving the identified problem) is the primary objective (Biesta, 2010). The methods used consisted of quantitative and qualitative approaches using literature reviews, interviews with stakeholders and statistical analysis of process repositories. The methods vary from positivist (statistical analysis of historical process repositories) to interpretivist (interviews being used to evaluate the artefact in a real setting), depending on the research question in focus. The research data, instruments, data collection, and sampling risks are now discussed for each DSR stage.

3.6.1 Research Data Overview

The data collected in this research project was both qualitative and quantitative. Qualitative data was collected from interviews with participants, while quantitative data was collected by exporting the data from the repository and importing it into an SQL Server database for processing.

Quantitative Data

Quantitative data was measured by analysing the process repository of the organisation where the research was conducted. The repository data was analysed by exporting the required information from the repository and importing it into Microsoft Access for further analysis, as the repository itself was not suitable for conducting the analysis required. The process for this is shown in Appendix 3.

The structure of the process repository is such that it is possible to measure the current state and the impact of the artefact resulting from this research project in the same repository. The process models designed using the new modelling method were specifically identified in the repository, and therefore, the results of the EPReS could be compared to the results of the current method. The process models created using the EPReS resided in a separate folder structure in the repository, and this enabled this comparison to take place.

The approach to obtaining the quantitative data is described in Appendix 3. Although the frequency of reuse of a particular model did not change during the analysis, the population of the models used in the analysis could vary depending on the criteria used to include the model in the population (e.g., business area; modelling method used). In the context of this discussion, "population" is used to mean all the models that meet a certain criterion. It was not necessary to select a sample from this population because the analysis is automated, and therefore, the entire population, once identified, was included in the sample.

Qualitative data

Qualitative data was collected in the form of interviews with participants and was used to answer RQ 2 by analysing the transcripts of the interviews and searching for quotations which confirmed the relationships between elements related to BPM, process model reuse, and costs to the business. To evaluate the artifact, the interview transcripts were analysed by searching for quotations that related to the benefits and results of the new process modelling method.

Semi-structured interviews with ten PMs who tested and used the new modelling method were conducted to obtain stakeholder perspectives of the problem, solution objectives, and the evaluation of the new modelling method. The demographic details of these PMs are described in Appendix 9. Semi-structured interviews were selected because this approach permits the interviewer to ask additional questions to probe the interviewee for better clarity or completeness while still following a structured set of predefined questions (Kajornboon, 2008).

Purposeful sampling (Marshall, 1996) was adopted when identifying stakeholders to interview. Purposeful sampling is recommended when the information being sought is specialized and limited to certain individuals within an organisation who are best able to answer the research questions.

The 11 participants interviewed were PMs on that team who were using EPReS. The interview questions were piloted with a limited number of subjects to confirm that interviewees understood the questions in the manner intended (Foddy, 1994). All interviews were recorded and transcribed. All interviewees were prepared to be recorded.

The participant details are described in Appendix 9. All participants had over ten years of process modelling experience, and 9 out of 11 participants had worked at the research organisation for more than ten years. The interview guide is contained in Appendix 8.

3.6.2 Research instruments Overview

Research instruments are the tools used to collect, measure, and analyse the data related to the research question(s). Research instruments that were employed were computer algorithms, which were used to analyse the process repository and generate the quantitative data, interviews (with appropriate stakeholders), and a literature review.

3.6.3 Sampling, Sampling Risks, and Bias -Overview

Although there is a risk of the researcher introducing bias during interviews, the researcher's history within the organisation enabled the researcher to probe more deeply during the interviews. The criteria for including models in the population did not introduce sampling bias because the entirety of the population was sampled.

Sections 3.6.4 to 3.6.9 now discuss the research data, research instruments, data collection, and sampling applicable to each stage of the DSR process in detail. The research questions for each stage have been repeated for the sake of completeness.

3.6.4 Stage 1: Problem identification and motivation

Research Question – Problem identification and motivation

The research question for Stage 1 is: Why does the lack of process model reuse impact business costs?

Research Data – Problem identification and motivation

This research question was answered using system dynamics to understand how the lack of process model reuse impacts business costs. System dynamics (SD) is used to explain behaviour by developing a causal theory and then using that theory to guide interventions that will change the resulting behaviour of the system and improve performance (Lane, 2008). Deductive reasoning was used to answer this research question. Using constructs and relationships between constructs identified in the literature, a model (in the form of a causal-loop diagram) was proposed. These constructs and relationships were then verified by analysing interviews conducted with PMs within the organisation being studied. No research was found in the literature regarding this perspective, so a three-step approach was taken:

1) Factors influencing process model reuse in the literature were identified.
2) Conceptual models of process model reuse were identified and reviewed to identify potential elements (Erol, 2018; Nolte et al., 2013).
3) Papers describing issues related to process modelling and process model reuse were reviewed to identify potential elements (Aldin & de Cesare, 2011; Alotaibi, 2016; Indulska, Recker, et al., 2009; Jonnavithula et al., 2015; Nolte et al., 2016; Raduescu et al., 2006).

Research Instruments – Problem identification and motivation

A literature survey and semi-structured interviews were used in this Stage to collect the required data.

Data Collection – Problem identification and motivation

A literature survey was conducted to understand the research that had been conducted regarding reuse in the BPM domain. The initial literature search was conducted in four steps. 1) Search Google Scholar using relevant search strings. 2) Identify and review relevant articles. 3) Review relevant articles which had cited those found in step 2. 4) Review any relevant references in the articles reviewed. The initial search strings used were "business process model reuse", "business process patterns", and appropriate permutations of these phrases.

Semi-structured interviews with PMs were conducted and analysed to confirm the constructs and relationships identified in the literature survey. (Qualitative data, Type: Descriptive)

Sampling, Sampling Risks and Bias – Problem identification and motivation

As previously discussed, purposeful sampling for the semi-structured interviews was adopted as the sampling strategy.

3.6.5 Stage 2: Objectives of a solution

Research Question – Objectives of a solution

The research question for Stage 2 is: What should the characteristics of a solution be that increase process model reuse?

Research Data – Objectives of a solution

The Problem Identification and Motivation stage provided the data required to define the objectives of a solution needed to solve the problem identified in Stage 1. The interviews conducted with participants provided qualitative descriptive data, which was analysed to identify further objectives from the PM's perspective.

Research Instruments – Objectives of a solution

The research instruments for this stage were semi-structured interviews.

Data Collection – Objectives of a solution

Semi-structured interviews with PMs were conducted and analysed to identify any additional objectives.

Sampling, Sampling Risks and Bias – Objectives of a solution

Purposeful sampling for semi-structured interviews

3.6.6 Stage 3: Design and development

Research Question – Design and Development

The research questions for Stage 3 are:

How can the level of reuse of complete process models be increased?

How can the level of reuse of complete process models be measured?

Research Data – Design and Development

Factors impacting process model reuse were identified from the literature so that actions which would improve reuse could focus on these factors. Methods for calculating reuse were identified from a literature search using the term "software reuse measurement" and combinations thereof. The search was extended by reviewing relevant articles that were referenced in these papers. This research project comprised three DSR cycles, each of which implemented changes based on feedback from the PMs and the GK. In the first cycle, changes were made to the PA and modelling method, which focused on building reusable process models and the PA (folder structure). In the second cycle, changes were made to the rules regarding the folder structure in the repository, and some automation of GK/PM activities was introduced. In the third cycle, further changes were made to the PA. Additional GK/PM activities were automated.

Research Instruments – Design and Development

Literature surveys were used to collect input data for the design and development stage.

Data Collection – Design and Development

Not applicable – no data was collected during this stage of the research project.

Sampling, Sampling Risks and Bias – Design and Development

Not applicable because no data was collected.

3.6.7 Stage 4: Demonstration

Research Question – Demonstration

The research questions for Stage 4 are:

What was the level of reuse of complete process models prior to this research project?

What is the level of reuse of complete process models when using EPReS?

Research Data - Demonstration

The data required to demonstrate that EPReS increases the LOR of process models was extracted from the repository. The details of this data were as follows:

- The number of times each model has been reused (the concept of complete model reuse is described in Appendix 2). Quantitative data; Type: Ratio.
- Model attributes:
    - Architecture level
    - Modelling team or Business owner (organisational area) of model
    - Modelling method used

Research Instruments - Demonstration

A statistical analysis of the process repository, which required the development of scripts and SQL Server queries, was carried out.

Data Collection - Demonstration

The repository data was collected by exporting the required information from the repository and importing it into Microsoft Access for further analysis, as the repository itself was not suitable for conducting the analysis required. The model data included attributes which enabled different populations of process models to be identified and analysed. The process for this is shown in Appendix 3. The number of process models referenced by other process models in the repository was extracted using scripting functionality provided by the modelling tool. The artifact then used this data to determine the level of reuse of process models in the data sample. The artifact was further validated by applying the measure to different samples of models (comprising 1-50 models in each sample) in the repository.

Sampling, Sampling Risks and Bias - Demonstration

The entire population identified was used in the quantitative repository analysis because it was stored in a database, and the analysis could be automated. Accordingly, there was no sampling risk or risk of bias.

3.6.8 Stage 5: Evaluation

Research Question – Evaluation

The research question for Stage 5 is: How effective and efficient is the proposed method?

Research Data - Evaluation

This stage used research data from three sources:

- Analysis of feedback from PMs obtained using semi-structured interviews (Qualitative / Descriptive)
- Cost benefits related to using the new modelling method (Quantitative / Ratio).
- The number of projects which used the new modelling method (Quantitative / Ratio).

Feedback from PMs was analysed to obtain their perspective on EPReS.

Research Instruments - Evaluation

The evaluation strategy was based on the DSR Evaluation Strategy Selection Framework proposed by Venable et al. (2012). Using this framework, it was determined that both Naturalistic and Artificial Ex Post evaluation strategies would be suitable. The naturalistic strategy would use surveys to determine the effectiveness and utility of the proposed method, while the statistical analysis of the process repositories would constitute an artificial strategy. All evaluations were conducted after the artefact had been designed (Ex Post).

Research instruments utilised in this stage were:

- Semi-structured interviews
- Internal documentation related to projects using the new modelling method and a presentation prepared by a PM estimating the cost savings of using the new modelling method.

Data Collection - Evaluation

Semi-structured interviews were analysed to obtain participants' views regarding the success of the proposed modelling method, and internal modelling team documentation was reviewed to identify information relating to the cost benefits and number of projects published.

Sampling, Sampling Risks and Bias - Evaluation

Purposeful sampling for the semi-structured interviews was adopted as the sampling strategy. Although there is a risk of the researcher introducing bias during interviews, the researcher's history within the organisation also enabled the researcher to probe more deeply during the interviews. The criteria for selecting respondents did not introduce sampling bias because all PMs who modelled using the new modelling method were included.

3.6.9 Stage 6: Communication

Research Question – Communication

The research question for Stage 6 is: How will the findings of this research be communicated?

Research Data - Communication

N/A

Research Instruments - Communication

This thesis serves as the communication instrument for this research project.

Data Collection - Communication

N/A – no data collection was required

Sampling, Sampling Risks and Bias - Communication

N/A

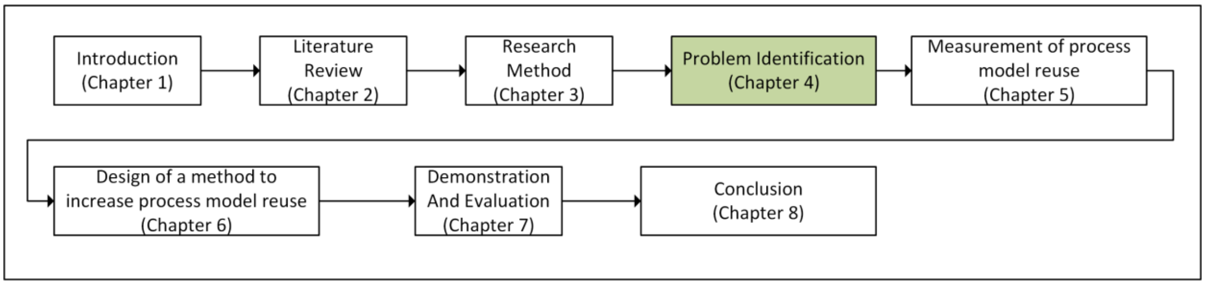
3.7 Method Summary

In this section we explained that a pragmatic ontology was adopted because a real-world problem was being addressed, and knowledge derived would depend on the usefulness of the solution. Epistemologically, a pragmatic stance was adopted. Knowledge was neither measured directly nor interpreted, but the best method is the one which solves the problem. To ensure rigour in a 6-stage DSR strategy was followed, consisting of Problem identification and motivation, Objectives of the solution, Design and development, Demonstration, Evaluation, and Communication. This research project answered three research questions which contribute to IS theory. The IS theory was developed using the approach of Gregor (2006) as summarised in Table 3.10. Ethically, permission to conduct this research project was obtained from the organisation where the research projected was conducted, Table 3.5and informed consent was obtained from all interview participants.

Table 3.10: Research Questions' Theory Types

Research Question No.	Question	Theory Type
RQ1	How can the level of complete process model reuse be increased?	Theory for Design and Action
RQ2	Why does a lack of process model reuse impact business costs?	Theory for Explaining
RQ3	How can the level of reuse of process models be measured?	Theory for Design and Action

4 Chapter 4: Stage 1 – Problem Identification – Findings and Discussion



The DSR method adopted requires that the artifact designed and instantiated addresses a real-world problem. In this chapter, we explore the consequences of not reusing business process models and develop a causal-loop-diagram (CLD), which illustrates that the system dynamics that develop due to a lack of reuse of process models result in increasing business costs. This provides the justification for this research project and the artifact developed. We answer the following research question in this chapter:

Why does the lack of complete process model reuse impact business costs?

The aim of system dynamics (SD) is to explain behaviour by developing a causal theory and then using that theory to guide interventions that will change the resulting behaviour of the system and improve performance (Lane, 2008). While the benefits of model reuse and the factors impacting reuse have been studied, the system dynamics arising from not reusing process models have not been studied. This research question was answered using system dynamics to understand how the lack of process model reuse impacts business costs. Deductive reasoning was used to answer this research question. Using constructs and relationships between constructs identified in the literature, a model (in the form of a causal-loop diagram) was proposed. These constructs and relationships were then verified by analysing interviews conducted with PMs within the organisation being studied.

Business process modelling is the foundation of all automation projects, and business process domain analysis is an expensive consulting service in reengineering projects (Daneva, 1999). Accordingly, identifying and analysing the relationships between factors that influence business process modelling costs could provide new insights that may enable the reduction of project costs.

It has been argued by Alter that there is an imbalance in IS research between tool thinking and systems thinking (Alter, 2004) and in referring to systems thinking, Alter attributes IT project failures to inadequate systems analysis of the impact of technology on the people, their practices and hence business results (Alter, 2004).

The concept of a system is not new and can be traced back even to Aristotle’s statement, “The whole is more than the sum of its parts” (Bertalanffy, 1972). However, the approach to scientific research, until recently, was one of breaking the problem down into smaller parts (effectively reducing the number of variables involved to a level that could be solved using the mathematical techniques available at that time (Weaver, 1948)) and attacking those parts individually, and while this worked, as evidenced by scientific progress in the 18th, 19th and 20th centuries, the problem of solving multi-variable problems remained.

While no single definition of model validation exists (Barlas, 1996), and no single test of model validity exists (Forrester, J. & Senge, 1980), the validity of a system dynamics model depends on its purpose. The validity of the usefulness of a model can only be assessed relative to its purpose (Barlas, 1996; Forrester, J. & Senge, 1980). Validation of system dynamics models is based on the validation of two aspects of the model: 1) Validation of the internal structure of the model and 2) validation of the model behaviour (Barlas, 1996; Forrester, J. & Senge, 1980).The seminal paper by Forrester. & Senge (1980) describes 3 categories of tests for validating SD models: 1) Tests of model structure, 2) Tests of model behaviour, and 3) Tests of policy implications.

Tests of model structure involve comparing the model structure with real-system structure with the model structure not permitted to contradict knowledge about the real-system structure. The model should not give the right answers for the wrong reasons. Further tests in this regard involve developing a plausible hypothesis relating the proposed structure to the problem being addressed. If a hypothesis demonstrating the need for more structure cannot be developed, then the model passes the test.

Behaviour reproduction tests evaluate the model structure by analysing the behaviour generated by the structure. The symptoms' generation test examines whether a model recreates the symptoms of difficulty that motivated the construction of the model (Forrester, J. & Senge, 1980). Behaviour prediction tests examine whether a model qualitatively generates correct patterns of future behaviour. Boundary-adequacy tests consider whether the model includes the structure necessary to address the issues for which it was designed. While it is tempting to try and validate the behaviour of the model developed quantitatively using simulation, it has been argued that while there could be value in simulation, it may be of questionable value if too much uncertainty exists regarding the parameter values. This was summarized by Nuthmann (1994) as "quantification of this model might be plausible nonsense". In such instances the analysis should be restricted to the qualitative level.

In this research, the relationships were identified in the literature and validated qualitatively from interviews conducted with participants. Attempting to quantify the system behaviour noted would have no meaning as the purpose of this RQ2 is to understand qualitatively how not reusing process models impacts business costs.

4.1 Elements of Causal Loop Diagram (CLD)

Drawing on these sources in the literature, the following constructs were conceptualised:

Accessibility

Accessibility is defined as the ease with which a process model can be located by the PM. It is axiomatic that if a model cannot be located, then it cannot be reused. Previous research has confirmed that accessibility is a factor influencing the reuse of process models (Nolte et al., 2016; Shahzad et al., 2010).

Quality (Usefulness)

If a PM perceives that the model is of poor quality, then the PM is likely to create a new model. Previous research has confirmed that perceived quality is a factor influencing the reuse of process models (Nolte et al., 2016; Shahzad et al., 2010).

Model Variants

We define a model variant to be a different process model which is similar or identical to another existing model. Model variants result in "huge amounts of redundant data as the variant models are identical or similar for most parts" (Branco et al., 2014; Cuesta et al., 2015; Hallerbach et al., 2008). Multiple versions of the same or similar processes make it difficult for PMs to identify which is the correct or latest version and increase the likelihood of creating a new model.

Project Requirements Effort

BPM forms part of the business analysis/requirements gathering phase of a project and has become a common activity in large enterprises as it is fundamental to the improvement and automation of business processes (Indulska, Green, et al., 2009; Radgui et al., 2013; van der Aalst, 2013).

Model Reuse

Model Reuse represents how often a specific model has been reused in the repository. In the context of a process repository, this research project is the first to formally define and measure process model reuse in a repository. We define process model reuse in a repository as follows:

"*A process model is reused when that model is referenced by two or more other models in a process repository*".

This variable is the key element as this research question considers the impact(s) of this variable, and the research project has designed a modelling method to increase the Level of Reuse. Chapter 5 describes in further detail how this definition is used in a process repository to calculate the Level of Reuse.

Process Modelling

BPM is the foundation of all automation projects and constitutes an expensive component of such projects (Daneva, 1999).

Project Costs

Commercial organisations need to keep their operating costs as low as possible. We consider the impact of the system dynamics on this key variable because it is of fundamental importance to the organisation. The value of process modelling has been ranked as the top business process modelling challenge (Indulska, Recker, et al., 2009).

Number of Models in the Repository

The literature indicates that an increased number of models in the repository increases the maintenance and governance required (Hallerbach et al., 2008; Rosemann, 2006).

Governance

Governance relates to the management of the process repository and the maintenance of the process models. The literature indicates that more models in the repository result in increased governance (Hallerbach et al., 2008).

4.2 Dynamics of business process model reuse

Drawing on the previous discussion, the causal loop diagram shown in Figure 4.1 was conceptualised to reflect the relationships between elements of the process modelling system.

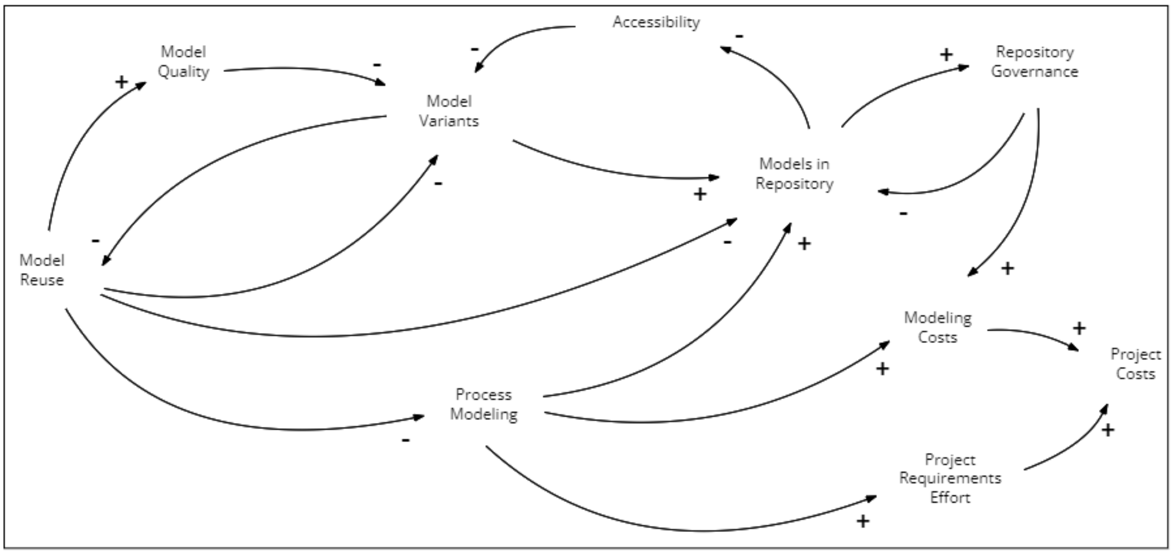


Figure 4.1: Process Modelling Causal Loop Diagram

Each relationship in the diagram is now explained with the corresponding references from the literature and/or quotes from the interviewees.

4.2.1 Causal Relationships Impacting Repository Governance

Figure 4.2: Causal Relationships Impacting Repository Governance

The literature confirms that the more models that exist in the repository, the more governance and management are required (Hallerbach et al., 2008; Indulska, Recker, et al., 2009; Raduescu et al., 2006; Rosemann, 2006; Weber et al., 2011).

4.2.2 Causal Relationships Impacting Models in the Repository

Figure 4.3: Causal Relationships Impacting Models in the Repository

Governance of the process repository involves actively managing the creation of new models as well as existing models. Increased governance will result in the removal of obsolete and redundant models, as explained by P07: "if my portfolio is service, which processes should I go have a look and see? Some are delinquent; some are out of date. We should maintain them. You can't do the maintenance. It's scattered everywhere, it's scattered. And also because if you don't apply RCM, you can't lock those processes, so there's no version control."

As more models are reused, fewer models are created in the repository, and hence, the repository has fewer models. This was confirmed by P01: "you are reducing the size of your asset base because you are reusing."

More variations of models result in increased models in the repository because the PM has created a new model rather than using an existing model (Hallerbach et al., 2008, 2010b; Raduescu et al., 2006; Weber et al., 2011). P05 referred to the example of not reusing existing models and creating a new variant: "you find that some people would go and try to redo that process that's already mapped."

Increased process modelling increases the number of process models in the repository (Hallerbach et al., 2008; Indulska, Recker, et al., 2009)

4.2.3 Causal Relationships Impacting Model Reuse

Figure 4.4: Causal Relationships Impacting Model Reuse

Increased variations of a model in a repository result in less model reuse as the PM is likely to decide that it is too much effort to identify the correct variant and simpler to just build another variant from scratch (Hallerbach et al., 2008; Nolte et al., 2016; Raduescu et al., 2006).

This was confirmed by P01: "what is happening is that certain business areas want a model to be specific for their business area, which then reduces your reuse". P05 described the need for their models to be more specific or have particular terminology: "you find that some people would go and try to redo that process that's already mapped instead of just focusing on what they need to add. Again, that is time".

4.2.4 Causal Relationships Impacting Model Variants

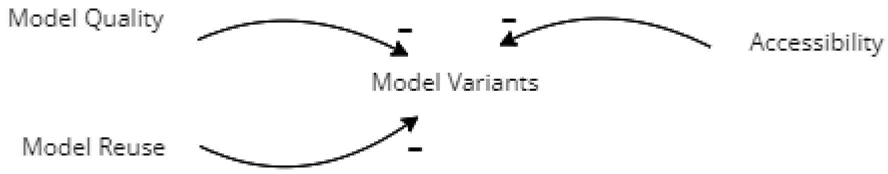


Figure 4.5: Causal Relationships Impacting Model Variants

The perceived quality of a process model directly impacts the intent to reuse the process model because a PM will prefer to create a new model (new variant) if the existing model is not considered useful or if the PM dislikes the semantics used in the current model (Nolte et al., 2013). This theme was evident in the interviews: Multiple respondents shared the need for their models to be more specific or have particular terminology. P04 explained that "you find that some people would go and try to redo that process that's already mapped instead of just focusing on what they need to add. Again, that is time", while P01 described it as "The next person coming along, will say I don't like the verbiage that you have used, and I don't like the structure that you have used. So I'm leaving your processes as a top-down, and I'm creating another version of it with my terminology in it".

Increased process model reuse will result in fewer variations of models. This is supported by literature (Reijers et al., 2009). P01 explained: "You would find in our database five, six versions of the same process and 98% of it the same. So, there was no way that we could manage and control how many models were being created. Each person could create their own version of a model, they could just go copy and paste, and there really was no asset base management if I can say it like that".

If the required model is difficult to locate and access, research has found that the PM is likely to create another model in the repository because of the time and effort spent trying to locate the correct model (Hallerbach et al., 2008). This was confirmed by participant P07, who stated, "So they end up duplicating and triplicating the same process. They can't find them, or if they do, three versions. So they don't know which one to update".

4.2.5 Causal Relationships Impacting Accessibility

Figure 4.6: Causal Relationships Impacting Accessibility

As the size of the repository increases (Models in the repository), locating the correct model to reuse becomes increasingly difficult and reduced accessibility results in more model variants (La Rosa et al., 2010).

The larger a repository is, the more folders and models it contains. This makes finding the correct model (if it exists) difficult unless the PM knows the exact name and location of the model. As P07 stated, "So, they end up duplicating and triplicating the same process. They can't find them, or if they do, three versions. So, they don't know which one to update".

4.2.6 Causal Relationships Impacting Process Modelling

Figure 4.7: Causal Relationships Impacting Process Modelling

As you reuse more models, less modelling is required. As explained by P01: "if you've got a group of 10 people, and you have an existing asset base, and as people start working on projects, and so instead of remodelling every time, and a current view process, you've got the asset base to refer back to. So the, the benefit of that is that you're cutting down on resource usage having an existing base, and included in that is the reusability", while P02 described it like this: "because without the, without the modelling in place, you're always going to be starting on a, on a blank sheet which then takes a lot of, a lot of effort to catch up, which means your sprints are actually unachievable."

4.2.7 Causal Relationships Impacting Project Requirements Effort

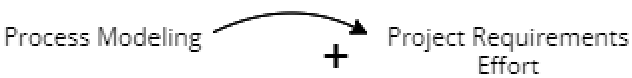


Figure 4.8: Causal Relationships Impacting Project Requirements Effort

Process modelling is part of the project requirements process, and increased process modelling means increased requirements effort. P02 described it when looking at a reduction in the modelling effort: “So the cost on delivering that process model and the requirements, that's where the reduction comes in.”

4.2.8 Causal Relationships Impacting Model Quality

Figure 4.9: Causal Relationships Impacting Model Quality

Process model reuse also results in improved quality of process models (Aldin & de Cesare, 2011; Awad et al., 2011; Fellmann et al., 2014; Nolte et al., 2013, 2016). As the process models are reused, errors in logic are corrected. The comments of P04 support this: “different people would work on the same diagram and update their diagrams instead of having hundreds of different diagrams that are of the same thing.”

4.2.9 Causal Relationships Impacting Modelling Costs

Figure 4.10: Causal Relationships Impacting Modelling Costs

Process modelling requires resources and incurs costs. This is reflected in the literature (Hallerbach et al., 2008; Indulska, Recker et al., 2009) and again confirmed by P09 looking at a reduction in modelling due to reuse: “it's the reusability of processes. And the time that it's taking to model all these processes is certainly one benefit”.

Large process repositories require more governance to delete redundant models and maintain existing models (Rosemann, 2006). This was confirmed in the interviews with P11 with statements such as “if there's no management of the versions, people start documenting all over. And you won't have one view of a process. So, people go in and they delete certain stuff from the process, or they add certain stuff. Because there's no version control, some of the stuff gets lost. And that, that results in rework. So, control of the processes being worked on is quite important.”

4.2.10 Causal Relationships Impacting Project Costs

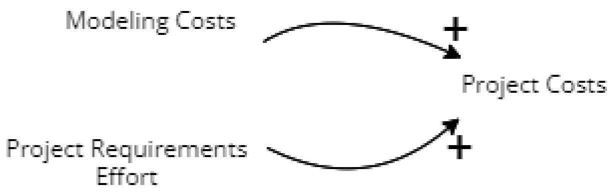


Figure 4.11: Causal Relationships Impacting Project Costs

Increased modelling costs increase project costs (Indulska, Recker, et al., 2009). Explaining this, P01 stated: "So instead of going to do business observations, you've got the current view of the process readily available. So, you're not impacting on the business to go and do observations; you can do it remotely. From a project perspective, it cuts down your project costs. It assists you to deliver much faster".

Business process requirement analysis is considered to be the most expensive consulting service in a reengineering project, confirming that project process costs are impacted by project requirements (Daneva, 1999), and as positioned in by P01: "when you start a project, and you've got an existing database, the time it takes you to obtain the relevant business information is cut down".

4.3 Application of System Dynamics to Process Modelling

Demetis and Lee (2016) proposed six requirements that a systems theory should meet. In this section, we consider each requirement and discuss how it is met by the system described in Figure 4.1.

Requirement to recognize that "the whole is more than the sum of the parts". Looking at Figure 4.1, the elements could be analysed individually, and that would provide insight into the elements on their own, e.g. Models in a repository, Model Quality, etc. However, it would not reveal the relationships that exist between such elements and the resulting overall behaviour of the process modelling system as depicted:

1. Increasing numbers of models in the repository are associated with an increasing number of model variants (Indulska, Recker, et al., 2009; Raduescu et al., 2006)
2. An increase in the number of model variants is associated with an increase in the number of new models because it can become too difficult to determine which variant should be reused (Hallerbach et al., 2008)
3. Increased modelling increases the number of processes in the repository (Hallerbach et al., 2008)

4. Poor quality models cannot be reused, resulting in additional process modelling, which once again increases the number of models in the repository (Raduescu et al., 2006; Saarsen & Dumas, 2016). Consequently, the system is displaying behaviour that cannot be determined by analyzing the individual components.

Requirement to recognize “goal seeking”. In the context of the system depicted, the goal of the system is to reach a steady state where the number of models created is in balance with the number of models being deleted. However, this balance is dependent on the models in the repository, their quality, and the number of variations of the same process model.

Requirement to recognize the “transformation process” by which a system “must transform inputs into outputs” in order to attain its goal. The system depicted takes project process requirements as inputs and transforms them into project costs. These requirements would be based on the scope of the project and the as-is processes. The transformation may be as simple as finding an existing model in the repository, updating an existing model, or creating a new process model. The nature of the transformation is dependent on what already exists in the repository as well as its quality.

Requirement to recognize “self-reference” and “autopoiesis”. The system depicted in Figure 4.1 is self-referential in that the process modelling element feeds back into the process repository element. This feedback loop has the potential to turn into a vicious cycle if the creation of process models reinforces further creation of process models via the process repository, model quality and model variants elements. The system displays autopoeisis in that the system generates elements of the system itself (process models), which in turn may influence the behaviour of the system further, as discussed.

Requirement to recognize the “system/environment” distinction. The process modelling system discussed takes place within an organisation, which can be considered to be the environment. This environment would comprise all business areas within the organisation while the system has been defined from a process modelling perspective. Other perspectives could possibly result in different systems being identified since the system defined is dependent on the perspective of the observer (Demetis & Lee, 2016). Within this environment, the system can acquire self-reference, which begins to display autopoiesis.

Requirement to recognize “communication”. In our process modelling system, we can take a single “requirement” as an example and consider it an “utterance” initiated by a project. This requirement contains further information (possibly product details, channel details, etc., related to the requirement) that is relevant to other elements in the system. For example, the PMs would use this information (or parts of the information) to determine which process models in the repository are relevant or whether the quality of any relevant models found was adequate. Process modelling can also be considered to be communicating "how the business operates" while the business processes themselves contain the information. In turn, the project process requirements interpret (understand) the information (but not necessarily all of the information) via the interactions with the process repository.

4.4 Consequences of not reusing process models

We now discuss the consequences of not reusing process models by considering four examples of dynamic behaviour illustrated in Figure 4.12.

Example 1

Figure 4.12 illustrates the vicious cycle resulting from reduced model reuse:

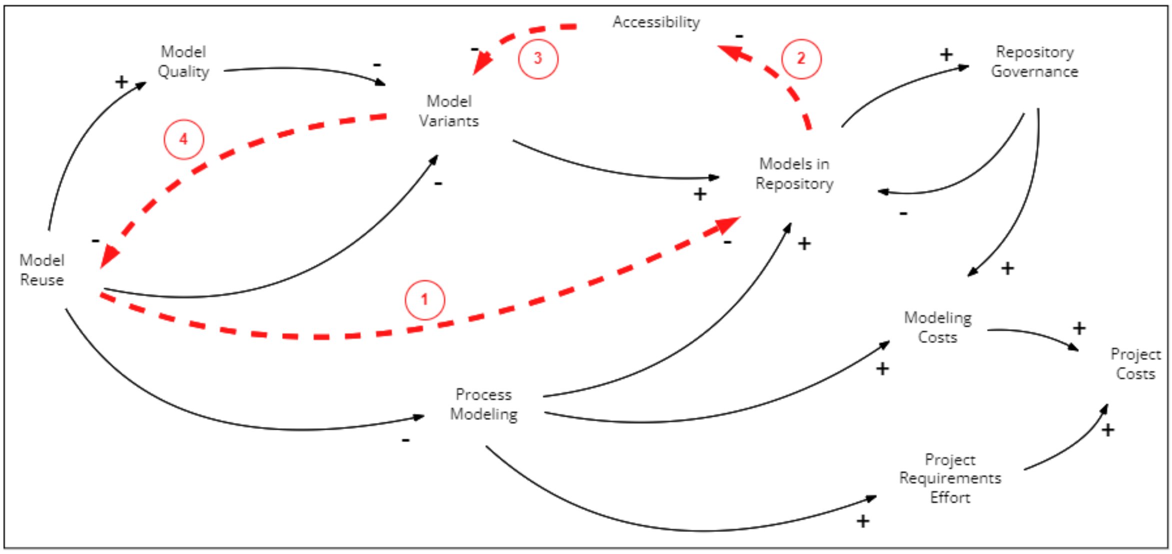


Figure 4.12: Vicious cycle one resulting from reduced model reuse

1. Decreased reuse results in more models in the repository
2. As the number of models in the repository increases, it becomes more difficult for users to find the correct model (Accessibility decreases).
3. Reduced accessibility results in PMs preferring to create a new model.
4. Increased variants result in decreased reuse of process models

The cycle continues as Model Reuse decreases and the number of models in the repository increases.

Example 2

Figure 4.13 illustrates another vicious cycle resulting from reduced model reuse and model quality:

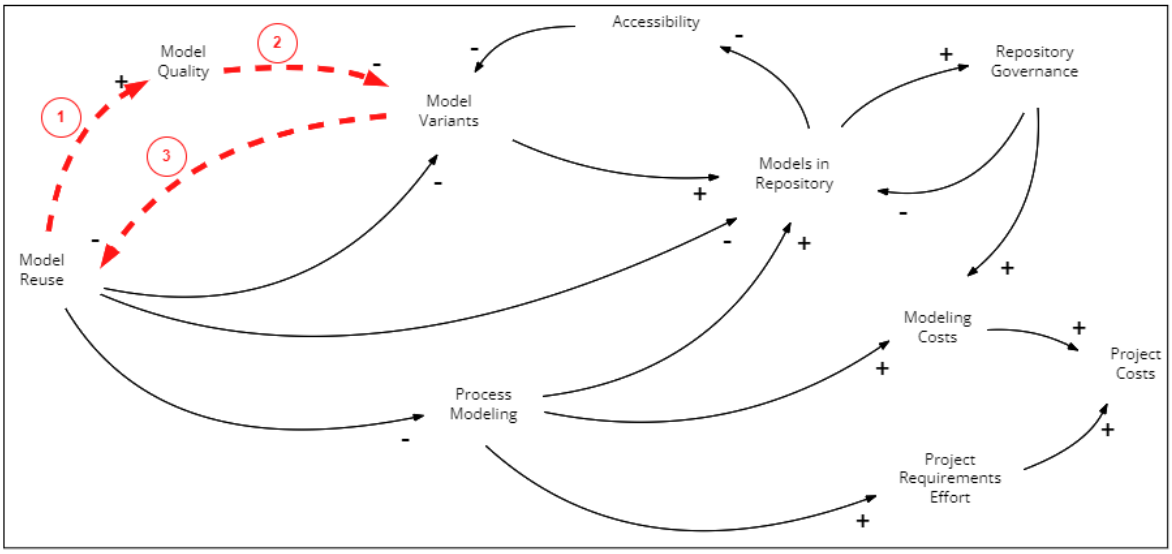


Figure 4.13: Vicious cycle two resulting from reduced model reuse

1. Reduced model reuse results in reduced model quality as errors are not corrected.
2. Reduced model quality increases the number of variants because the PMs are not comfortable with the quality and prefer to create a new variant.
3. Increased variants result in reduced reuse.

Example 3

Figure 4.14 extends Example 1 to illustrate the impact of reuse on modelling costs.

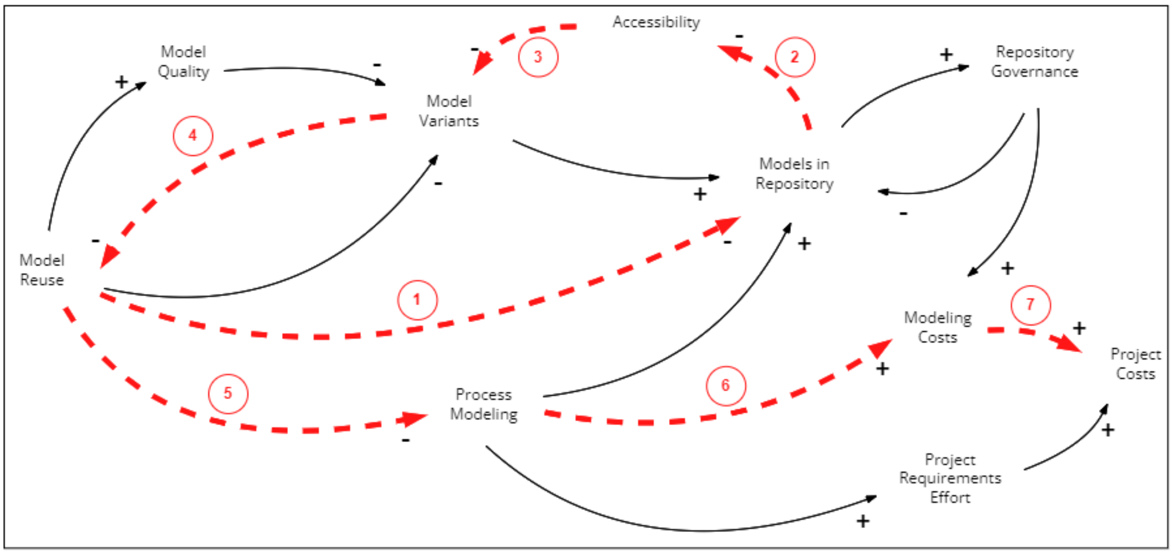


Figure 4.14: Impact of reduced model reuse on project costs.

1. Relationships 1, 2, 3, and 4 were explained in example 1.
5. Reduced reuse results in increased process modelling
6. Increased modelling results in increased modelling costs.
7. Increased modelling costs result in increased project costs.

Example 4

Figure 4.15 illustrates how the lack of model reuse increases governance and costs.

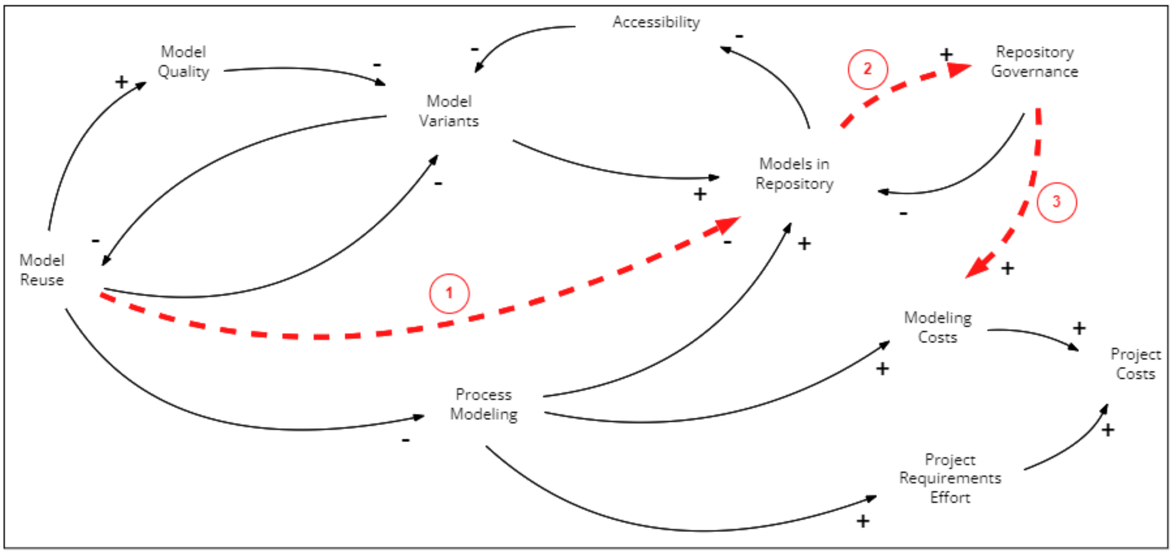


Figure 4.15: Second path to increased project costs

1. Relationship 1 has been previously explained.
2. Increased models in the repository require more governance.
3. Increased governance results in increased modelling costs.
4. Increased modelling costs result in increased project costs.

The process modelling system depicted in Figure 4.13 illustrates how a number of elements (activities in an organisation) are able to combine and, through their interactions, result in a characteristic (ever-increasing project costs) which is not possible to determine from an analysis of the individual elements. In fact, within an organisation, it is unlikely that this characteristic would even be noticed initially as the impact of the feedback loop on the process repository and project costs is subtle and increases slowly over time. This attests to the suitability of using SD in IS and the practical benefits that can be derived from this approach.

Looking at Figure 4.12, an important vicious/virtuous cycle can be observed between Model Reuse, Models in the repository, Model Accessibility, and Model variants. If process model reuse decreases, the number of models in the repository increases, resulting in reduced Model Accessibility. This results in more Model Variants being created, which again reduces Model Reuse, repeating the vicious cycle again. However, if model reuse increases, then the opposite dynamics are established, and a virtuous cycle is set up.

Figure 4.16 illustrates a practical example of how these dynamics manifest in the real world. A PM on a project needs to model a process, proceeds to search in the repository, and finds multiple variations of the process model. After trying to understand which model is the most recent and accurate, the PM gives up due to project deliverable pressures and decides to create a new model, thereby adding another model of the same process to the repository. This cycle will continue to be repeated, resulting in each project unnecessarily creating new variations and the organisation continuing to incur unnecessary costs.

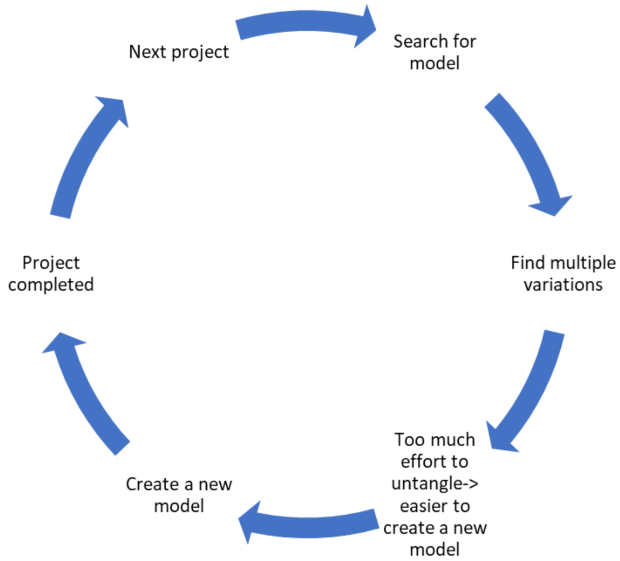


Figure 4.16:A vicious cycle due to lack of process model reuse

4.5 Summary and Limitations

Adopting an SD approach, we have shown how the lack of reuse of existing processes can result in several negative outcomes for an organisation, both in the short term and in the longer term. Firstly, it leads to unnecessary duplication of process models in the repository. The simple act of unnecessarily recreating a process model is unnecessary effort and a cost to an organisation. Secondly, the additional process models in the repository result in a larger repository, thereby increasing the management overhead associated with the repository. This impact is not immediate but evolves over a longer period. Thirdly, multiple versions of the same process in the repository result in additional investigation being required in the future when that process ought to be reused. It is then not clear which version of the process should be used, and time is wasted trying to figure out the answer. This problem may even set up a vicious cycle whereby it becomes easier for the PM to just create another version of the process, which in turn exacerbates the problem. Fourthly, the number of process models in the repository which are no longer used increases, again impeding reuse for the same reasons as just explained. Table 4.1 summarizes some of the negative consequences of not reusing process models.

Table 4.1. Consequences of not reusing process models

Consequence
Redundant process models
Unnecessary costs to the organisation
Increased repository management overhead
Increased process modelling cost
Increased project requirements' cost
Further impediments to process model reuse
Inconsistent model quality

The reported lack of process model reuse suggests that these dynamics already exist and will generate vicious cycles in organisations as described. Considering that the "Value of process modelling" was identified as the second most important process modelling issue, we argue that the system dynamics identified are a major contributor to this issue (Indulska, Recker, et al., 2009). This can also be considered a response to Alter's lament regarding inadequate, techno-centric analysis and calling for more systems thinking.

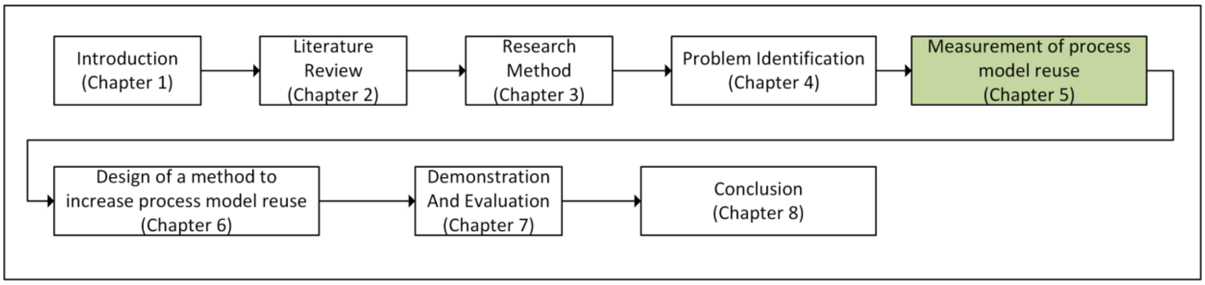
Although systems theory has not seen as much use in IS research as other theories (Smith & Weistroffer, 2016), this research illustrates how different IS research topics and findings can be integrated using systems theory to explain other characteristics (increasing project costs in this example) in an organisation. This principle is practically important as the structure of large organisations is such that although the different departments of an organisation work together, such departments are by their nature focused on what they have been assigned to do and are not aware of the "systemic" characteristics arising from the interactions, particularly when such systems are self-referencing and autopoietic.

These findings are not without their limitations. The conceptual model presented was developed based on the literature and experience of the researcher, and then validated by interview participants. However, these are not the only relationships related to process model reuse which may exist, and depending on the perspective of other researchers, further study may refine the role and importance of such relationships in process model reuse. Furthermore, the real-world validation of the elements and relationships identified is based on a single financial services organization. Other organizations may identify different elements and relationships.

There are costs associated with process model reuse such as training, identifying impacted models, additional model quality checks, and particularly, the creation of the first process model which must be “built for reuse”. On the surface, these add to the cost of the process modeling work on a particular project. Project Managers, focussed on their own project deliverables, resist the overhead associated with these reuse activities. The project managers budget for the additional overhead of recreating the model from scratch but resist the costs of building a model for reuse. The organisation continues to incur the cost of recreating the same process models, project after project. These are not just process modelling costs: they include not only the cost of the modeller, but also the cost of time that SMEs must spend providing process information (which already exists in the repository), the cost that inferior/incorrect process models (i.e. information) have on the project later in the project lifecycle, and the cost of unintended consequences (e.g. when the impact of the project on another process is not recognised early in the project lifecycle). Each organisation must trade-off the benefits of model reuse against the cost of model reuse.

The negative impact on an organization of not reusing complete process models opens the following potential avenues for future research: 1) developing methods to increase complete process model reuse, and 2) developing economic models of process model reuse. Taking up the challenge of the first suggestion, this research project designed, demonstrated and then evaluated a method to increase the reuse of complete process models in a repository.

5 Chapter 5: Measurement of Process Model Reuse



In section 2.4.5 the use of fragments (parts of process models) was discussed. This research project differentiates between the reuse of process fragments and complete process models, and focusses on the reuse of complete process models. In this context a sub-process or component (e.g. a Lego block) is a process model while a fragment forms part of an existing process model. These components can then be reused in multiple processes if designed correctly (the equivalent of a service in the software domain). The literature and this research confirmed that only approx. 10% of process models are reused (can be considered a component). Reusing these sub-processes / components will reduce the cost of process modeling. The purpose of this research then is to increase the level of reuse of sub-processes / components in a process repository

It stands to reason that if we are designing a modelling method which increases the reuse of process models, then it will be necessary to measure the reuse of the models both before and after the new design has been implemented. While quantitative measures of software reuse are described in the literature (Frakes et al., 1996; Poulin, 1997), no such corresponding measures relating to the reuse of complete process models were found. A measure which can determine the level of reuse needed to be designed and validated. This measure could then be used to assess the effectiveness of the proposed method. Accordingly, the problem statement is the research question:

RQ3: How can the level of reuse of complete process models be measured in a repository?

This section describes the development and validation of a quantitative measure of reuse of complete process models.

A deductive approach to answering this research question was adopted because deduction involves the development of a theory, which is then subjected to a rigorous test (Saunders et al., 2023).

The method proposed was designed by drawing on the measurement of software reuse in the application development domain to develop a measure (as elaborated in Section 5.2). The number of times each process model is referenced by other process models (as elaborated in Section 5.4) was determined quantitatively by exporting the data from the repository and then validating the proposed measure.

A real-world process repository was used to guide the design and evaluation of the measure. The model was then evaluated in its context using the validation framework of Fenton, Kitchenham, and Pfleeger (N. E. Fenton et al., 1995). Theoretical validation was conducted by assessing each measurement item against the criteria required by the framework. Empirical validation was conducted by confirming that the model generated the same results for attribute values which could be confirmed by manually inspecting the process repository.

This cycle adopted a Design Science Research (DSR) methodology and, accordingly, a pragmatic philosophy. Design science research is considered an appropriate approach because the purpose is to develop an IS artefact (a new method to calculate the Level of Reuse), and it provides a framework that can be used for applied IS research (March & Smith, 1995; Nunamaker et al., 1990). DSR is concerned with developing or improving artefacts (constructs, methods, models, and instantiations) which are of use to society (Gleasure, 2015; Hevner, March, Park, & Ram, 2004; March & Smith, 1995; Vaishnavi & Kuechler, 2004)

A pragmatic epistemology is also adopted in this research project. The reality (knowledge) is neither measured directly nor interpreted (IS artefacts have no inherent truth value), but instead, the “truth value” lies in statements about their efficiency and effectiveness (Iivari, 2007).

The DSR methodology proposed by Peffers ((Peffers et al., 2007)) was used for this cycle, which involves problem identification, objectives of the solution, design and development, demonstration, and evaluation. It should be noted that this constitutes the development of a different artifact, also using DSR methodology, but within the DSR methodology used to answer the main RQ.

The answer to this research question is prescriptive in nature because it describes a method to measure the level of reuse of complete process models in a process repository.

5.1 Solution Objectives

In DSR, the objectives of the solution would typically emerge from current research. In the context of this research project and the research organisation, the following objectives were identified for this cycle:

- The solution must design a quantitative measure for the LOR of process models in a repository that can quantitatively compare the current and proposed modelling methods.
- The calculation of LOR must be automatable because the number of models in the repository makes it impractical to determine manually.

5.2 Design and Development

In this section, we first provide a summary of the terminology used, then define reuse and how to measure the reuse of one model. Finally, we draw on the software development domain and the principles used to measure the reuse of software, to develop a measure of the level of reuse of process models.

5.2.1 Terminology

A summary of the terminology used in this section is described in Table 5.1. It should be noted that this terminology is based on the terminology and definition used in the ARIS Toolset. The details of these terms are explained in Appendix 1, Appendix 2, and Appendix 3.

Table 5.1: Summary of terminology

Term	Description
Process model	Depicts the sequence of actions or activities that must be executed to achieve a particular outcome.
Function	Represent actions or activities in a process model. The Function itself is an object in the repository.
Occurrence	A Function appears in a process model as an occurrence. It can have multiple occurrences in the same model. Each occurrence is depicted by a symbol, which determines how it appears in the process model. The same Function can be represented by different symbols.
Symbol	Determines how a Function appears in a process model. The appropriate symbol depends on the modelling notation used, as well as the context.
Parent / Connected Object	An activity (i.e., a Function in a process model) that decomposes into a process model containing further detail is known as the parent object of that model. It is also known as the Connected Object of that model.

5.2.2 Defining complete process model reuse

We define the use of a complete process model in another model to be the occurrence of the parent Function of the model in another model, such as the occurrence of F2 in M1 (Figure 5.1). This means that M2 has been used in M1.

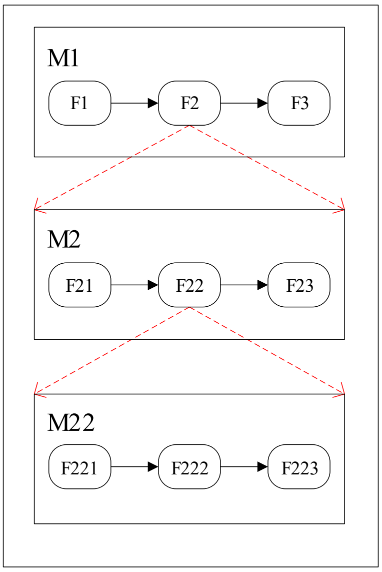


Figure 5.1. Business Process Architecture hierarchy

However, the first use of a model cannot be classified as reuse, and accordingly, a model can only count as reused if it is used more than once. Therefore, we define complete process model reuse as follows:

Complete process model reuse is the occurrence of the parent object of a model in more than one model in the repository, and such occurrence does not act as a 'cross-page connector' to the next process model.

5.2.3 Developing an equation for complete process model reuse

We now develop an equation which can be used to determine how many times a single model has been reused.

Let:
k = process architecture level

m_k be the process model *m* at architecture level *k*

t_k be the number of models in the repository at architecture level *k*

t_{m_k} be the number of activities in model m_k

p_{tot} be the total number of parent objects in the repository

$A_{m_k} = \left\{a_{m_k,1}, \dots, a_{m_k,t_{m_k}}\right\}$ is the set of activities occurring in model *m* at architecture level *k* (i.e. in model m_k)

t_{m_R}= number of process models in the repository

Complete process model reuse occurs when an entire process model (m_k) is abstracted into a single activity ($a_{m_{k-1}}$) in another process model $m_{k-1,i}$, one architecture level higher.
Referring to parent-child terminology, which is easier to understand:

$a_{m_{k-1,i}}$ is the parent activity of model $m_{k,i}$

We can say that $a_{m_{k-1,i}}$ occurs in model m if $a_{m_{k-1,i}} \in A_{m_i}$

And the count of process models which contain a reference to model $m_{k,i}$ will be:

$$C(m_{k,i}) = \sum_{k=1}^{k=k_{max}} \sum_{p=1}^{p=p_{tot}} \sum_{t=1}^{t=t_k} \left| \{a_{m_{k,p}}\} \cap A_{m_{k-1,t}} \right| \quad \text{(Eq. 4)}$$

Where $\{a_{m_{k,p}}\}$ is a single-element set consisting of the element $\{a_{m_{k,p}}\}$ only.

$\{a_{m_{k,p}}\} \cap A_{m_{k-1,t}}$ is the intersection of $\{a_{m_{k,p}}\}$ with the set of activities occurring in model $m_{k-1,t}$. Then $|\ \ |$ denotes cardinality and will evaluate to 1 if $a_{m_{k,p}}$ occurs in $m_{k-1,t}$.

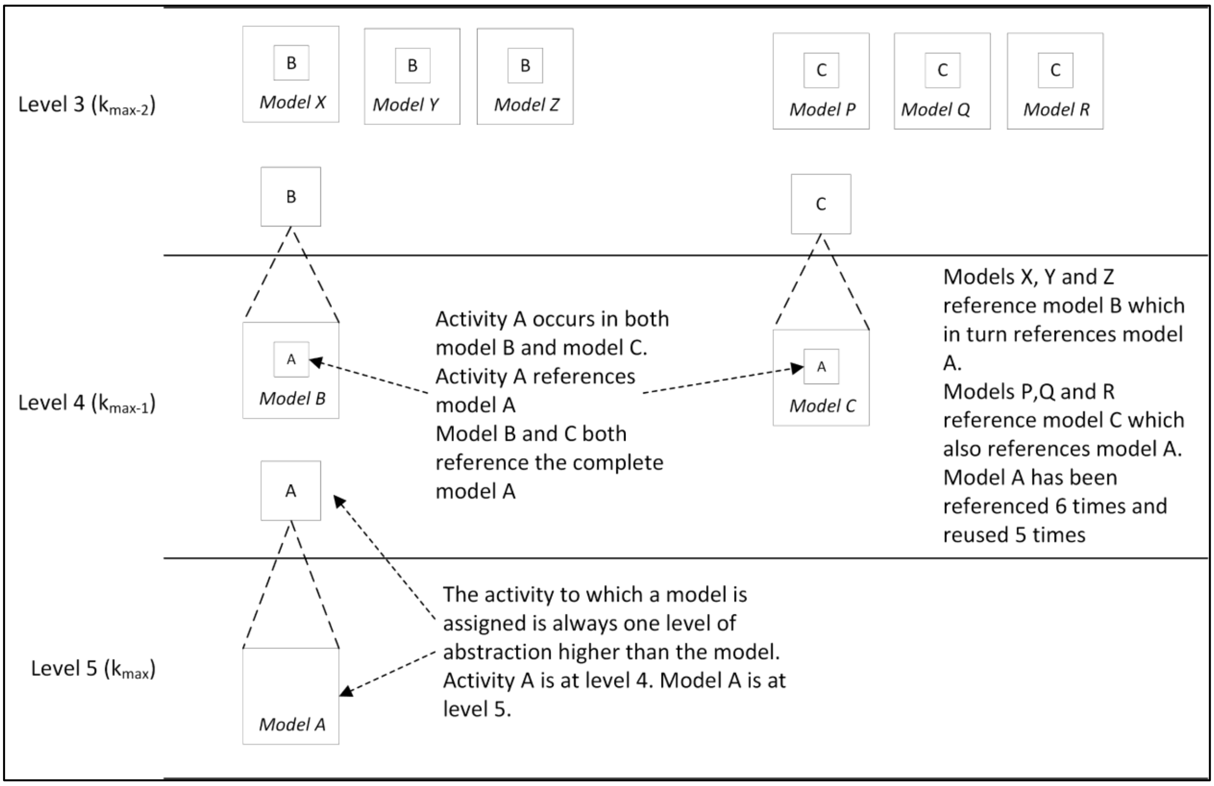


Figure 5.2. Process Hierarchy

5.2.4 Defining what to measure

Reuse Level. For the measurement of the reuse of process models in a repository, we only measure the internal reuse by ensuring that only process models created by the organization are considered. Drawing on the de-facto standard for measuring software reuse (Poulin, 1997), we accordingly define the reuse level as follows:

Process Reuse Level = M / L (Eq. 5)

Where:

M = the number of models in the repository that are used more than once. Using Eq. (4), the number of times each model is used is calculated. Referring to Eq. (2), we set the threshold value for process model reuse (ITL) to be at least 2. A process model can only be considered reused when it is used at least twice.

L = the total number of models in the repository.

How do we handle complexity? While in the measurement of software reuse, it has been proposed that the lower-level item is weighted based on its size (e.g., number of lines of code) (Frakes et al., 1996), we argue that this approach should not be used to weight process models when measuring reuse as frequently two processes with the same purpose may vary significantly in terms of size. For example, obtaining the approval of a customer can be both a simple process (the customer taps on "I approve" in a mobile app) while it could also be a complex process (the company must send a driver to visit the customer to obtain an approval signature on a piece of paper). Two such processes should be equally weighted. Accordingly, we do not provide for complexity when measuring process model reuse.

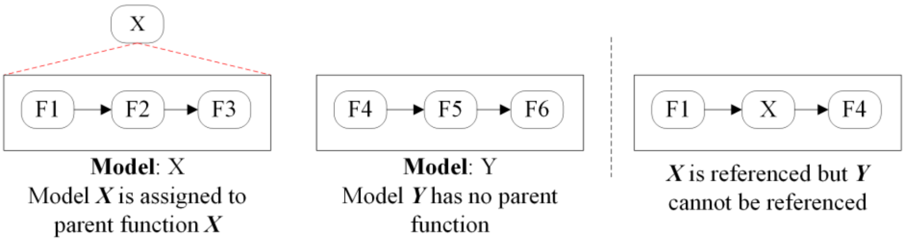


Figure 5.3. Counting process model references

What do we count? An activity in a process model can be considered to be the equivalent of a line of code (LOC) in software (Gruhn & Laue, 2006). At the lowest level of granularity in a process model (e.g., in a level 5 process model in a 5-level process architecture), an activity does not decompose into any further detail. In ARIS, activities are represented using Function objects, which, when combined with other object types and connections, make up a process model. Every process model can be (but does not have to be) linked (assigned) to another Function object, which in turn can form part of a different process model, and in this manner, the process hierarchy is constructed (Figure 11.4 and Figure 5.2). Only Functions with a process model assigned to them (Figure 5.3) can contribute to the reuse count. A process model that is not assigned to a Function (e.g., Y in Figure 5.3) cannot be reused because no mechanism exists by which to reference that process model.

We only consider models which depict process flows (EPCs, rowEPCs, column EPCs, BPMN Process diagrams, and BPMN Collaboration diagrams) to measure reuse in the repository used for the proof of concept. The number of references to a model must be greater than 1: A model which has only been referenced once has not been reused and, therefore, must not be counted - referring to Eq. (5), we set the threshold level (ITL) at 2 when calculating M. We only count a process model as reused once: whether a model is referenced twice or ten times, it is counted as reused only once.

What don't we count? Certain modelling notations (e.g., EPC) make use of a symbol known as a "Process Interface", which is the link to the next process in the sequence (Figure 5.4). Occurrences of Functions as a process interface symbol are excluded from the count because they are effectively just "pointers" to the next process in the sequence, and this does not represent the reuse of a model in a higher-level process. From an architectural levelling perspective, a process interface symbol is one level of abstraction higher than the rest of the Functions in the same model. Figure 5.4 (a) shows the L4 process in which B follows A, C follows B, and D follows A. Figure 5.4(b) shows the details of process A and the link from process A to process B can be seen. The reference to process B in Figure 5.4 (a) is counted as a reference but not the reference to process B in Figure 5.4 (b).

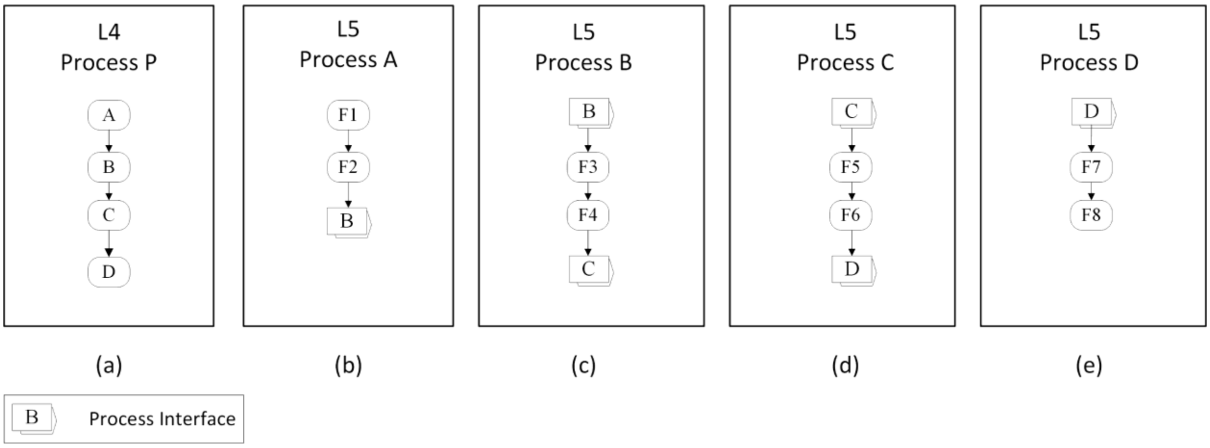


Figure 5.4. Process interface example

A model which has only been referenced once has not been reused and, therefore, cannot be counted as reused. Referring to Eq.(5), if we set the threshold level at 2 (ITL=2 for IU in Eq. (Eq. 2)), then the model is not counted as reused until it has been referenced at least twice.

Fragments of processes: A Function which is not decomposed into a lower-level process model (i.e., more granular detail) is not counted because there is no associated process model.

5.3 Validation of the measure

Validating a measure is the process of ensuring that the measure is a proper numerical characterization of the claimed attribute (Fenton & Bieman, 2015). We adopt the "Framework for software measurement validation" of Kitchenham and Pfleeger (Fenton et al., 1995) to validate the measure proposed in this research. They break a measurement down into various elements and models, which are then used in the measurement process where, for the measure to be valid, both of the following conditions must hold:

- The measure must not violate any necessary property of its elements.
- Each model used in the measurement process must be valid.

According to their approach, it is also necessary to confirm the following:

- Attribute validity: is the attribute in which we are interested exhibited by the entity that we are measuring?
- Unit validity: is the measurement unit the appropriate means of measuring the attribute?
- Instrument validity: is the model underlying the measuring instrument valid and properly calibrated?
- Protocol validity: was an acceptable measuring protocol adopted?

Theoretical validation of the measure confirms that the measurement does not violate any of the necessary properties of the elements or the definition models. Empirical validation corroborates that the measured values of the attributes are consistent with the values predicted by the models. They identify four theoretical criteria which need to be satisfied by a valid measure:

- For an attribute to be measurable, it must allow different entities to be distinguished from one another.
- A valid measure must obey the Representation Condition.
- Each unit of an attribute contributing to a valid measure is equivalent.
- Different entities can have the same attribute value.

Furthermore, an indirect measure is invalid if any of the following conditions hold:

- There is no underlying model to justify its construction.
- There is an underlying model, but the measure can be shown to be invalid in the circumstances when applied.
- The measurement fails a dimensional analysis.
- The measure uses scales incorrectly.
- The measure uses units inconsistently.

Applying the measurement validation framework, the identified entities and attributes that are required are listed in Table 5.2**.**

Table 5.2. Measurement Elements

Entity	Attribute
Repository	Level of reuse (derived measure)
	Number of process models (direct measure)
Process model	Number of times used (direct measure)
	Number of times reused (derived measure)

5.3.1 Theoretical Validation

We validate the measurement theoretically by confirming the validation of each attribute used in terms of the attribute validity and unit validity (Table 5.3) and then the instrument validity and protocol validity (Table 5.4). We also review and confirm the direct (Table 5.5) and indirect (Table 5.6) measure requirements that must be satisfied for the measure to be valid.

Table 5.3. Attribute and Unit Validity

Entity	Measure	Attribute Validity	Unit Validity
Repository	Level of reuse	The level of reuse is an attribute exhibited by the process repository	Percentage (%) is used as the unit of reuse. This is in line with our definition of reuse.
	Number of process models	The number of process models is an attribute exhibited by the process repository.	Count is the unit of measure for the number of models in the repository.
Process model	Number of times used	The number of times a process model is used is an attribute exhibited by a process model.	Count is the unit of measure for the number of times a process model is used.
	Number of times reused	The number of times a process model is reused is an attribute exhibited by the process model.	Count is the unit of measure for the number of times a process model is reused.

Table 5.4. Instrument and Protocol Validity

Entity	Measure	Instrument Validity	Protocol Validity
Repository	Level of reuse	This measure is derived from the number of process models and the number of times each model is referenced based on Eq 4: (Eq. 4)	This measurement is obtained by plotting the cumulative frequency distribution of the number of references to a process model.
	Number of process models	The total number of models in the repository.	This measure is obtained by extracting and analysing model data from the repository. The number of models in the model list is counted.
Process model	Number of times used	The number of occurrences of a parent Function of a model in other process models where the parent Function does not appear as a process interface.	This measure is obtained by extracting and analysing model data from the repository.
	Number of times reused	The number of times a process model is reused is one less than the number of times it was used with a minimum value of zero.	This measure is obtained by subtracting one from the total number of process model uses with a minimum value of 0

Table 5.5. Direct Measure Requirements

Direct Measure Requirement	Attribute	
	Number of process models	Number of times the process model used
Allow different entities to be distinguished.	True	True
Obeys the Representation Condition	True	True
Each unit of an attribute contributing to a valid measure is equivalent	True	True
Different entities can have the same attribute value	True	True

Table 5.6. Indirect Measure Requirements

Indirect Measure Requirement	Attribute	
	Level of reuse	Number of times reused
Underlying model valid	True	True
Underlying model is valid and applied correctly	True	True
Dimensional analysis correct	True	True
Scales used correctly	True	True
Measurement units used correctly	True	True

5.3.2 Empirical Validation

The attributes and measurements were empirically validated by manually identifying and counting process model references in the repository and comparing the results with the results using the measurement protocol. A sample of three models was taken, which, based on the results of the measurement protocol, were referenced 2,3 and 5 times, respectively. These results were confirmed by identifying the three models in the repository and manually confirming that the number of references reported for each model by the measurement protocol was correct (Table 5.7).

Table 5.7. Empirical Validation: Model Count

Model ID	Manual count of model references	Measurement count of model references
1	2	2
2	3	3
3	5	5

The count of the number of models in the repository was also validated manually. The number of process models in three folders (and sub-folders) was counted and compared to the model counts in the same three folders reported using the measurement protocol (Table 5.8). Furthermore, the total number of process models in the repository was also compared to the measurement protocol results and found to be the same.

Table 5.8. Empirical Validation: Model References' Count

Folder ID	Manual count of models	Measurement count of models
1	3	3
2	2	2
3	17	17

5.4 Demonstration

The proposed measure of process model reuse was tested on the ARIS process repository of a large South African financial services company, which has used ARIS as the process modelling tool for the past 23 years. A five-level hierarchical process architecture is used, and the modelling method used within the organization is maintained by an organizational Center of Excellence and enforced using the functionality provided by ARIS. Between 100 and 200 PMs are active at any one time. The size of the ARIS process repository (20000+ models) and the duration of use within the organization make this a suitable repository to use as a proof of concept for the proposed measure. Some of the characteristics of the repository are shown in Table 5.9.

Table 5.9. Repository characteristics

Characteristic	Value
EPCs (EPC, Row EPCs, Column EPCs)	22104
BPMN Models (BPMN Process Diagrams, BPMN Collaboration Diagrams)	2141
Total Number of Models (Control Flow) EPCs + BPMN Models	24245
Other model types	38601
Total Number of Models	62846
Total Number of objects	511354
Functions	185598
Others object types	325756
Total Number of object occurrences	1258832
Function objects	376642
Events	326595
Other occurrence symbols	555595
Number of Folders (ARIS Groups)	13969

Using the ARIS scripting functionality, details of the models, objects, relationships, and symbols were exported into spreadsheets and then imported into Microsoft Access for data analysis. The steps followed to determine the overall level of reuse within the repository were as follows:

1. Determine the number of times each process model is referenced by other process models. Then, using the reuse threshold level, determine how many times the process model has been reused.
2. Determine the frequency distribution of the number of models and the corresponding number of uses in the repository (Table 5.10). It can be seen that 85.2% of process models in this repository are referenced once or less. This can be restated by saying that 85.2% of models in this repository are not reused at all.
3. Using the frequency distribution table, the levels of reuse for different threshold values (ITL) can be determined.

Table 5.10. Model Reference Frequency

Reference Frequency	Cum %
0	33.6%
1	85.2%
2	93.5%
3	95.8%
4	96.9%
5	97.4%
6	97.9%
7	98.3%
8	98.5%
9	98.7%
10	99.6%

The level of reuse of the process models in the repository was determined for different threshold values calculated and are shown in Table 5.11. The threshold values shown in Table 5.11 have the following meanings:

- 0: The process model has no parent object (i.e., it is not possible to reference this process model), or the process model has a parent object which does not occur in any process model.
- 1: The process model has a parent object, which is only referenced once in a process model. Because this process model has only been referenced once, it has not yet been "reused".
- 2-5: The process model has a parent object which has been referenced 2,3,4 or 5 times. Accordingly, these process models have been reused. For example, 8% of process models are referenced twice or more. 2% of process models are referenced four or more times.

Although threshold values of 0 and 1 do not indicate reuse, they have been included because they indicate that most of the models in the repository are not reused at all.

Table 5.11. Percentage of models vs Number of references to models

Threshold (Number of references)	# Models	%	Cumulative %
0	43241	63.6	100.0
1	15759	28.4	36.4
2	3457	4.5	8.0
3	1507	1.5	3.5
4	876	0.6	2.0
5	615	1.4	1.4

5.5 Communication

The paper "Measuring Business Process Model Reuse in a Process Repository" was presented in the Business Process Management Workshops at the BPM 2019 Conference in Vienna (Veitch & Seymour, 2019).

We believe that this is the first research to quantitatively define a measure of the reuse of complete business process models by other process models in a process repository, as well as the empirical measurement of such reuse based on the proposed measure. The measure was tested on a real-world process repository containing more than 35000 process models.

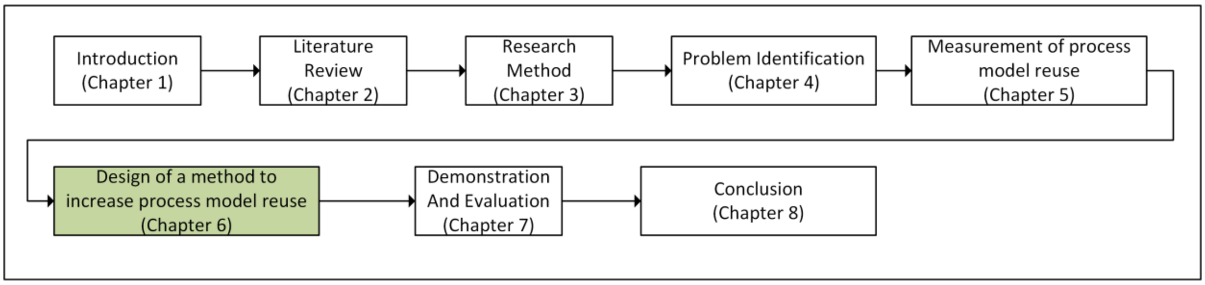
Although ARIS terminology has been used, the measure and approach are not dependent on the repository being an ARIS repository but rather on the structure and relationships of the models and objects. These are dependent on both the modelling tool functionality and the modelling method used by the organization. The proposed measure was tested on an ARIS repository, which is structured as described in Section 5.4. Furthermore, the modelling method and process architecture adopted by the organization where the measure was tested supported the method described in this research.

5.6 Summary

As the number of channels and devices that are used to interact with customers increases, the principle of reuse of complete process models within a repository will likely increase in importance. The current lack of reported measures for complete process model reuse will need to be adequately addressed before new modelling methods can be quantitatively assessed in this regard. Similarly, economic models of process model reuse will need a suitable measure of reuse if such models are to be compared.

For practitioners, the measure provides a tool which enables the reuse of process models in a repository to be measured and managed. The measure can also be applied to different sets of models in a repository to determine, for example, the reuse level per organizational unit carrying out the modelling activities.

6 Chapter 6: Design of a method to increase process model reuse



In this section, a method is designed to increase process model reuse. The required steps fall into one of three areas: “modelling for reuse”, “modelling by reuse”, or “ease of use”.

6.1 Objectives

In Section 4, it was shown that not reusing process models results in increased costs for the organisation. In this section, a process modelling method is designed with the following objectives:

- Increase the level of reuse of process models.
- Reduce the costs of BPM within the organisation.

However, these objectives are high-level, and specific steps need to focus on specific aspects of BPM. Recognising this, three BPM areas which could be improved were identified, as indicated in Table 6.1.

Table 6.1: Sub-Objectives

Focus Area	Sub-Objectives
Modelling for reuse	Design reusable processes
Modelling by reuse	Ensure architectural alignment. Enable a user to navigate through an end-to-end process. Enable users to easily locate required process models.
Ease of use	Correctly manage model and object governance to improve model reuse. Cater for multiple projects impacting the same process.

We turn to capabilities and services as constructs to group similar processes at L3. A capability is provided by one or more L3 services. For example, the capability called “Gather initial client details” consists of one or more L3 services (L4 process models) that gather the initial client information, which can be used to create a client record and authenticate the client. Depending on the scenario, these details could be captured on a mobile device by the client, captured into the system in a branch by a banker, or even scribbled on a piece of paper for later use. From a process modelling perspective, whether these processes are modelled in separate models or combined into a single model with multiple branches depends on the organisational preferences. However, all process models within this capability should:

- Have the same initial conditions.
- Have the same outcome.
- Have the same process model Architectural Hierarchy level (L4).
- Have a parent object of each process in this set, which is a Service (L3 Function).

6.1.1 Capability Identification

Identifying capabilities is an iterative process. While simply naming a capability was often quick, it was found that identifying the process boundaries required detailed consideration of the potential scenarios that might be involved. Using “Authenticate client” as an example, in a multi-product, multi-channel enterprise, the channel and product will impact the level of risk involved in the process and thereby impact which authentication processes are permitted. Authentication requires at least the following:

1. Some client information
2. A process which is executed to authenticate the client.

However, the client information required is also dependent on which authentication process is to be used. Furthermore, the client information required might also be required again later in the end-to-end process when opening an account. If obtaining and capturing the client information is tightly coupled to the authentication process, then that logic will be difficult to use in a different context from authentication. Therefore, in this example, the obtaining of the client information was separated from the detailed authentication process and the capability “Obtain client information” was identified.

Another issue that arose was related to the detailed definition of the process boundaries. Using the same example (“Obtain client information”), it is important to ensure that the boundaries are clear. If the client is using a mobile app and logs in, the email address and password (for example) will constitute the client's data, and the process for authenticating is built into the back-end system. But where is the boundary between these two capabilities? Does “Obtain client information” end before or after the client clicks “Submit” in the mobile app? While there is no absolutely correct or incorrect answer to this question, considering other end-to-end scenarios offers guidance. For example, the question was then asked, “Are there any scenarios where the client data is collected manually, and authentication is carried out manually? A scenario was identified where a client may be wealthy and elderly and not comfortable using a mobile device. Instead, a Private Banker or a courier might physically visit the client, the client signs the appropriate forms, and then the forms are returned to the office to complete the process. In this scenario, the transferring of the data from the client to the organisation is a physical process before the authentication process can be completed on the system. For the “Authenticate client” capability, all the authentication services should have the same boundaries (Principle 1. In this instance, it was decided to exclude the information transfer from the “Authenticate Client” capability and include it in the “Obtain client information capability”.

6.1.2 Capability Definition Template

To ensure a common understanding of the definition of a capability, the following information was recorded regarding a capability:

- Name
- Objective
- Description
- Initial Conditions
- Final Conditions
- Comments
- Example(s)

In the repository, this information was captured into an attribute of the object used to represent a capability. This ensured that the definition of the capability was available to all PMs, and if updated, then the update was immediately available. Furthermore, it provided a single source of truth of the definition and enabled the generation of a catalogue of capabilities using the ARIS reporting functionality.

This section has discussed the principles for designing and creating reusable models (Modelling-for-reuse) by:

- Introducing the construct of a Capability which identifies a group of process models which provide the organisation with that Capability. A Capability should be positioned as an L3 object in the PA. The capability is realised by a service.
- Drawing on the principles of SOA and Service Design and applying them to the domain of process modelling to create reusable process models. These reusable process models are analogous to services in the software application domain and should be positioned at L4 in the PA.
- Describing the process of identifying a capability.

- Defining a Capability Definition Template which is used to clearly define a Capability to minimise any potential confusion regarding a specific capability and how it can be used.

We now discuss the components required for Model-by-reuse.

6.2 Design and Development – Modelling Method

Erol (2016) proposed a conceptual model of “the process of process model reuse” where the modelling process was broken into two stages: 1) Model-for-reuse, and 2) Model-by-reuse. Using this conceptual model to guide the design of EPReS, the design principles and characteristics of reusable process models are presented. While both stages are essential to achieve process model reuse, if the process models are not reusable, then no modelling method will be able to achieve reuse. However, if the process models are reusable, then an appropriate process modelling method will be able to achieve process model reuse. However, while a modelling method might exist, it must also be scalable and practical to sustain the practice of “modelling-by-reuse”. Secondly, a process of “modelling-by-reuse” is presented, where a practical and scalable modelling method is designed. The design of this method required changes to the PA, modelling method, and automation to reduce the manual effort required to correctly perform the model and object management in the repository. The steps identified and corresponding components are illustrated in Figure 6.1.

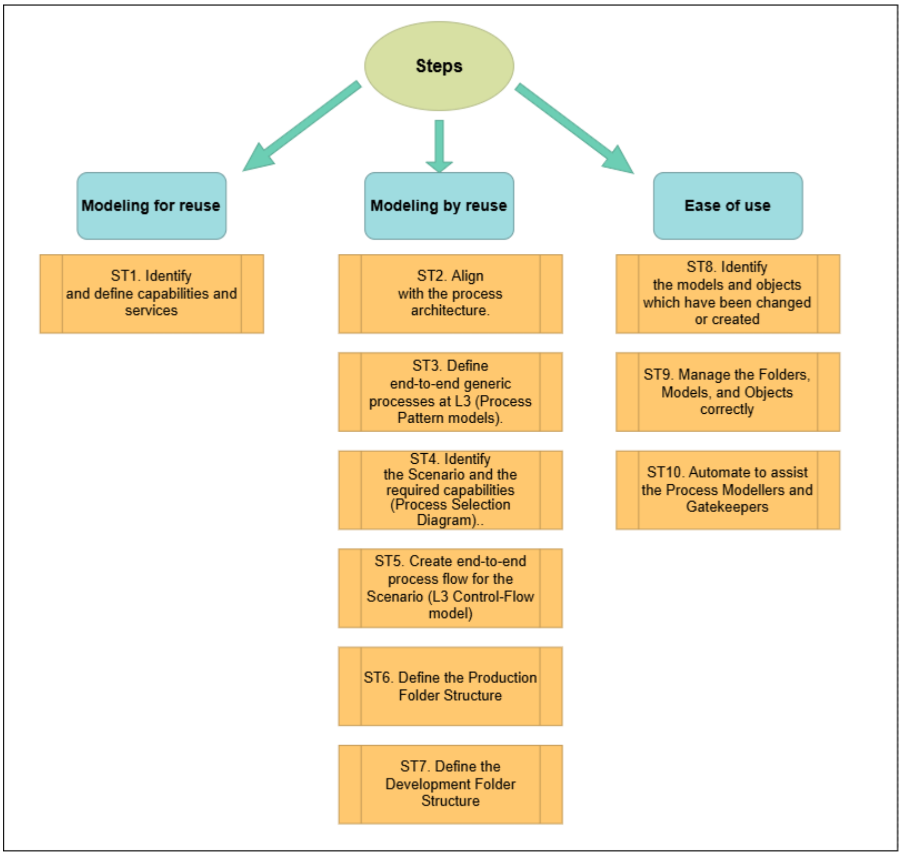


Figure 6.1: Design Steps

The Steps are listed in Table 6.2 and each step is then described in detail.

Table 6.2: EPReS Design Steps

Step	Description	Comment
ST1. Identify and define capabilities and services	Capabilities are "What" the organisation does. Services are "How" the organisation does it. A service is a specific process which enables a particular capability.	Capabilities are used in EPReS to: 1) inform the folder structure in the repository, 2) group services which enable the same capability, and 3) build the Process Pattern Model as explained in ST3. This assists the PM in locating a particular model, and the automation (elaborated in Section 0). Service design principles applied to process modelling enable loosely coupled process models which are reusable.
ST2. Align with the process architecture	The PA is a structured view of how the organisation views its processes.	The PA informs the folder structure that EPReS uses in the repository. Ensuring that the folder structure is rigorously defined facilitates the automation of 1) the creation of new folders, and 2) the automation of model and object management
ST3. Define end-to-end generic processes at L3 (Process Pattern models).	The process pattern is a process model template which applies to a specified grouping of end-to-end process models. The Activities in the Process Pattern Model will be the relevant Capabilities previously identified.	The process pattern model should be constructed based on the identified capabilities, which then 1) expedites the creation of L3 process models (Scenarios), and 2) ensures consistency in the quality of the new models.
ST4. Identify the Scenario and the required capabilities (Process Selection Diagram)	The PSD defines the relationships between the Scenarios, Capabilities, and Services in a Scenario.	The PSD needs to be constructed because the relationships between the Capabilities, Services, and Scenarios are created in this model. This model is essential when setting up a new project and then modelling because it informs 1) which folders are required, 2) which new models are required, 3) which existing models need to be updated, and 4) the creation of the Process Publication Model described in ST9. Furthermore, the initial version of this model can be automated using scripting described in ST10.
ST5. Create end-to-end process flow for the Scenario (L3 Control-Flow model)	A L3 Control-Flow model is an instantiation of the process pattern model described in Step ST5.	The end-to-end process flow for the Scenario being modelled is created based on the Process Pattern Model and the corresponding Services identified in the PSD. The process pattern model described in Step ST3 is the template defining the structure of multiple instances of L3 Control-Flow models. The relationship in and between the Process Selection Diagram, and the Process Pattern model enables the automated creation of the L3 Control-Flow model (described in this ST10)

Step	Description	Comment
ST6. Define the Production Folder Structure	These are the folders in the Production Environment (Prod), which contain the latest official version of each process. These models reflect the current practice in the business.	While the Production folder structure is based on the process architecture, the L4 folder names are the same as the Capability names. Also, the sub-folder names under the L4 folder are the same under all L4 folders. When publishing a project process model, it needs to be moved (or copied) from the Development project folder to the correct Production folder. The corresponding objects also need to be moved to the correct Production folder.
ST7. Define the Development Folder Structure	These are the folders in the Development Environment (Dev) which contain the project version of each process. These models reflect the future practice in the business once the process has been published.	The Development folder structure should include a reference to the project that requires process modelling. This enables multiple projects to work on the same process simultaneously. There is a clearly defined relationship between the Development and Production folder structures. This enables the creation of project folders to be automated, as well as the publication of project process models.
ST8. Identify the models and objects which have been changed or created	The Process Publication model defines: 1) which process models need to be published, 2) which models are new models, and 3) which models are updates to existing models.	The actions required to move a new model from Development to Production are different to those required when updating an existing model. Each model in a project needs to be identified as “new” or “updated” so that it can be correctly treated. The Process Publication model is essential when publishing the process models for a specific project because it defines which models are considered to be new, which models are considered to be updates, and which Production models will be updated with which Development models from a project.
ST9. Manage the Folders, Models, and Objects correctly	Folder, Model, and Object Management describes how the Folders in the repository are named based on the Process Selection Diagram and the rules and method used to create new models, copy models, and correctly move repository models and objects from the Development folder structure to the Production folder structure.	The management of folders, models, and objects has to be correctly executed. These rules and the method adopted are critical to the overall success of the new modelling method. Due to the complexity inherent in model and object management, defining a clear set of rules enables the automation of folder, model, and object management.
ST10. Automate to assist the PM	Describes the automation of various steps in the process modelling lifecycle. In this context, automation means the running of a script by the PM/GK which automatically executes	Step ST1 -St9 need to be supported by automation. This eliminates the potential for human error by the PM/GK. (e.g. accidentally spelling the name of a folder slightly differently).

Step	Description	Comment
	one or more tasks which would have been manually carried out by the PM/GK.	Such scripts are used extensively throughout the process modelling lifecycle.

To further illustrate how the different components of this design impact the level of reuse of process models, we link each step to the determinants of process model reuse as presented by Nolte et al. (2013). The design process for the new modelling method is now described, and how such design steps are supported by the determinants of process model reuse reported in Section 2.3.

The design of the new modelling method evolved over three DSR cycles, as illustrated in Figure 6.2. During the first cycle (2014-2015), the focus was on how to design reusable processes, the models and objects required, and the folder structure. The second cycle (2015-2017) focussed on automating some of the required steps, which in turn required an additional model type and changes to the folder structure. In the third cycle (2017-2020), the folder structure evolved further based on recommendations from the GK and PMs. Automation of manual GK/PM tasks was further enhanced in the third cycle.

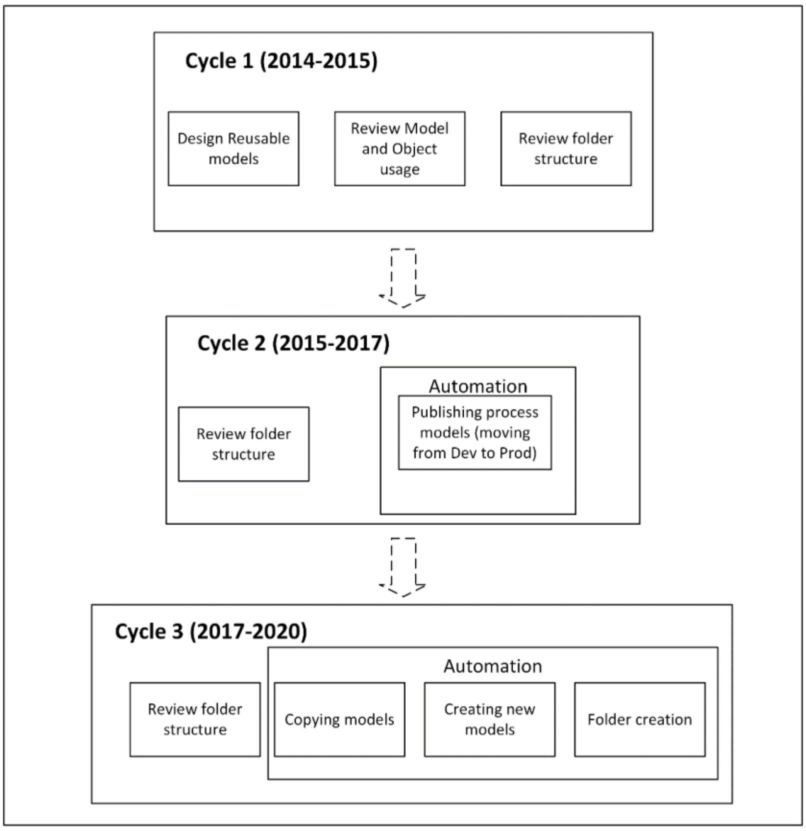


Figure 6.2: DSR Cycles

6.3 Modelling for reuse

To increase the LOR, the process models created must be reusable. For the design of reusable processes, we turned to the concept of service orientation and service design, which has become common in the domain of application design. Services evolved in the application development domain specifically to improve productivity in software development and maintenance. Business process models can be considered as a programming abstraction for business users where the commands are not executable but syntactical in nature (Fantinato et al., 2012; Zhao et al., 2006).

Accordingly, we argue that this makes the principles and concepts related to Service design and SOA appropriate when designing reusable business process models. We also use the concept of a "Capability" to classify different types of services. Capabilities were introduced in Section 2.10.3. A capability defines "What the business does" as opposed to "How the business does it". In the context of this research, a capability is a classification that is applied to one or more services. For example, while a financial services company must have a "Client Authentication" capability ("what it does"), there could be multiple methods ("how it does it") to authenticate a client. The authentication methods used would be the service(s) which enable the capability. This is illustrated in Figure 6.3.

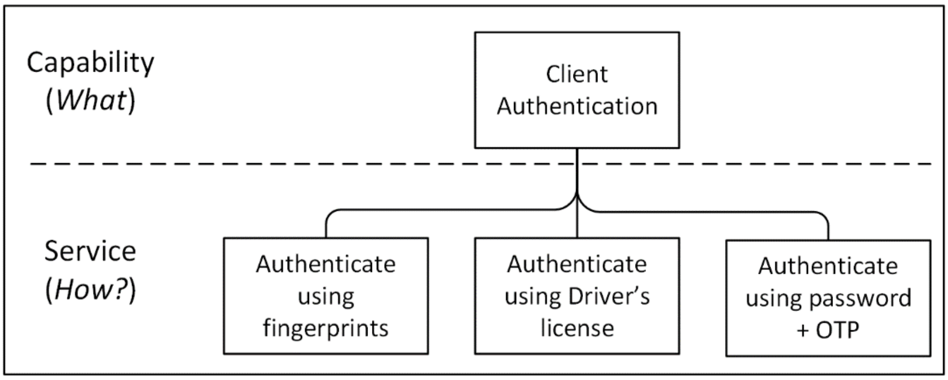


Figure 6.3: Capability vs Service

Drawing on the eight principles of Service design described by Erl (2008), each principle and its application to reusable process model design was considered. Table 6.3 lists these principles and the interpretation of how each was applied.

Table 6.3: SOA principles applied to reusable process model design

#	Principle	Description	Application for Process model design
1.	Services share standardised contracts.	A service contract describes the purpose and capabilities of a service. This is the information that is exposed to external consumers and includes a technical interface or one or more description documents.	We take this to mean that the interfaces for a set of services which enable a specific capability are standardised. For example, the initial and final conditions must be defined. If these services are defined as different process models, then these process models become interchangeable in the process models at a higher level of abstraction (Plug-and-play)
2.	Services are loosely coupled.	Coupling can be thought of in terms of "dependency". Tightly coupled services cannot execute independently of each other, whereas loosely coupled services are able to execute independently of each other. In contrast, loose coupling means that a service can be improved without impacting the service consumers as long as the interface (contract remains unchanged).	The process model must not explicitly reference either the preceding process or the succeeding process at the same architectural level. To do so will tightly bind this process to the referenced process and create a dependency.
3.	Non-essential service information is abstracted.	In this context, abstraction means hiding any information from consumers that they do not require to use the service effectively. This means that the internal workings of the service can evolve and be improved as long as the contract and purpose of the service remain unchanged.	Consumers of the process only need to know the outcome(s) of the process and the required input conditions. Providing additional unnecessary details will mean that such details cannot be changed without impacting the consumers.
4.	Services are reusable	Services are supposed to be reusable. This means that the intention is for the service to be used by multiple consumers in different solutions.	Multiple processes in different scenarios must be able to use this process without making any changes to the process model.
5.	Services are autonomous	Something that is autonomous has the power to make its own decisions. In the context of a service, autonomy means that the service is independent of its surroundings and is able to execute independently from outside influences.	Process models must only contain activities at the same architectural level as the model.
6.	Services minimise statefulness	In application development, state often refers to the condition of an instance of a business object, say an account or a transaction, at run-time. An account may be open, dormant, or active, for example. However, maintaining state information requires resources (memory, CPU, etc) and in applications with many concurrent users, maintaining this information for each user can impact the scalability of the solution. Minimising the statefulness of a service is therefore important for its scalability.	Not applicable This principle was not used in the design. The implications of "how" the proposed process would be technically implemented are not the PM's responsibility, and accordingly, this principle is not applicable.

#	Principle	Description	Application for Process model design
7.	Services are discoverable	Services are discoverable resources within an enterprise and can be referenced when discovery queries are issued.	A process that cannot be located (discovered) cannot be reused. The design of the architecture and repository folder structure must facilitate locating the appropriate process model. It must also facilitate the automation of tasks manually performed by the PMs and GKs.
8.	Services are composable	Services can be assembled into a composition through which they are coordinated to collectively solve a problem. Each service can participate as a building block in another higher-level service composition, thereby resulting in a hierarchal structure of services.	This principle underpins the design of the reusable processes. End-to-end L3 processes must be composed of L4 processes, which in turn must be composed of L5 processes.

6.4 Modelling-by-reuse

The process of modelling by reusing process models is now discussed. This touches on several aspects of the modelling process:

- The location of the process models (both Production and Development versions)
- The folder structures within the Production and Development environments.
- The process of copying models and objects from Production to Development
- The process of moving models and objects from Development to Production

6.4.1 ST2: Align with the business architecture

As discussed in Section 3.5.2, the top 2 levels of the PA were defined by the Enterprise BA team, and Levels 3, 4 and 5 were the responsibility of the modelling teams. Following the architectural guidelines, the L2 Functions (L3 process models) represent an end-to-end process from the client's perspective.

Figure 6.4 illustrates a section of the PA. Within the "Core Value-Chain" (L1), there is another group of processes identified as "Manage Sales of Products and Services", and within that group of processes is another set identified as "Sell Products and Services". Sell Products and Services is generic and covers the provision of all products and services to the customer across all channels. Considering the number of account types, client types, and possible channels, it was decided to introduce another classification based on the process pattern that would be applicable. This would enable sales processes which followed the same structure to be grouped together. Examples of such further classification are:

- Sell Transactional Products

- Sell Investment Products

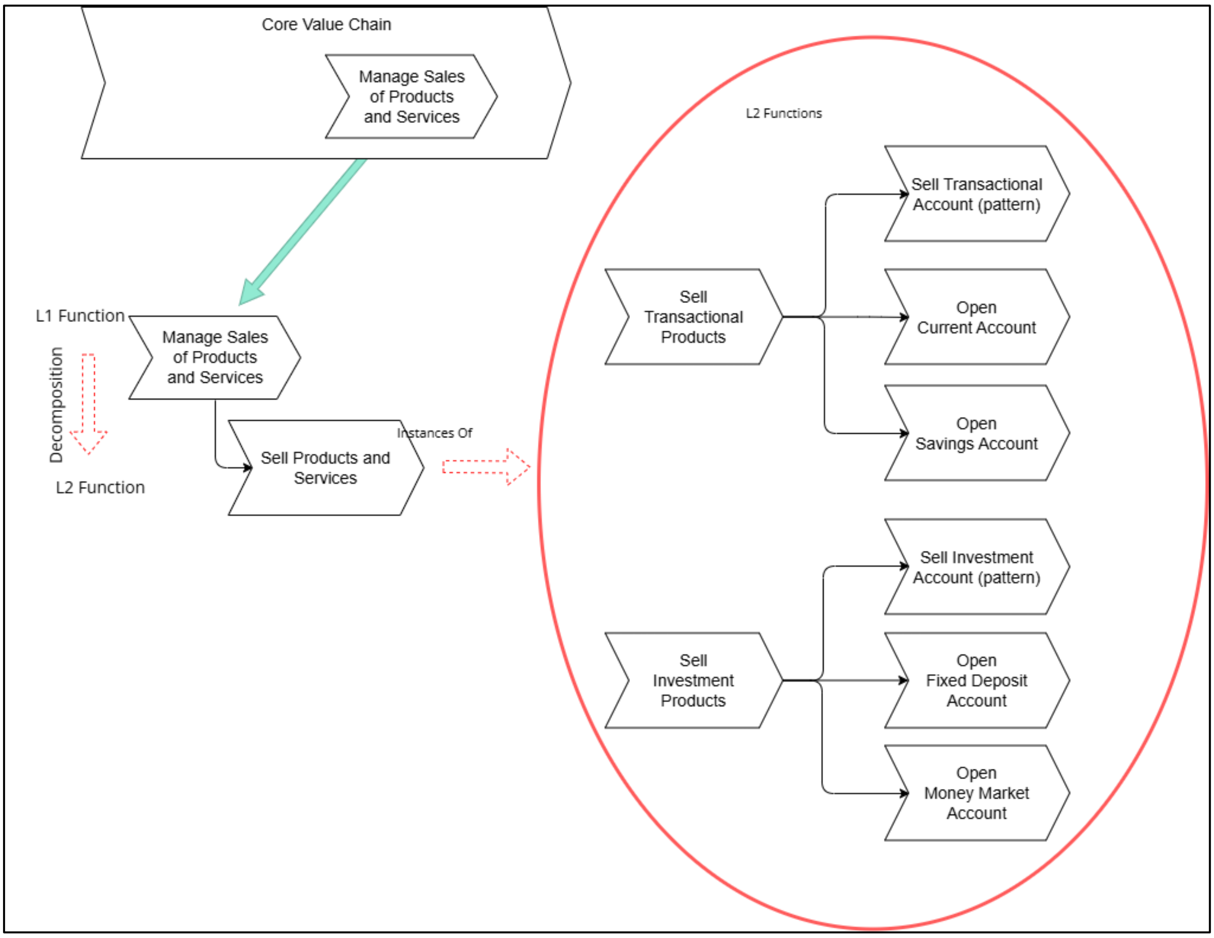


Figure 6.4: Level 1 and L2 Process Architecture

This section discussed the alignment with and integration with the high-level PA. The high-level PA, specifically Level 2, provides the anchors for the L3 Process Pattern Models (Step ST3) and L3 Scenario models (Step ST5).

6.4.2 ST3: Define end-to-end generic processes at L3 (Process Pattern models)

In a multi-product, multi-channel environment, there are processes which are similar in structure. When opening an account, for example, the identification and authentication of a client will be the same when carried out in a branch, whether the account is a current account or a savings account. However, the process for capturing the account details might be different based on the account type. By abstracting the process further, it is possible to construct a process model and treat the capabilities as the activities in this model. By considering different scenarios, this process model can be improved to create a pattern that applies to a group of similar processes in the organisation. This pattern can then act as a template for PMs should they need to model an end-to-end process in this group.

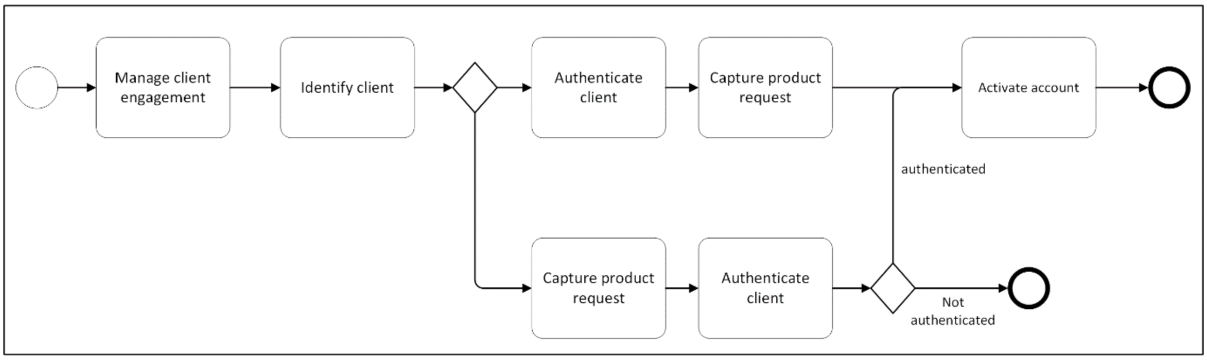


Figure 6.5: Example of a pattern model

This section described a pattern model which has the following three key characteristics:

- It is end-to-end from a client perspective.
- It is positioned at L3 in the PA.
- The name of the process had the word "- pattern" appended to it. This was implemented to facilitate easy identification of the pattern models. Many models have similar names, and PMs found it useful to be able to easily identify pattern models in a list of process models.

We now explain how the pattern model is used to create the end-to-end process model for a Scenario.

6.4.3 ST4: Identify the Scenario and the required capabilities (Process Selection Diagram)

The Process Selection Diagram (PSD) is a model available in ARIS which creates relationships between the elements of a matrix and the left-most column and top-most row of the matrix. This model automatically creates these relationships as the matrix is populated. In the context of this research project, the PSD is used to create relationships between the services used in an L3 Control-Flow model: 1) the corresponding capability and 2) the scenario in which the element occurs. These relationships are illustrated in Figure 6.6, where the service (L3 Function) is the parent of a L4 process model, which provides an instance of the capability C1. The L3 Function is also an element of the L3 model (Scenario) of which S1 is the parent. The structure of the Scenario (L3 Control-Flow model) is based on the structure of the corresponding pattern model.

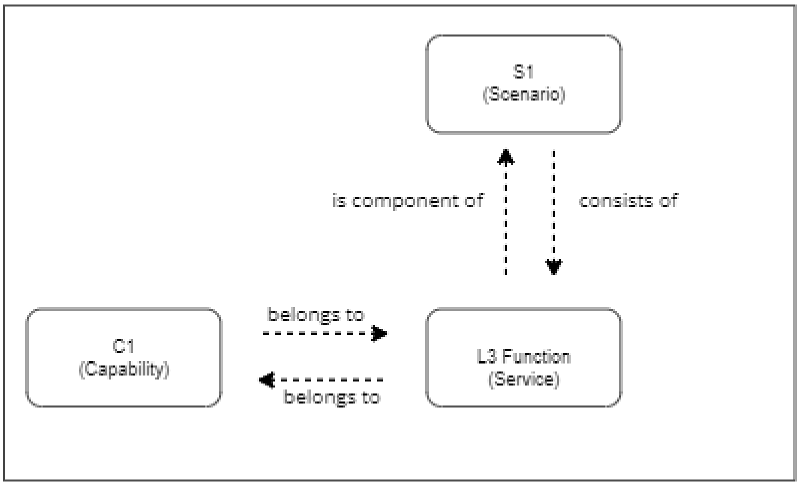


Figure 6.6: PSD Relationships

Using the pattern process model illustrated in Figure 6.5, we can construct the PSD depicted in Figure 6.7. The PSD is the mechanism that is used to create the relationships between the Capabilities, Scenarios, and the Services. The PSD is suitable for this purpose for the following reasons:

- It contains the list of the scenarios that utilise the same pattern.
- The capabilities that are used in the process pattern are depicted in the PSD.
- All services used in a scenario are depicted in the PSD.
- The relationships between these objects are automatically created as the PSD is constructed.

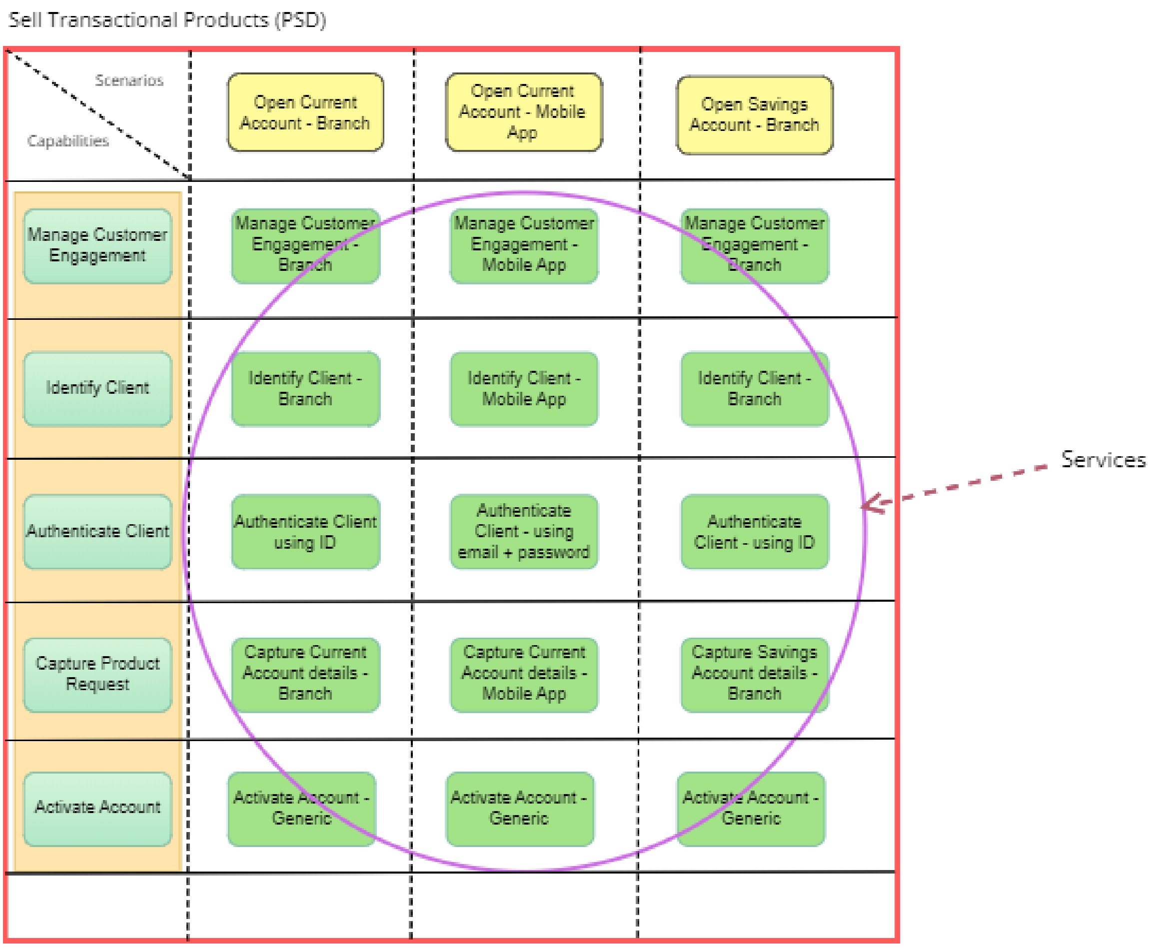


Figure 6.7: Example of a PSD

For PMs, the PSD provides a high-level view of the services used in the end-to-end process and enables the PM to quickly identify which L4 process is impacted by the current project. The PM needs to focus only on that process and any related lower-level processes. The PM is also able to identify all other related scenarios that also use the same L4 process. For example, in Figure 6.7, the Function “Manage Customer Engagement – Branch” is used in two Scenarios: “Open Current Account – Branch” and “Open Savings Account – Branch”. A report can be generated that lists all the relationships of “Manage Customer Engagement – Branch” to identify which other scenarios will be impacted by making a change to the “Manage Customer Engagement – Branch” process. Such a report would list all the relationships of “Manage Customer Engagement – Branch” in the repository and would, therefore also list Scenarios which occur in other PSDs. While we have positioned the PSD at Level 3 in the PA, it should be noted that the PSD contains both L2 and L3 objects. The Scenarios are L2 Functions (they are the parents of L3 models), and the Functions in the same columns as the Scenarios are L3 Functions (they are the parents of L4 models).

How are these models related?

Figure 6.8 illustrates the relationships between these models:

- The L2 Function, which is an abstraction of a number of processes that have the same structure, is the parent of the pattern model and the corresponding PSD.
- The pattern model informs the Capabilities (left column) in the PSD
- Scenarios are listed in the top row of the PSD.

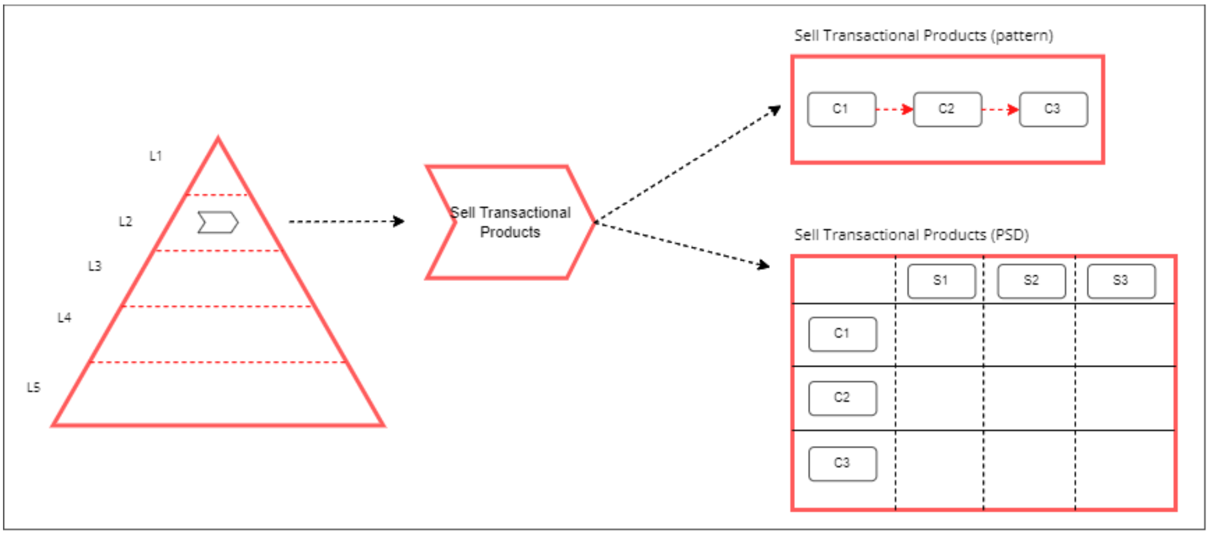


Figure 6.8: Pattern and PSD model in the Process Architecture

Figure 6.9 illustrates these relationships with an example.

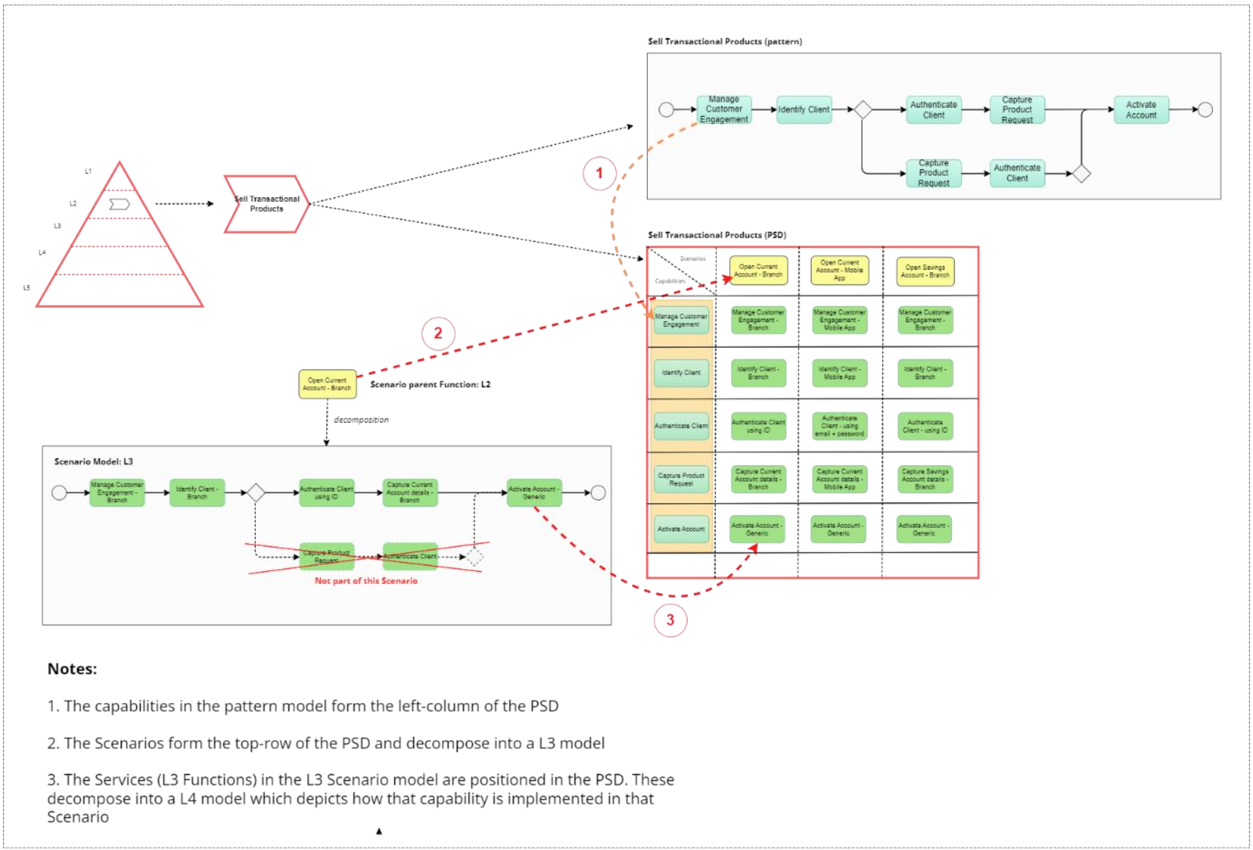


Figure 6.9: Process Pattern, PSD, and L3 Control-Flow Model example

In this section, we described the creation of the PSD, which has the following key characteristics:

- The PSD creates formal relationships between the Capabilities, Scenarios, and Services.
- The PSD is positioned in L3 in the PA.
- The PSD is related to a specific pattern model. In other words, the pattern model and the corresponding PSD have the same parent Function.
- The Capabilities from the corresponding pattern model form the left column of the PSD.
- The parent Functions of the Scenarios, which have the same structure as the pattern model, form the top row of the PSD.
- The Service which realises the appropriate capability in a given scenario is positioned at the intersection of that row and column.

6.4.4 ST5: Create end-to-end process flow for the Scenario (L3 Control-Flow model)

A Scenario has been defined as the end-to-end process from the customer's perspective. By ensuring that the services realising a particular capability (as discussed in Section 6.3) have the same initial and final conditions, the parent Functions of such models can be substituted into the pattern model to provide an end-to-end process flow for the required scenario. This Scenario model is also at L3.

It was previously stated that the pattern model is a structural template of a set of Scenarios which share a similar structure. This does not mean that all Scenarios based on the same pattern model have an identical structure. One Scenario may use a branch from the pattern model which a different Scenario does not use. In this situation, the unused branch must simply be deleted from the Scenario model (L4 Control-Flow model) which does not use it.

In this section, we have described the creation of the process flow for a specific Scenario based on the PSD as follows:

- Scenario processes are created by making a copy of the pattern model related to the PSD which contains the Scenario Function
- Each Capability in the Scenario process is replaced with the corresponding Service function from the PSD.
- Branches and Capabilities that are not relevant to this Scenario should be removed.

6.4.5 ST6: Define the Production Folder Structure

Folder structure

The models and objects in ARIS are stored in logical folders, and, as with Windows Explorer, an easily understandable and consistent folder structure facilitates the locating of the models and objects. The following requirements for the folder structure were identified:

- It must cater for both the Development and Production environments.
- It must cater for multiple projects impacting the same process model.
- It must cater for different model types.
- It must facilitate model and object management.
- It must facilitate automation of the process modelling lifecycle.

The folder structure for the first two architecture levels was defined by Architecture to be as follows:

- Division X
    - Core Value-Chain (L1)
        - Manage Sales of Products and Services (L2)
            - Sell Products and Services (L3)

At L3, different folders are created based on whether the model is a PSD, Control-Flow model, or FAD. At L3, the folder will be as follows:

- Sell Products and Services (L3)
    - L3 Control-Flow Models
    - L3 PSDs

At L4, a folder is also created in the Sell Products and Services folder for each capability to reduce the number of folder levels. The folder name is based on the name of that capability in the PSD. Then a folder is created at L4 and L5 for each model type and named “L4 XXX” or “L5 XXX”, where “XXX” is the model type. The detailed models built by the PMs are at L4 and L5. To ensure that similar models were located in the same folder, the L4 folder structure is based on the capabilities described earlier. This ensures that all L4 models which realise a particular capability are located in a L4 folder with the same name as that capability. Figure 6.10 illustrates this folder structure.

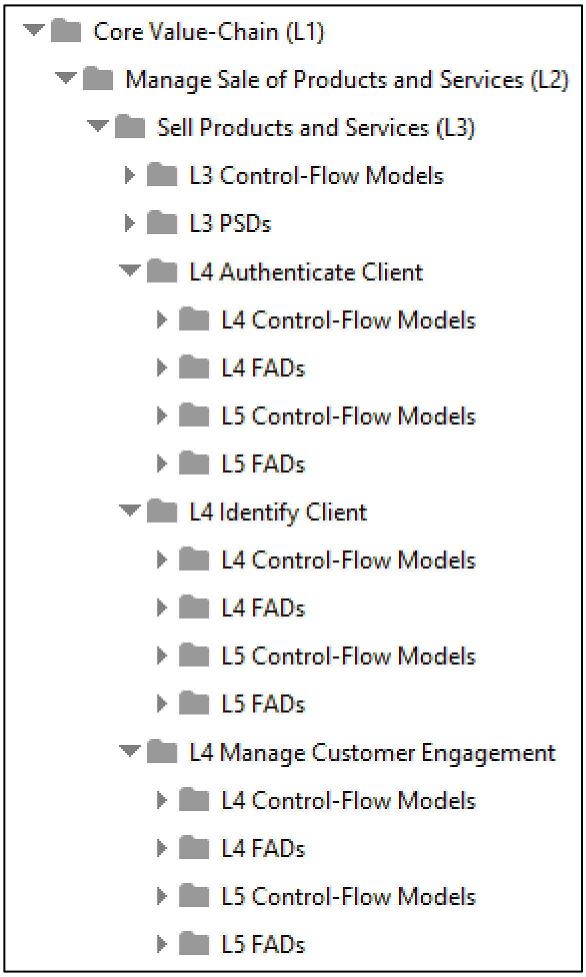


Figure 6.10: Production Folder Structure

6.4.6 ST7: Define the Development Folder Structure

The Dev folder structure adopted must be based on the Production folder structure but with the following changes to simplify the structure and cater for multiple projects impacting the same process model:

- L1 and L2 of the PA are replaced by a single folder called “Projects.”

- The "Projects" folder contains a sub-folder for each project with the project name and abbreviation.

Each project sub-folder must then contain the exact same folders as the Production folders from L3 downwards. This meant that the name of the PM was completely removed from the Dev folder structure. An example of the resulting Dev folder structure is illustrated in Figure 6.11.

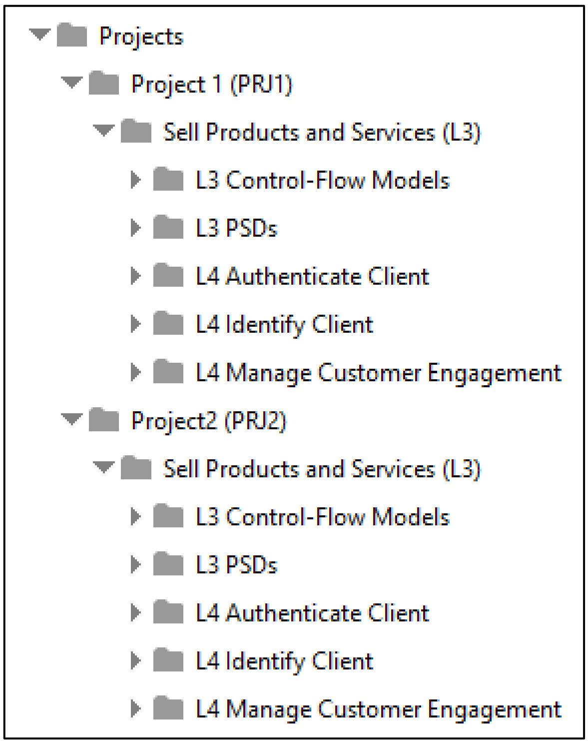


Figure 6.11: Development Folder Structure

The folder structure described uses the capabilities identified in the process pattern models to define the names of the L4 folders in the repository. It is possible to identify L5 capabilities and apply the same principles (using L5 patterns and PSDs) to the L5 process models and folder structures. This was considered but not adopted for the following reasons:

- Each L5 process model would need to be assigned to an L4 service (for consistency and positioning in an L4 PSD)
- Additional overhead would be introduced in trying to manage the capabilities and patterns at L5.

However, the location of the L5 process models in the L5 folder structure still needed to be resolved because different services (i.e. L4 process models in different folders) will sometimes contain the same activity, and the location of the corresponding L5 process model must be resolved as there will be multiple L5 folders where it could legitimately be located (each service containing this activity would have a L5 folder which could contain the corresponding process model). It turned out that by automating the process of moving models and objects from Development to Production, the location of the L5 models was not important. A particular L5 model might change location if it was reused by a service which delivered a different capability.

6.5 Ease of Use

6.5.1 ST8: Identify the models and objects which have been changed or created

In Section 3.5.5, the model and object management required for updating existing models or moving new models from Development to Production was explained. While it is possible to perform these actions in projects where only a few models need to be published, it becomes impractical on larger projects as the PM must identify every process that has been changed by the project, whether that process is a new process model or whether it is an update to an existing process model, and if it is an update to an existing process model, then the Production version of the process model to be updated must be correctly identified. We introduce the Process Publication Model (PPM) to enable automation to assist in this regard. Similarly, all new or updated objects must be identified.

The PPM is a model that lists the parent objects of all process models and model objects that need to be published for a project. Furthermore, it also contains the parent objects of any corresponding Production models that need to be updated. Where a Production model needs to be updated, a relationship between the Development parent object and the corresponding Production parent object is created in the PPM. If the project process model is a new model, then the parent object in the PPM will have no relationship to another parent object in the PPM. Different symbols were used for the Development models and the Production models to enable the automated publishing of these process models, as this enables the script to differentiate between Development and Production objects. This is illustrated in Figure 6.12.

The automated generation of the PPM is described in Step ST0. However, the creation of the relationship between a Development object and a Production object (illustrated in Figure 6.12) must be performed manually.

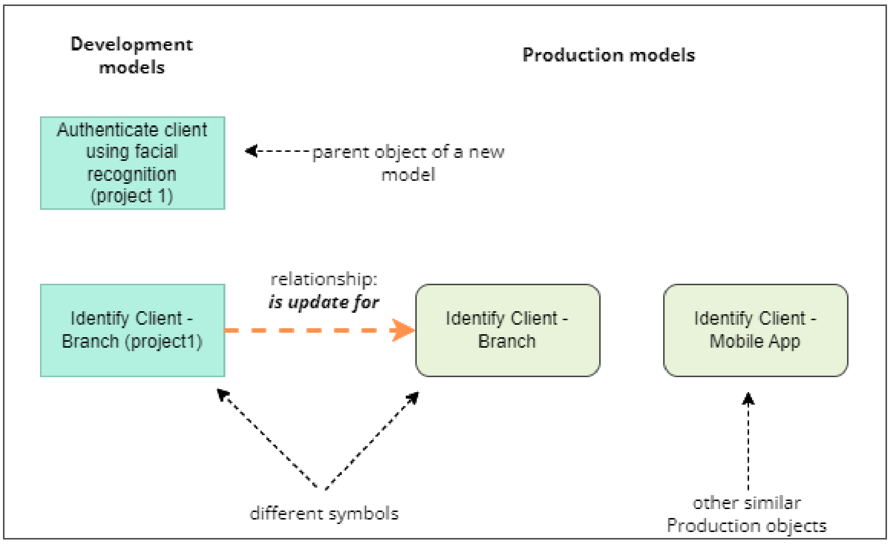


Figure 6.12: Process Publication Model (PPM)

In this section we introduced the PPM and described how it is essential for model and object management. The PPM must be created and is key to the automation (described in Step ST10) of the following activities:

- Running all the required validation reports on all the correct folders, models, and objects for a project.
- Managing the models and objects when publishing the projects (integrating the new/updated project models from the Development environment into the Production environment).

6.5.2 ST9: Manage the Folders, Models, and Objects correctly

In Section 6.3, the design of reusable process models was discussed, and in Section 6.4, further steps that were developed as part of this research project were introduced. These are sufficient to define the EPReS lifecycle for process modelling, as illustrated in Figure 6.13. The details of each step numbered are described in Table 6.4. In Figure 6.13, it can be seen that the modelling process described always begins with the PSD and the Scenario (L3 Control-Flow model). As a result, this modelling method became known as the "Top-Down" method within the modelling team.

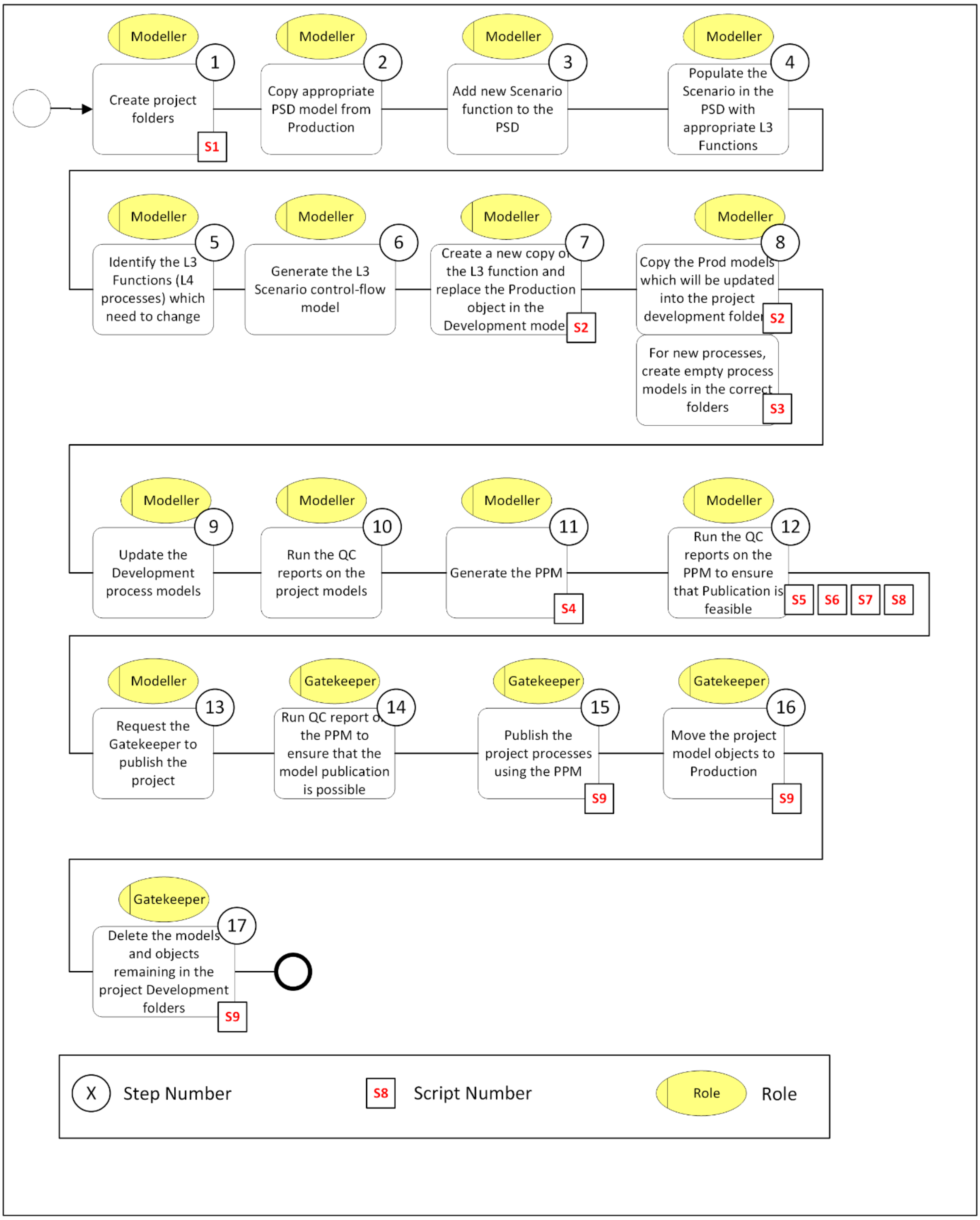


Figure 6.13: The EPReS modelling lifecycle

In this section we integrated the previous steps to illustrate the EPReS process modelling lifecycle. The lifecycle depicts the required steps as well as those steps where automation using scripts must be applied. EPReS is not practical without the automation indicated due to the complexities involved in managing the folder structure, models, objects, and a Development and Production environment in a large organisation.

Table 6.4: EPReS modelling lifecycle step descriptions

Step#	Step Name	Description	Comment
1	Create project folders	The required project folder must be created in the "Projects" folder, and the corresponding L2 and L3 model folders for this project must also be created. The path for the main "Projects" folder is fixed, but the name of the subfolders (actual project) is dependent on user input.	Create a new project. The PM enters a project name and abbreviation. The script automatically creates the project folders based on the information that the PM entered.
2	Copy the appropriate PSD model from Production.	The PSD, which contains the Scenario being updated, must be copied from the Production folder to the Development folder. If the PSD does not exist, the L2 architecture model must be updated by adding an object to which the PSD and Pattern for this PSD can be assigned. This is a manual process.	
3	Add a new Scenario Function to the PSD.	The new Scenario Function must be added to the top row of the PSD, and the corresponding L3 control-flow model must be created and assigned to the new Scenario Function.	
4	Populate the Scenario in the PSD with appropriate L3 Functions	The body of the PSD must be populated with the instance of the corresponding capability in the PSD left column.	
5	Identify the L3 Functions (L4 processes) which need to change	In the PSD, the PM must identify which L3 Functions would be changed by this project.	
6	Generate the L3 Scenario control-flow model.	The Scenario Function (L2 Function) is the parent object of the L3 Scenario control-flow model. This model must be created by making a copy of the pattern model and substituting the Capability Functions with the corresponding Service for that Capability based on the relationships contained in the PSD.	
7	Create a new copy of the L3 Function and replace the Production object in the Development model.	This must be done in both the PSD and the Scenario control-flow model. However, the same object must be used, so once it has been created for the PSD, for example, an occurrence copy of that object must be used in the control-flow model. A new L3 Function must be created for these. For model updates, a definition copy of the Production object must be created and positioned in the row corresponding to the appropriate capability.	

Step#	Step Name	Description	Comment
8	Copy the Production models, which will be updated into the Development folders. For new processes, create empty process models in the correct folders	For model updates, the Production models that were assigned to the original Production L3 Function must be copied into the correct folder. For new models, an empty model must be created and assigned to the new Function. The correct folder is based on the corresponding capability.	For new processes: Create assigned models
9	Update the Development process models	The PM must update the Development models as required.	
10	Run the QC reports on the project models	The PM must run the QC reports on all the Development models and objects.	
11	Generate the PPM	The PM must generate a PPM which contains all the information required to publish the process models updated by the project. The PPM contains the information identifying which process models in Development must update which process models in Production.	
12	Run QC report on the PPM to ensure that the model publication is possible	The PM must run a QC report on the PPM to ensure that all the models and objects are able to be published.	For example, the report checks that only models and objects related to this project will be published, as the PM may have inadvertently included a mode/object from another project.
13	Request the GK to publish the project	The PM must request that the GK publish the project.	Publication is carried out per project. The publication of a project publishes all models and objects referenced in the PPM.
14	Run QC report on the PPM to ensure that the model publication is possible	The GK must rerun the QC report on the PPM.	
15	Publish the project processes using the PPM	The GK must then: - Either move (new model) or copy (model update) content from the Development model into the Production model) - Consolidate (merge) the parent objects of Production and Development models, which were updated - Delete the assignment from the parent object to the Development model.	After the consolidation of the parent objects, the parent object will have assignments to both the Production and Development models. The assignment to the Development model must be deleted. These two models will be identical because the Development model was copied into the Production model.
16	Move the project model objects to Production.	Models that were moved/copied from Development to Production might contain objects that are still located in Development. These objects must also be moved from Development to Production.	
17	Delete the models and objects remaining in the project Development folders	Any models and objects remaining in the project Development folders must be deleted.	All the models and objects required in Production will have been moved. Leaving unused models and objects in the project Development folders will contaminate the repository.

6.5.3 ST10: Automate to assist the Process Modellers and Gatekeepers

While the process depicted in Figure 6.13 can be performed manually for very small projects, it is not scalable for larger projects. Correctly copying all the required models and objects, maintaining the correct relationships when updating the models, and then reversing the process when publishing the models is error-prone and time-consuming. In Section 6.4, we saw that the folder structures in Prod and Dev were determined by the business PA and Capabilities. Using these principles, the PSD and the PPM enable "automated" support in the form of scripts to be provided in different stages of the process modelling lifecycle.

The set of scripts described in Table 6.5, which the PM and GK could execute on a folder, model or object, depending on the context, was developed. The purpose of the scripts is to improve the productivity of the PM and GK, as well as reduce the number of errors introduced by performing the operations manually. The Script # in Table 6.5 (e.g. S1) corresponds to the Script Number shown in Figure 6.13.

Table 6.5: Automation Scripts

Scrip #	Script Name	Purpose	Description
S1.	Create new project.	Creates the L3 folder structure for a new project.	Runs on any folder. The PM inputs the project name and abbreviation. The new project and folder structure are created in the designated Development folder structure.
S2.	Create new Function and copy assigned models from Prod Function.	Replaces a Prod Function in a Dev model with a Dev Function and creates copies of all prod models assigned to the Dev Function.	Runs on a Production Function in a Development model. The attributes of all objects and models are copied from the Prod version to the Dev version. All models and objects are placed in the correct folder.
S3.	Create assigned models.	Creates the required assigned model to a new Function	Runs on a new Function in a Development model. New Functions are required as the parent object of a new process model. To ensure the integrity of the architecture, the PM must position the new Function in the model and then run this report. The new models are named correctly and created in the correct folders.
S4.	Create process publishing model.	Creates the PPM based on the L3 Scenario process which needs to be published.	Runs on one or more Scenario objects in a PSD. Multiple Scenarios can be included in the same PPM. From the Scenario, all new and updated

Scrip #	Script Name	Purpose	Description
			models and objects can be identified by navigating through the architectural hierarchy and placed in the PPM. The script also identifies Production models with similar names and places those parent objects adjacent to the Development object to assist the PM in identifying the correct Production version of the object that must be updated (Figure 6.12).
S5.	QC List content deleted from Production model.	Produced a report listing any content which will be removed from a Production model when updated with a Development model	Runs on the PPM. Used to double-check that content was not inadvertently deleted by the PM when updating an existing model in Development. The script compares the existing Production model with the Development model to be published.
S6.	QC Validate process publishing model	Uses the PPM to check that the "Process Publishing Model" script will run correctly.	Runs on the PPM. Performs a set of QC checks: - Only models and objects belonging to this project will be published. - Verifies that models and objects are in the correct folders based on their architectural level and type
S7.	Validate Project folder structure.	Checks that the project L4 folder structure exists in Production	Runs on the PPM Checks that all the folders holding models to be published also exist in the Production folder structure. This ensures that the target folders in Production for new models exist prior to trying to publish.
S8.	Verify Architecture Model or Group	Checks that the models and objects to be published all form part of the formal architecture folder structure	Runs on the PPM The repository contained folders and models which did not form part of the formal PA. This script was used to check that none of these had inadvertently been included.
S9.	Process publishing model	Processes the PPM to copy and move the models and objects from Development to Production.	Runs on the PPM. Handles the model and object management related to the model and parent objects. It must be run on the PPM.

In this section we reviewed the automation (using scripts) that is required to relieve the GKand PMs of the complexities associated with correctly managing repository folder structures, models, and objects throughout the process modelling lifecycle. These scripts are dependent on the previous steps discussed because, for example, it is not possible to correctly manage the models and associated objects (ST9) if the folder structure rules (ST6/ST7) and model/object relationships (ST3/ST4/ST5) are not rigorously enforced.

6.6 Entity Relationship Model

The entities and relationships described in the steps in Sections 6.3, 6.4, and 6.5 are now generalised in an ERD model, as illustrated in Figure 6.14. Each relationship is then described in Table 6.6.

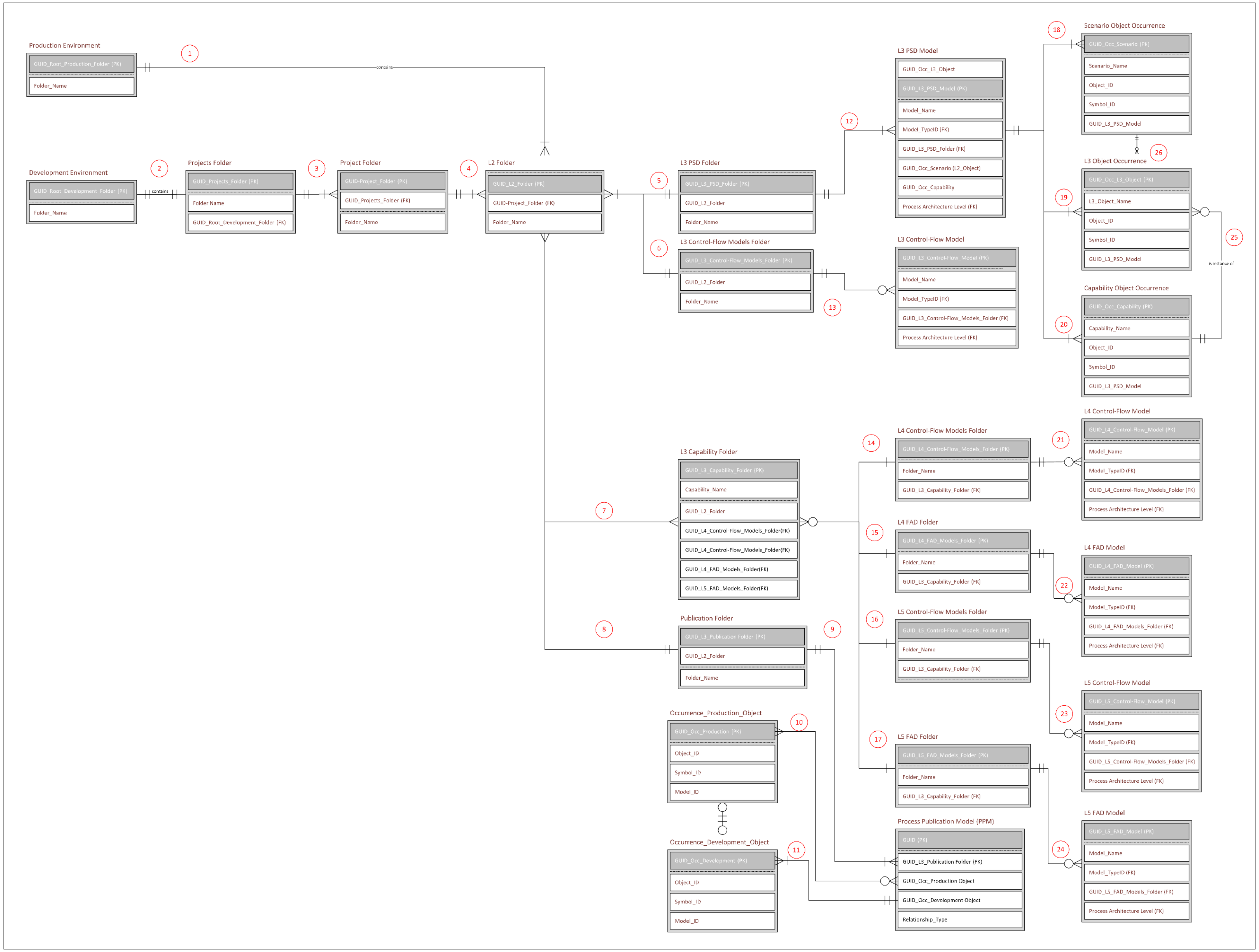


Figure 6.14: Entity Relationship Diagram

Table 6.6: Explanation of ERD Relationships

Relationship No.	Entity 1	Entity 2	Relationship Description
1	Production Environment	L2 Folder	The Production Environment contains one or more L2 Folders The L2 Folder exists in one and only one Production Environment
2	Development Environment	Projects Folder	The Development Environment contains one and only one Projects Folder. Projects Folder exists in one and only one Development Environment.
3	Projects	Project Folder	The Projects folder contains one or more Project Folders The Project Folders exist in one and only one Projects Folder
4	Project Folder	L2 Folder	The Project folder contains one or more L2 Folders The L2 Folder exists in one and only one Project Folder
5	L2 Folder	L3 PSD Folder	The L2 Folder contains one and only one L3 PSD Folder The L3 PSD Folder exists in one and only one L2 Folder
6	L2 Folder	L3 Control-Flow Models Folder	The L2 Folder contains one and only one L3 Control-Flow Models Folder The L3 Control-Flow Models Folder exists in 1 or more L2 Folders
7	L2 Folder	L3 Capability Folder	The L2 Folder contains one and only one L3 Capability Folder. The L3 Capability Folder exists in 1 or more L2 Folders.
8	L2 Folder	Publication Folder	The L2 Folder contains one and only one Publication Folder. The Publication Folder exists in one and only one L2 Folder.
9	Publication Folder	Process Publication Model	The Publication Folder contains one or more Process Publication Models. The Process Publication Model exists in one and only one Publication Folder.
10	Production_Object_Occurrence	Process Publication Model	The Production_Object_Occurrence occurs in 0 or more Process Publication Models. The Process Publication Model contains Production_Object_Occurrence.
11	Development_Object_Occurrence	Process Publication Model	The Development_Object_Occurrence in one and only one Process Publication Model

Relationship No.	Entity 1	Entity 2	Relationship Description
			The Process Publication Model contains at least 1 Development_Object_Occurrence.
12	L3 PSD Folder	L3 PSD Model	The L3 PSD Folder contains one or more L3 PSD Models The L3 PSD Model exists in one and only one L3 PSD Folder
13	L3 Control-Flow Models Folder	L3 Control-Flow Model	The L3 Control-Flow Model Folder contains one or more L3 Control-Flow Models. The L3 Control-Flow Model exists in one and only one L3 Control-Flow Models Folder.
14	L3 Capability Folder	L4 Control-Flow Models Folder	The L3 Capability Folder contains one and only one L4 Control-Flow Models Folder. The L4 Control-Flow Models Folder exists in one and only one L3 Capability Folder.
15	L3 Capability Folder	L4 FAD Folder	The L3 Capability Folder contains one and only one L4 FAD Models Folder L4 FAD Folder exists in one and only one L3 Capability Folder.
16	L3 Capability Folder	L5 Control-Flow Models Folder	The L3 Capability Folder contains one and only one L5 Control-Flow Models Folder. The L5 Control-Flow Models Folder exists in one and only one L3 Capability Folder.
17	L3 Capability Folder	L5 FAD Folder	The L3 Capability Folder contains one and only one L5 FAD Models Folder The L5 FAD Folder exists in one and only one L3 Capability Folder
18	L3 PSD Model	Scenario Object Occurrence	The L3 PSD Model contains one or more Scenario Object Occurrences The Scenario Object Occurrence occurs in one and only one L3 PSD Model
19	L3 PSD Model	L3 Object Occurrence	The L3 PSD Model contains one or more L3 Object Occurrences The L3 Object Occurrence occurs in one and only one L3 PSD Model
20	L3 PSD Model	Capability Object Occurrence	The L3 PSD Model contains one or more Capability Object Occurrences The Capability Object Occurrence occurs in one and only one L3 PSD Model

Relationship No.	Entity 1	Entity 2	Relationship Description
21	L4 Control-Flow Models Folder	L4 Control-Flow Model	The L4 Control-Flow Models Folder contains 0 or more L4 Control-Flow Models. The L4 Control-Flow Model exists in one and only one L4 Control-Flow Models Folder.
22	L4 FAD Folder	L4 FAD Model	L4 FAD Folder contains 0 or more L4 FAD Models L4 FAD Model exists in one and only one L4 FAD Folder
23	L5 Control-Flow Models Folder	L5 Control-Flow Model	L5 Control-Flow Models Folder contains 0 or more L5 Control-Flow Models L5 Control-Flow Model exists in one and only one L5 Control-Flow Models Folder
24	L5 FAD Folder	L5 FAD Model	L5 FAD Folder contains 0 or more L5 FAD Models L5 FAD Model exists in one and only one L5 FAD Folder
25	Capability Object Occurrence	L3 Object Occurrence	Capability Object Occurrence is a type of L3 Object Occurrence L3 Object Occurrence belongs to Capability Object
26	Scenario Object Occurrence	L3 Object Occurrence	Scenario Object Occurrence has component L3 Object Occurrence L3 Object Occurrence is a component of Scenario Object Occurrence

6.7 Summary

In Sections 6.3, 6.4, and 6.5, the ten steps that were described in detail are listed in Table 6.7

Table 6.7: Step Listing

Step No.	Step Name
ST1	Identify and define capabilities
ST2	Align with the business architecture.
ST3	Define end-to-end generic processes at L3 (Process Pattern models).
ST4	Identify the Scenario and the required capabilities (Process Selection Diagram)
ST5	Create end-to-end process flow for the Scenario (L3 Control-Flow model)
ST6	Define the Production Folder Structure
ST7	Define the Development Folder Structure
ST8	Identify the models and objects which have been changed or created
ST9	Manage the Folders, Models, and Objects correctly
ST10	Automate to assist the PM

Referencing the new process modelling lifecycle described in Figure 6.13, these steps ensure the following:

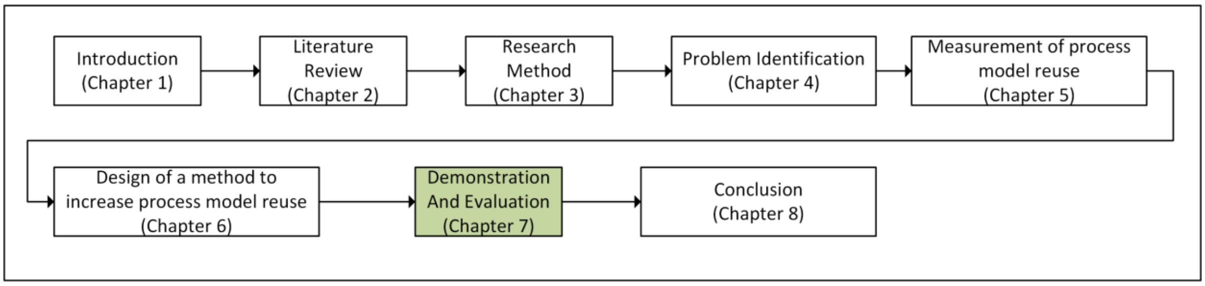
1. Creating the correct project folder structure in the correct location
2. Naming the folders correctly
3. Copying models from Production into Development correctly (into the correct folders) for updating
4. Creating blank (new) models correctly in the Development project folder
5. Running quality checks on the development models in a project before publishing the project models to production
6. Moving the project models from Development to Production correctly.
7. Managing the models, objects, and relationships correctly when publishing from Development to Production.

The approach of providing scripts for the PM to use also then eliminated the requirement for the GK to create new folders, create empty models, and make copies of existing models. The automation scripts specifically eliminated the manual effort required for the following modelling tasks:

1. Creating project folders.

2. Creating the PSD.
3. Copying models from Production to Development.
4. Copying models from Development to Production.
5. Replacing Functions and Copying assigned models when updating a process model; Copying attributes.
6. Creating a new model assigned to a new Function.
7. Executing quality Control checks.
8. Creating the Process Publication Model.
9. Validating the integrity of the PPM.
10. Managing the objects.

7 Chapter 7: Demonstration and Evaluation



Metrics for the reuse of process models (and their elements) were previously proposed using a modified form of function point analysis (Daneva, 1999, 2000), and a similar approach was adopted in this research. The number of times a specific process model is reused can be counted directly. However, a process model that has reused another process model may, in turn, also be reused, which effectively results in further reuse of the original process model, and this situation should be accounted for (Daneva, 1999, 2000). This is described more fully in Appendix 2. To conduct a comprehensive analysis, the level of reuse was also analysed by taking other dimensions into account (e.g., business area, architecture level of the model). Appendix 2 explains the structure of the repository and this approach. Appendix 1 describes how process models are represented in ARIS.

The process model reuse was measured by calculating the frequency of process model reuse for each population identified. Each process model was assigned various attributes (e.g., new or old modelling method, business area, production or development model, model architecture level, model complexity) to enable reuse comparisons to be made between different dimensions. It should be noted that such attributes were not mutually exclusive. The comparisons were carried out by conducting either parametric or non-parametric tests (depending on the normality of the data) to determine whether there is a significant difference in process model reuse between the populations.

In this section, we demonstrate quantitatively that the proposed method increases the level of reuse of complete process models in the repository. This was done by comparing the level of reuse of the models created using EPReS to the level of reuse of models created using the “old” method. The proposed method was then evaluated qualitatively based on interviews with the participants.

7.1 Demonstration – Increasing the level of reuse of complete process models in a repository.

To demonstrate the improvement in the level of reuse, we calculated the level of reuse of each population indicated in Figure 3.3 using the measure that was developed in Chapter 5. This provides a view of the improvement within the same team (Team X) when using EPReS, as well as a comparison with the PMs in the rest of the organisation. These populations were identified based on the folder structure where the parent Function of each model is located, and each population is independent of the other two. The tabulated results are contained in Appendix 4. These results are best illustrated graphically in Figure 7.1.

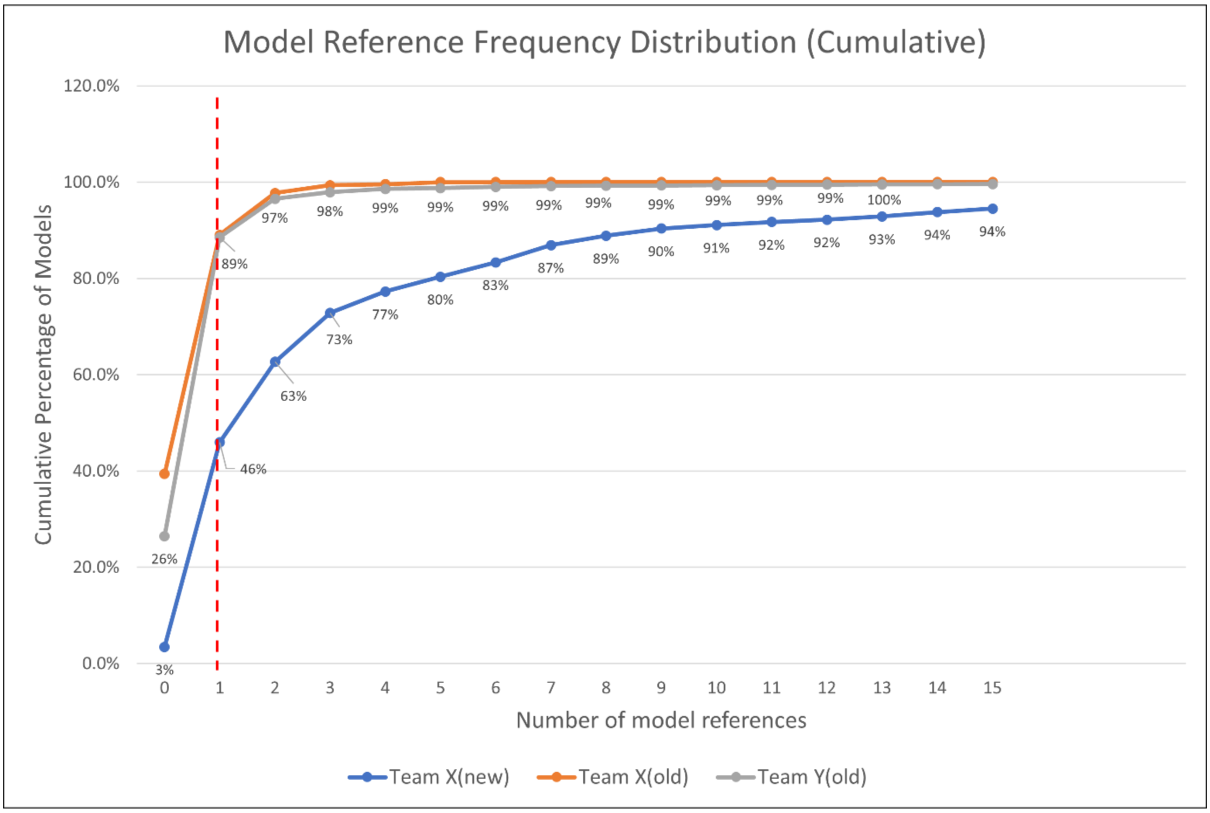


Figure 7.1: Model Reference Frequency Distribution

The critical data points to consider when interpreting Figure 7.1 are those that correspond to the value of "1" on the X-axis (indicated by the broken vertical line). The graphs depict the cumulative percent models referenced for each value on the X-axis, so the y-values correspond to the percent models referenced once or less (i.e. 0 or 1 time). Reuse can only be considered to have occurred when the model is referenced more than twice. Furthermore, even if a model was referenced 10x, it is only counted as reused once (as explained in Section 5.2.4). Therefore, the level of reuse for each population is the fraction of the population which is NOT referenced once or less, as illustrated in Table 7.1

Table 7.1: Level of Reuse of Identified Model Populations

Population	Referenced <=1 time	Level of Reuse (Referenced >=2 times)
Team X_{new}	46%	54%
Team X_{old}	89%	11%
Team Y_{old}	89%	11%

Table 7.1 also suggests that the improvement in the Level of Reuse is not just due to the quality of the PMs themselves. Team X_{new} and Team X_{old} are the same team when using EPReS and when using the old method. Furthermore, it was interesting to note that Team X_{old} and Team Y_{old} displayed the same level of reuse (11%), which triangulates well with the limited estimates of process model reuse of approximately 10% reported in the literature previously (Fellmann et al., 2014; Koschmider et al., 2014). Accordingly, we claim that the proposed method does increase the level of complete process model reuse.

It is difficult to know whether a Level of Reuse of 54% is good or can be improved because a level of reuse of 100% is neither realistic nor possible because there will always be some processes which are not reused. This is likely to depend on the nature of the business, where businesses providing multiple products across multiple channels (as in the financial services sector) will have greater opportunities for process model reuse than single-line or single-channel businesses.

7.2 Evaluation

In the previous section, it was demonstrated quantitatively that the new modelling method increased the LOR from 11% to 54%. In this section, we evaluate EPReS in the eyes of the organisation by considering 1) feedback from interview participants, 2) an analysis of the benefit of EPReS based on a real-world project that was conducted, and 3) the projects in which the new modelling method was used.

7.2.1 Participant Feedback

Feedback from the participants pointed to specific benefits of EPReS, which can be categorised into six areas:

<u>Reduced time to model</u>

Confirming that EPReS reduces modelling time. P09 stated, "*I think, for now, it's the reusability of processes. And the time that it's taking to model all these processes is certainly one benefit. And I* *think I just wait for, for the day that probably close to a 95% on reuse of models. And I think the businesses, as well, are starting to accept the method*", while P11 also saved time in their project: "*In my project, yes definitely, and you can refer back to the analysis that I did. But it definitely saved a lot of time and then also cost, and then the benefit of getting through documenting the processes by reusing and not recreating*".

Scope Management

The PSD enables the scope of the project modelling required to be more precisely defined by identifying only those models that need to be updated rather than remapping the entire end-to-end process. P11 explained it like this: "*So, the experience that I have had is that when we are trying to do updates, we're trying to model. Yes, the process does exist. So, it is it is nice to copy and paste, and you only need to do a certain amount of updates versus remapping. So yes, it does help*".

Impact Analysis

Related to Scope Management, identifying the impact of changing a process assists in avoiding unintended consequences caused by overlooking other processes or business areas that might be affected. EPReS correctly ensures the correct maintenance of relationships, which facilitates identifying, for example, which Scenarios all use a specific L4 process. As P3 puts it: "*Because, again, as I said, before you embark on any initiative, you want to go out there and see who else has done this? Does it exist in the organization? If it exists, is it in a stage where you can, you know, reuse, refer to it or do something? You know, unlike in the past, as I said, you, you get a piece of work, you know, that silo mentality, you say, I'm doing this for, here are my stakeholders. But, but this, this, this way of doing things, it forces you to look at the bigger picture. Where does it fit in the whole enterprise? Who else might be impacted by this? Has it been done before? If not, if I do it, who else might benefit from this piece of work that I'm doing?*"

End-to-end Processes

The presentation of an end-to-end process can become laborious because it is usually decomposed into smaller processes that have been modelled separately without a model at a higher level of abstraction, which guides how the lower-level processes are related. For example, the old modelling method did not provide for L3 process models, and therefore, it was difficult to find the L4 process models that integrate to deliver an end-to-end process. EPReS forces the creation of the corresponding L3 process models. As P09 explained when referring to a business walkthrough (of the process): "*And I think I just wait for, for the day that probably close to a 95% on reuse of models. And I think the businesses as well, are starting to accept the method. At first, it wasn't. But it is becoming a bit easier as well when it comes to process walkthroughs*".

Knowledge Management

Understanding which process models exist, what they were created for, where to find them, and what their shortcomings might be, is important because it naturally increases perceived confidence in the models, which increases the inclination to reuse such models. P05 described it as "*I think all the benefits that I have, the quality that goes with it, and the reusability that goes with it. And the fact that you know, when you keep reusing the models, and I'm not even talking about the objects, the models, from a PR level, or SP level, when you keep reusing them, you know, they become second nature in your mind. So, you understand. So that's one of the benefits. So, models are hardly ever new to you, even if it's a new area that you're modelling for. They're hardly ever new on a new project. So that's, I think that those are some of the benefits that that you can get.*"

Support for Agile Methodology

EPReS supports the Agile Project Methodology. In some Agile methodologies, a Sprint is a period of two calendar weeks where a deliverable must be completed. From a process modelling perspective, it is not possible to complete an end-to-end process in two weeks. EPReS forces the decomposition of the end-to-end processes (Scenarios) into L4 and L5 processes which would be completed within two weeks. P02 described it like this: "*So from an agile perspective, if you don't have reuse, you can't be agile. Then we've missed it there. So it does make it easier. And it's, and it's quicker for everybody to get onto the same page from understanding. So it means you can bring in people outside of the organization. They don't need to know your business, they understand it, at a high level, can be reused.*"

Table 7.2 Participant views regarding EPReS

Participant No.	Quotation
P09	And if, you know, engineers can, can grasp this information as quickly as possible, then I think we'll be able to get a proper sense, and we can definitely see this method working for us.
P05	I think all the benefits that I have, the quality that goes with it, and the reusability that goes with it. And the fact that you know, when you keep reusing the models, and I'm not even talking about the objects, the models, from a PR level, or SP level, when you keep reusing them, you know, they become second nature in your mind. So, you understand. So that's one of the benefits. So, models are hardly ever new to you, even if it's a new area that you're modelling for. They're hardly ever new on a new project. So that's, I think that those are some of the benefits that that you can get.
P09	I think, for now, it's the reusability of processes. And the time that it's taking to model all these processes is certainly one benefit. And I think I just wait for, for the day that probably close to a 95% on reuse of models. And I think the businesses as well, are starting to accept the method. At first, it wasn't. But it is becoming a bit easier as well when it comes to process walkthroughs.

Participant No.	Quotation
P03	Because, again, as I said, before you embark on any initiative, you want to go out there and see who else has done this. Does it exist in the organization? If it exists, is it in a stage where you can, you know, reuse, refer to it or do something? You know, unlike in the past, as I said, you, you get a piece of work, you know, that silo mentality, you say, I'm doing this for, here are my stakeholders. But, but this, this, this way of doing things, it forces you to look at the bigger picture. Where does it fit in the whole enterprise? Who else might be impacted by this? Has it been done before? If not, if I do it, who else might benefit from this piece of work that I'm doing?
P04	So, the experience that I have had is that when we are trying to do updates, we're trying to model. Yes, the process does exist. So, it is it is nice to copy and paste, and you only need to do a certain amount of updates versus remapping. So yes, it does help. On the one hand, and this is not within a project.
P07	In my environment. I'm looking at statistics. My statistics teach me that almost 80 to 90% of the processes we build end up being reused. It's just a question of time. How much time goes past before this gets reused? If it hasn't been reused at all, one could say it's a standalone process of no reuse. But if it is reused once, there you go. You started the journey.
P11	In my project, yes, definitely, and you can refer back to the analysis that I did. But it definitely saved a lot of time, and then also cost, and then the benefit of getting through documenting the processes by reusing and not recreating.
P03	But now whenever you get a piece of work or initiative that you're working at, that you're supposed to be working on you, you, you're not only looking at what needs to be done and, and you Firstly, the whole exercise of, say it's a new initiative, it's new processes. The first thing that you do in terms of trying to establish what the architecture home is, is for this thing, you look at the enterprise view. Now you say, oh, okay, this is where this thing fits in. Who else is using this? Who else is going to be impacted for the first time you starting to look at things holistically? To say this is not only for D9, one division in D9, for example, you start to get the whole picture to say first, the architecture home. Has it been done? Who else in the organization? How is Dept2 doing this thing? Are they doing it at all? You know, you, you, you’re now starting to look at the bigger picture to say who's using these models or who's going to use them in the future. And, and, by, by, by looking at the whole enterprise view, looking at those value chains, and you know, everything that has to, it might be a small process just a little bit piece of a part, you're forced to look at it from outside the specific area that you, you’re modelling for.
P11	**##P11##** 13:24 Yes, definitely. I think when it, when we first started off with it, it was quite overwhelming for the team to understand it, and understand why we, we had to now, all of a sudden, reuse and not recreate all of the processes. So, it was quite frustrating. But, the benefits of reuse were quite obvious after the first project. **Interviewer** 13:46 Okay, and how did you see those in your first project? **##P11##** 13:48 In the first project, we actually took the time it would have taken if we did it the old method, and also the new method to actually show what the benefits are and what the time saved is on the projects. **Interviewer** 14:08 And can you recall what, what the difference was or what the, what the outcome of that was? **##P11##** 14:13

Participant No.	Quotation
	I can't recall the details on that. But I am able to give you the stats in the presentation that was done where feedback was given to the business.

Challenges

When asked what issues they had experienced with EPReS, two perspectives of the modelling process taking longer than expected were mentioned: 1) small quick changes to an existing model and 2) the publication process.

Small quick changes to an existing model

EPReS must always start with the PSD. When the change is required to a L5 model, the whole hierarchy from L5 up to L3 will potentially require an update. To ensure that the integrity of the relationships and hierarchy is maintained, this is a rule of EPReS. As P07 explained: "*I must say there may be times when reuse extends the time of modelling. If I asked, if I was asked to go do a quick standalone, dirty level five on how to log into an application, it could take me five minutes, 10 minutes, but if they ask me to do a top-down, I still have to book out a scenario, book out a L4 where they're opening NTE and add in the leg for this application on a L5 model. Then I need to publish the entire pack, which means I'm no longer only going to do my quality control on my activity standalone, but I need to make sure that the scenario I inherited and the subprocess is also quality controlled. So if you make a small change on a large reused model, If you report your reporting your quality control semantics and stuff, if you're not aufait with it. It might take you a little bit longer*".

Publication Process

The publication process was identified as taking too long. P04 stated: "*I think the publication process is a quite, a long one. That's one of the major issues and it does take time to do that. And that's just because of, there's a lot of things that need to be done before you can actually publish your, your process*", and "*And what happens is, is because of the top down and alignment of patterns, and PSDs and those kinds of things, publishing always takes time, or, or just the entire process takes time. So mod, modelling, an actual process doesn't take much time. But the rest of the admin behind it takes time.*". However, this is mainly due to the required QC checks, which are not related to EPReS.

7.2.2 Real-world project benefit

The time saved on a project due to EPReS was formally measured by one of the participants who provided the additional information listed in Table 7.3 and Table 7.4, based on a presentation prepared for the business unit involved:

Table 7.3: Real-world example project statistics

Models	Number
L3 Control-Flow	2
L4 Control-FLow	9
L4 FADs	25
L5 Control_Flow	25
L5 FADs	148

Table 7.4: Real-world project savings

	Time
Initial estimate to complete the project.	2640 hours
Actual cost to completion	1414 hours
Saving (hours)	1226 hours
Saving (%)	46.4%

7.2.3 Projects which have used the new modelling method

Another lens that can be used to evaluate EPReS is to consider how frequently it is being used. Table 7.5 shows the number of projects that have been published each year since 2016. Over the years 2022 and 2023, there was an average of 1 project being published per week.

Table 7.5: Number of projects published per year

Year	Number of projects published
2016	16
2017	17
2018	11
2019	36

Year	Number of projects published
2020	19
2021	35
2022	59
2023	53

7.3 Discussion

DSR is concerned with the design of an artifact which has practical utility. However, one must differentiate between the creation of an artifact based on existing knowledge and the creation of an artifact which has been designed and constructed in a novel manner. The former is not considered to be new knowledge while the latter will contribute to the IS knowledgebase if it is considered new knowledge. Another distinguishing feature of DSR is that the artifact should be the instantiation of an IS theory and therefore the rigorous evaluation of a DSR artifact is essential (Peffers et al).

How to evaluate DSR has been debated in the literature, with the focus of the evaluation ranging from ensuring rigour, validating the underlying theory, assessing the utility of the artifact, and ensuring that the epistemological roots of the evaluation are consistent with the design decisions and DSR methodology used (Gregor, 2006; Gregor & Jones, 2007; Hevner et al., 2004; Larsen et al., 2020; Peffers et al., 2007, 2012; Pries-Heje et al., 2008; Sonnenberg & vom Brocke, 2012; Venable et al., 2012)

Evaluation of the artifact must address artifact design and usage to ensure that the design was conducted in scientific manner and that the artifact is of practical benefit (Pries-Heje et al., 2008). If the artifact is not practical or does not solve the problem for which it was designed, then it is of no use and cannot be considered knowledge. In this research, to ensure rigor in the process, the DSR methodology of Peffers (2018) was adopted, and the approach to theory development proposed by Gregor was used to determine the type of theory appropriate for each research question (Gregor, 2006).

Focussing on the theoretical contributions of this research, Table 7.6 shows how each theory construct identified by Gregor is realised in this research. The specific sections in this thesis which realise each construct are indicated in bold italics for each research question (Contribution). It is noted that each contribution meets the requirements and can be considered a theory.

Table 7.6: Validating the theoretical contribution

Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
Components Common to All Theory			
Methods of representation	Diagram + Words ***See section 6.3-6.6***	Diagram + Words ***See section 4.4***	Words (Formula) ***See sections 5.2.3-5.2.4***
Primary Constructs	Level of reuse, business processes, patterns, PA, and services. ***See sections 6.3 6.6***	Relationships between factors resulting from or impacting the level of reuse ***See section 4.2***	Level of reuse, business processes, models, functions, object definitions, object occurrences ***See section 5.2***
Statements of relationship	Provides a method (architecture, ERD, process) and explanations for how to design a method which increases the level of reuse of process models. ***See section 6.6***	Causal Loop diagram indicating the system dynamic resulting from the lack of reuse of process models. ***See section 4.2***	Provides a formula for the calculation of the level of reuse based on variables which can be quantitatively measured in a repository. ***See Eq. 4 and Eq. 5***

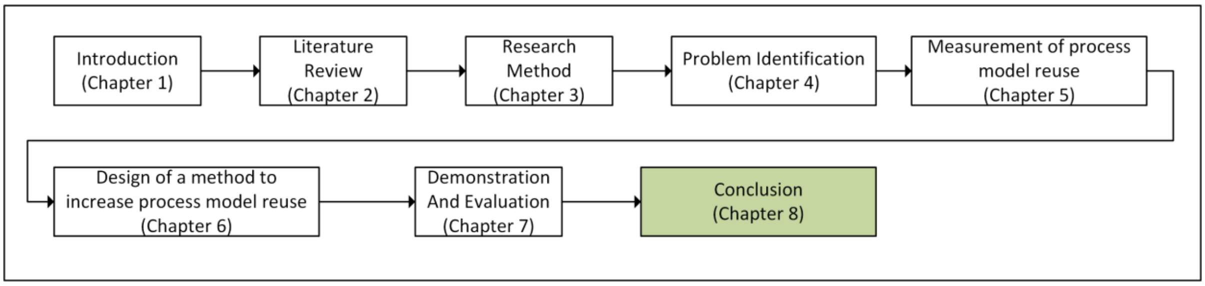
Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
Scope	Enterprises conducting BPM using a process repository. ***See section 1.1***	Enterprises conducting BPM using a process repository ***See section 1.1***	Enterprises conducting BPM using a process repository ***See sections 2.6.1 and 5.1***.
Components Contingent on Theory Purpose			
Causal Explanations	Underlying theories that explain the design include the Technology Acceptance Model when applied to BPM. Different factors impacting the PM's "inclination to reuse" are considered. ***See sections 6.2-6.5***	The statements of relationships include causal explanations. ***See section 4.2.1-4.2.10***	None
Testable propositions	The claim is made that the design theory will assist other organisations to increase the	The propositions (relationships between the elements) are tested and confirmed in interviews with participants. ***See section 4.2.1-4.2.10***	The claim is made that the design theory will assist organisations in measuring the level of reuse of process models in their repository.

Theory Overview:			
Theory Components	Instantiation		
	RQ1: How can the level of complete process model reuse be increased?	RQ2: Why does a lack of process model reuse impact business costs?	RQ3: How can the level of reuse of complete process models be measured?
	level of reuse of process models. ***See section 7.2***.		See section 5.4
Prescriptive Statements	The design theory defines the meta-requirements and principles which can be used to increase the level of reuse of process models. ***See sections 6.2-6.6***	Not present. The design theory is explanatory in nature.	The design theory specifies a formula and algorithm that can be used to measure the level of reuse of process models quantitatively. ***See sections 5.2.2-5.2.4***

7.4 Summary

In this chapter, the quantitative measurement of the LOR was shown to increase from 11% (9475 models) to 54% (1398 models) when the new modelling method was adopted. The cost saving in one project was estimated to be 46.4%. As of 2023, the process modelling work for a total of 246 projects had been completed using the proposed modelling method.

8 Chapter 8: Conclusion



In this section, we revisit the research questions and explain how they contributed to achieving the main objective of this research project. The research approach and research method used to answer each research question are described. A summary of the findings is then presented, followed by the contribution made by this research project and possible directions for future research. Finally, the limitations of this research are discussed.

8.1 Problem statement and research questions revisited

This research project is titled "Increasing business process model reuse in a multi-product multi-channel environment". It is focused on developing a new process modelling method which increases the level of reuse of complete process models. The research took place in a large South African financial services organisation which has practised business process modelling using a repository for over 25 years. Using a DSR methodology, we first confirmed that the level of process model reuse is a problem by why not reusing process models impacts business costs. Then, a method of measuring the level of reuse was developed. Finally, a modelling method to increase the level of reuse was developed and tested.

8.2 Literature review and research method

No research was discovered during the literature review that related to increasing the level of reuse of complete process models. This was underscored by the fact that, while literature existed related to measuring the reuse of software, no measure of reuse of process models was found. A pragmatic philosophy was appropriate as the main objective of this study was to develop a new modelling method using DSR, and knowledge in the pragmatic paradigm is based on the utility of the artifact designed. Furthermore, pragmatism supports the use of a mixed methods strategy where both quantitative and qualitative data are collected. Quantitative data used to measure the level of reuse was collected from the process repository, while qualitative data was collected from semi-structured interviews, which were used to determine the consequences of not reusing process models, as well as evaluate the usefulness of the proposed modelling method. The literature review confirmed that the research questions posed and answered in this research are still relevant.

8.3 Summary of Findings

In this section, we discuss the findings as they relate to each research question that constitutes a contribution of this research project. A DSR approach was adopted with different stages in the approach covered by the key research questions.

8.3.1 Research Question 1

How can the level of complete process model reuse be increased?

This research question also forms part of Stage 3 (Design and Development). To answer this research question, we introduced Capabilities and SOA principles to improve the reuse of process models, introduced two new model types into the method, adapted the PA and repository structure, and automated the model and object management in the repository as listed in Table 8.1.

Table 8.1: Summary of the new modelling method

Step	Description
ST1. Identify and define capabilities	Capabilities are “What” the organisation does. Services are “How” the organisation does it. A service is a specific process which enables a particular capability.
ST2. Align with the business architecture	The PA is a structured view of how the organisation views its processes.
ST3 Define end-to-end generic processes at L3 (Process Pattern models).	The process pattern is a process model template which applies to a specified grouping of end-to-end process models.
ST4. Identify the Scenario and the required capabilities (Process Selection Diagram).	The PSD defines the relationships between the Scenarios, Capabilities, and Capability instances in a Scenario.
ST5. Create end-to-end process flow for the Scenario (L3 Control-Flow model)	A L3 Control-Flow model is an instantiation of the process pattern model described in Step ST3.
ST6. Define the Production Folder Structure	These are the folders in the Production Environment (Prod), which contain the latest official version of each process. These models reflect the current practice in the business.
ST7. Define the Development Folder Structure	These are the folders in the Development Environment (Dev) which contain the project version of each process. These models reflect the future practice in the business once the process has been published.
ST8. Manage the Folders, Models, and Objects correctly	Folder, Model, and Object Management describes how the Folders in the repository are named based on the Process Selection Diagram and the rules and method used to create new models, copy models, and correctly move repository models and objects from the Development folder structure to the Production folder structure.
ST9. Identify the models and objects which have been changed or created	The Process Publication model defines: 1) which process models need to be published, 2) which models are new models, and 3) which models are updates to existing models.

Step	Description
ST10. Automate to assist the PM	Describes the automation of various steps in the process modelling lifecycle. In this context, automation means the running of a script by the PM/GK which automatically executes one or more tasks which would have been carried out by the PM/GK.

Using EPReS resulted in a LOR of 54% in a population of 1398 models in the repository.

8.3.2 Research Question 2

Why does a lack of process model reuse impact business costs?

This research question underpins the motivation for the research project by using a CLD to illustrate the impact on the organisation of not reusing process models. The CLD was developed by identifying potential elements and their relationships from the literature and then using analysis of semi-structured interviews to validate these relationships. It was found that the elements identified exhibited system dynamics' characteristics which resulted in virtuous/vicious cycles, resulting in reduced/increased costs to the organisation, depending on whether process models were reused (or not).

The lack of reuse of existing processes results in a number of negative outcomes for the organisation, both in the short term and in the longer term. Firstly, it leads to unnecessary duplication of process models in the repository. The simple act of unnecessarily recreating a process model means that that effort is an unnecessary cost to the organisation. Secondly, the additional process models in the repository result in a larger repository, thereby increasing the management overhead associated with the repository. This impact is not immediate but evolves over a longer time period. Thirdly, multiple versions of the same process in the repository result in additional investigation being required in the future when that process ought to be reused. It is then not clear which version of the process should be used, and time is wasted trying to figure out the answer. This problem may even set up a vicious cycle whereby it becomes easier for the PM to just create another version of the process, which in turn exacerbates the problem. Fourthly, the number of process models in the repository which are no longer used increases, again impeding reuse for the same reasons as just explained.

All of these issues increase the cost of modelling and managing the business processes, which eventually leads to the "Value of process modelling" (Issue #2) being brought into question. It is, therefore, in the interest of the organisation that the reuse of existing business process models is maximized where appropriate. This is supported by a study where process model reuse was identified as the 9th most important benefit that was expected and was also identified as a future challenge (Indulska, Recker, et al., 2009). It is intuitive that increasing the reuse of models in a collection will result in fewer new models having to be built. However, the reuse of process models in practice is proving difficult due to limited support for process model reuse currently offered by vendors of modelling software (Koschmider et al., 2014). This in itself will increase the cost of the process modelling unnecessarily, as it ought to be less expensive to reuse a process model rather than create it from scratch.

8.3.3 Research Question 3

How can the level of reuse of complete process models be measured?

This research question forms part of Stage 3 (Design and Development). It was important because if one is trying to improve something, one would like to measure the initial state and then measure the final state to obtain a measure of the improvement. A process repository should offer the opportunity to obtain such a measure quantitatively. However, no such measure was identified in the literature.

A quantitative measure was developed based on published research relating to the reuse of software. The generation of this measure involved exporting data from the repository and then processing this data (explained in Section 5) to arrive at the LOR. The structure of the repository enabled this measurement because populations of models created using the old modelling method and EPReS co-existed in the same repository and were easily identified. The application of this measure to models created using the old modelling method showed a LOR of 11%. This triangulates well with previously published estimates of 8%.

8.4 Research Contribution

This research project produced three contributions to IS knowledge.

8.4.1 Theoretical Contributions

RQ1 provides a theoretical contribution by prescribing a new process modelling method (EPReS), which results in an increase in the LOR of complete process models, thereby improving the productivity of modellers and reducing the cost of projects in an organisation. This was achieved by focusing on factors impacting the reuse of process models and integrating them into the PA and the process modelling lifecycle. The relationships between the elements of EPReS were illustrated in an ERD.

RQ 2 provides a theoretical contribution to the understanding of the dynamics of business process modelling and the impact on the costs to the business. This was achieved by constructing a CLD based on elements and relationships identified in the literature and then validating these relationships by analysing semi-structured interviews with PMs. While these relationships have been described in the literature, this is the first time that their system dynamics have been considered.

RQ 3 provides a theoretical contribution to the measurement of the reuse of process models in a process repository. This measure was developed by drawing on and adapting software reuse measurements. While one of the purported benefits of using a process repository is the ability to reuse process models, levels of process model reuse in the literature are limited, and those that exist are based on interviews or questionnaires with practitioners. This contribution provides the first quantitative method of measuring the level of reuse of models in a repository that we are aware of.

8.4.2 Practical Contributions

DSR is concerned with the development of artifacts which are of practical use. An artifact which is not useful does not constitute knowledge. In RQ1, we designed a method which increased the LOR of complete process models, which reduces costs to the organisation in several ways:

- Reduced cost of creating or editing process models
    - By reusing existing process models and not recreating models unnecessarily.
    - By the automating of model and object management which removes the complexity and understanding required from the PM.
- Improved quality of process models because successive iterations of updating the same model also result in corrections that may be necessary being applied.
- Reduced cost of business analysis because the process model has been maintained and the model can be located.
- Reduced cost of repository maintenance

This research project also offers a practical contribution for process modelling software vendors. Studies of how well process modelling software supports reuse indicate that the reuse of process models is not well supported. The modelling method developed in this research project offers an opportunity for vendors to facilitate the reuse of process models in their software by adopting elements of the method which their software may not yet support.

8.5 Limitations of the study

Framing of the problem

This study could have been conducted using AR or even ADR. The purpose of AR is to collaborate with the organisation to better understand the problems, actions, and the reasons for the behaviour, to solve the problem (Collatto et al., 2018). AR specifically involves intervening in the organisational context to solve the problem. ADR, which is a combination of AR and DSR, seeks to combine elements of both methodologies. Typically, combining AR and DSR could be achieved by using AR to better understand the problem in the “Problem Identification” stage of a DSR research project, or by using AR to test or implement the artifact (e.g. a new method) in the “Evaluation” stage of a DSR Research Project (Sein et al., 2011). While collaborating with the organisation to identify the problem and requirements of a solution is attractive, there is limited benefit to this approach when the root cause of the problem is technical (object and model management in a repository as discussed in section 6.5.2), and if modellers do not fully understand the mechanics involved. This approach could lead to artifact requirements which do not resolve the root cause of the problems being experienced. Although using AR in the evaluation stage of DSR provides for the artifact to be improved based on evaluation results, it can be argued that this pattern is provided for in the DSR methodology used, because multiple design-build-evaluate cycles are provided for. The differentiating factor, in the view of this researcher, would be the level of collaboration involved with the participants.

Method Limitations

This study used a purposeful sampling approach where only those PMs who were in the team using the new modelling method were interviewed. This meant that data was only collected from respondents who had been exposed to both the new and the old modelling methods. It is possible that collecting data from a broader range of respondents may have unveiled additional elements and relationships relating to the reuse of process models.

The design principles adopted when creating models were based on SOA design principles. While increasing the size/responsibility of a process model will result in that model being reused more, this results in models which are more complex and difficult to maintain. This trade-off is subjective, and a different organisation may use slightly different guidelines, which would result in a different LOR of models for the same domain.

Contextual Limitations

This study was conducted in a single large financial services organisation using the ARIS modelling toolset. The proposed modelling method relies on ARIS's underlying architecture and automation capabilities to achieve the increased LOR realised in this study. While the proposed method has been generalised, it may not be possible to implement in every modelling tool available due to the underlying repository architecture and automation capabilities required of the modelling tool.

8.6 Future directions

This chapter has considered the limitations of this study and summarised the findings relating to the main research questions. The system dynamics of not reusing process models were explained using a CLD, a measure of the LOR was developed, and a modelling method which increases the LOR was also developed. This study lays the foundation for further research in these areas:

1. An opportunity to better understand the BA costs related to IS projects. A future project could further refine the BA costs based on the different project activities undertaken by a BA on a project. This would better quantify the benefits of reuse.
2. An opportunity to investigate the relationship between project requirements' quality, process modelling, and IS project success. This could further emphasise (or not) the importance of process modelling in the IS project lifecycle.
3. Opportunities to develop modelling methods which increase reuse when using modelling tools with different functionality to ARIS.

9 References

Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. *Int. J. Production Economics*, *90*(2), 129–149. https://doi.org/10.1016/S0925-5273(03)00102-6

Aldin, L., & de Cesare, S. (2011). A literature review on business process modelling: new frontiers of reusability. *Enterprise Information Systems*, *5*(3), 359–383. https://doi.org/10.1080/17517575.2011.557443

Almonte, L., Guerra, E., Cantador, I., & de Lara, J. (2022). Recommender systems in model-driven engineering: A systematic mapping review. *Software and Systems Modeling*, *21*(1), 249–280. https://doi.org/10.1007/s10270-021-00905-x

Almonte, L., Pérez-Soler, S., Guerra, E., Cantador, I., & De Lara, J. (2021). Automating the synthesis of recommender systems for modelling languages. *SLE 2021 - Proceedings of the 14th ACM SIGPLAN International Conference on Software Language Engineering*, 22–35. https://doi.org/10.1145/3486608.3486905

Alotaibi, Y. (2016). Business process modelling challenges and solutions: a literature review. *Journal of Intelligent Manufacturing*, *27*(4), 701–723. https://doi.org/10.1007/s10845-014-0917-4

Alter, S. (2004). Desperately Seeking Systems Thinking in the IS Discipline. *International Conference on Information Systems*, *25*, 757–769.

Awad, A., Sakr, S., Kunze, M., & Weske, M. (2011). Design by Selection: A Reuse-Based Approach for Business Process Modeling. In M. Jeusfeld, L. Delcambre, & TW. Ling (Eds.), *Conceptual Modeling – ER 2011. Lecture Notes in Computer Science, vol 6698* (pp. 332–345). Springer. https://doi.org/10.1007/978-3-642-24606-7_25

Barlas, Y. (1996). Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, *12*(3), 183–210.

Baumann, F., Hinkelmann, K., & Montecchiari, D. (2023). Supporting Reuse of Business Process Models by Semantic Annotation. *Lecture Notes in Business Information Processing*, *482*, 29–35. https://doi.org/10.1007/978-3-031-34985-0_3

Becker, M., & Laue, R. (2011). A comparative survey of business process similarity measures. *Computers in Industry*, *63*(2), 148–167. https://doi.org/10.1016/j.compind.2011.11.003

Beerepoot, I., Di Ciccio, C., Reijers, H. A., Rinderle-Ma, S., Bandara, W., Burattin, A., Calvanese, D., Chen, T., Cohen, I., Depaire, B., Di Federico, G., Dumas, M., van Dun, C., Fehrer, T., Fischer, D. A., Gal, A., Indulska, M., Isahagian, V., Klinkmüller, C., … Zerbato, F. (2023). The biggest business process management problems to solve before we die. *Computers in Industry*, *146*. https://doi.org/10.1016/j.compind.2022.103837

Beimborn, D., Martin, S. F., & Homann, U. (2005). Capability-oriented modeling of the firm. *Proceedings of the IPSI Conference, Amalfi, Italy*.

Belhajjame Khalid and Grigori, D. and H. M. and B. Y. M. (2017). Keyword-Based Search of Workflow Fragments and Their Composition. In R. and P. A. M. and C. J. Nguyen Ngoc Thanh and Kowalczyk (Ed.), *Transactions on Computational Collective Intelligence XXVI* (pp. 67–90). Springer International Publishing. https://doi.org/10.1007/978-3-319-59268-8_4

Bertalanffy, L. Von. (1972). The History and Status of General Systems Theory. *Journal General Systems Theory*, *15*(4), 407–426.

Biesta, G. (2010). Pragmatism and the philosophical foundations of mixed methods research. In A. Tashakkori & C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research. Second edition*. SAGE Publications.

Blaikie, N. W. H., & Priest, J. (2019). *Designing Social Research : The Logic of Anticipation* (3rd ed.). Polity Press.

Branco, M. C., Xiong, Y., Czarnecki, K., Kuster, J., & Volzer, H. (2014). A case study on consistency management of business and IT process models in banking. *Software & Systems Modeling*, *13*(3), 913–940. https://doi.org/10.1007/s10270-013-0318-8

Burlton, R., Rhyne, J., St George, D., Dugan, L., & McWhorter, N. (2019). *Similar Yet Different: Value Streams and Business Processes: The Business Architecture Perspective [White Paper]*.

Business Architecture Guild. (2024). *A Guide to the Business Architecture Body of Knowledge* (v13.0 BIZBOK Guide).

Chen, X., Cao, H., Ye, L., & Zhang, L. (2020). Fulfilling Functional Demands of BPM in Long-tailed Change Environments. *Proceedings of International Conference on Service Science, ICSS*, *2020-August*, 129–135. https://doi.org/10.1109/ICSS50103.2020.00028

Coghlan, D. (2019). *Doing Action research in our own organization.* (A. Owen, Ed.; 5th ed.). Sage Publications.

Cole, R., Purao, S., Rossi, M., & Sein, M. K. (2005). Being Proactive: Where Action Research meets Design Research. *Proceedings of the Twenty-Sixth International Conference on Information Systems*, 325–336.

Collatto, D. C., Dresch, A., Lacerda, D. P., & Bentz, I. G. (2018). Is Action Design Research Indeed Necessary? Analysis and Synergies Between Action Research and Design Science Research.

Systemic Practice and Action Research, *31*(3), 239–267. https://doi.org/10.1007/s11213-017-9424-9

Costa, D. M., Teixeira, E. N., & Werner, C. M. L. (2020). A Repository to Support Software Process Reuse Based on Process Lines. *Proceedings of the XIX Brazilian Symposium on Software Quality, SBQS '20*, 1–10. https://doi.org/10.1145/3439961.3439962

Cuesta, C., Ruesta, M., Tuesta, D., & Urbiola, P. (2015). The digital transformation of the banking industry. *BBVA Research*. https://www.bbvaresearch.com/wp-content/uploads/2015/08/EN_Observatorio_Banca_Digital_vf3.pdf

Cui, X. (2017). An approach implementing template-based process development on BPMN. *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, 239–244. https://doi.org/10.1109/ICIS.2017.7960000

Dalal, N. P., Kamath, M., Kolarik, W. J., & Sivaraman, E. (2004). Toward an integrated framework for modeling enterprise processes. *Communications of the ACM*, *47*(3), 83–87. https://doi.org/10.1145/971617.971620

Daneva, M. (1999). Measuring reuse in SAP requirements: a model-based approach. *Proceedings of the 1999 Symposium on Software Reusability - SSR '99*, 141–150. https://doi.org/10.1145/303008.303069

Daneva, M. (2000). Establishing reuse measurement practices in SAP requirements engineering. *International Conference on Requirements Engineering (ICRE'00)*, *June 2000*, 168–177. https://doi.org/10.1109/ICRE. 2000.855607

Davenport, T. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press.

Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, *58*(3), 358–380. https://doi.org/10.1016/j.datak.2005.07.007

Demetis, D. S., & Lee, A. S. (2016). Crafting theory to satisfy the requirements of systems science. *Information and Organization*, *26*(4), 116–126. https://doi.org/10.1016/j.infoandorg.2016.09.002

Deng, S., Wang, D., Li, Y., Cao, B., Yin, J., Wu, Z., & Zhou, M. (2017). A Recommendation System to Facilitate Business Process Modeling. *IEEE Transactions on Cybernetics*, *47*(6), 1380–1394. https://doi.org/10.1109/TCYB.2016.2545688

Dijkman, R., Rosa, M. La, & Reijers, H. A. (2012). Managing large collections of business process models - Current techniques and challenges. *Computers in Industry*, *63*(2), 91–97. https://doi.org/10.1016/j.compind.2011.12.003

Dijkman, R., Vanderfeesten, I., & Reijers, H. A. (2016). Business process architectures: overview, comparison and framework. *Enterprise Information Systems*, *10*(2), 129–158. https://doi.org/10.1080/17517575.2014.928951

Dumas, M., García-Bañuelos, L., & Dijkman, R. (2009). Similarity search of business process models. *IEEE Data Eng. Bull.*, *32*(3), 23–28. https://pure.tue.nl/ws/files/3329300/Metis232408.pdf

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of business process management*. Springer. https://doi.org/10.1007/978-3-642-33143-5

Erl, Thomas. (2008). SOA : principles of service design. In *SOA : principles of service design*. Prentice Hall.

Erol, S. (2018). *A process model of business process model reuse*. https://www.researchgate.net/profile/Selim-Erol/publication/296486911_A_process_model_of_business_process_model_reuse/links/5a8409d0aca272d6501f609e/A-process-model-of-business-process-model-reuse.pdf

Eshuis, R. (2023). Extracting Reusable Fragments From Data-Centric Process Variants. *IEEE Transactions on Services Computing*, *16*(3), 1833–1845. https://doi.org/10.1109/TSC.2022.3190987

Fantinato, M., de Toledo, M., Thom, L. H., Gimenes, I. M., Rocha, R., & Garcia, D. (2012). A survey on reuse in the business process management domain. *Int. J. Business Process Integration and Management*, *6*(1), 52–76.

Federation of Enterprise Architecture Professional Organizations (FEAPO) Taxonomy Working Group. (2017). *FEAPO Enterprise Architecture Definitions*. https://cdn.ymaws.com/www.businessarchitectureguild.org/resource/resmgr/docs/feapo_adopted_architecture_d.pdf

Fellmann, M., Koschmider, A., & Schoknecht, A. (2014). Analysis of business process model reuse literature: Are research concepts empirically validated? *Modellierung*, 185–192.

Fenton, N. E., & Bieman, J. (2015). *Software Metrics: A Rigorous and Practical Approach* (3rd ed.). CRC Press.

Fenton, N. E., Kitchenham, B., & Pfleeger, S. L. (1995). Towards a Framework for Software Measurement Validation. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, *21*(12), 929–944. https://doi.org/10.1109/32.489070

Ferrucci, F., Tortota, G., & Vitiello, G. (2002). Exploiting Visual Languages In Software Engineering. In S.-K. Chang (Ed.), *Handbook of Software Engineering and Knowledge Engineering* (pp. 53–76). World Scientific Pub Co Inc. https://doi.org/10.1142/9789812389701_0003

Figl, K., Koschmider, A., & Kriglstein, S. (2013). Visualising Process Model Hierarchies. *ECIS*, *2013*, 180.

Foddy, W. (1994). *Constructing Questions for Interviews and Questionaires: Theory and Pracice in Social Research*. Cambridge University Press.

Forrester, J., W., & Senge, P. (1980). Tests for building confidence in system dynamics models. In A. Legasto, A., W. Forrester, J., & M. Lyneis, J. (Eds.), *System Dynamics* (pp. 209–228). North-Holland.

Frakes, W. B., Tech, V., & Terry, C. (1996). Software Reuse: Metrics and Models. *ACM Computing Surveys (CSUR)*, *28*(2), 415–435.

François, P. A., Kampmann, M., Plattfaut, R., & Coners, A. (2023). Systematically embedding automation reuse in business process management projects. *Projektmanagement Und Vorgehensmodelle 2023-Nachhaltige IT-Projekte*.

François, P. A., & Plattfaut, R. (2023). The Reuse of Business Process Automation Artefacts. In *INFORMATIK 2023 - Designing Futures: Zukünfte gestalten* (pp. 1923–1942). Gesellschaft für Informatik e.V. https://doi.org/10.18420/inf2023_193

François, P. A., & Plattfaut, R. (2024). Designing the Organizational Reuse Environment - Enabling Citizen Developers to Reuse Process Automation Artifacts. *Lecture Notes in Business Information Processing*, *527 LNBIP*, 138–153. https://doi.org/10.1007/978-3-031-70445-1_9

Gleasure, R. (2015). When is a problem a design science problem? *An International Journal on Information Technology, Action, Communication and Workpractices*, *9*(1), 9–25.

Goldkuhl, G. (2012). Pragmatism vs interpretivism in qualitative information systems research. *European Journal of Information Systems*, *2*(21), 135–146. https://doi.org/10.1057/ejis.2011.54

Goldstein, M., & Gonzalez-Alvarez, C. (2021). Augmenting Modelers with Semantic Autocompletion of Processes. *Business Process Management Forum: BPM Forum 2021, Rome, Italy, September 06–10*, 20–36. http://arxiv.org/abs/2105.11385

Görsch, D. (2002). Multi-Channel Integration and Its Implications for Retail Web Sites. *Ecis*, *2002*, 748–758.

Gregor, S. (2006). The Nature of Theory in Information Systems. *MIS Quarterly*, *30*(3), 611–642.

Gregor, S. (2009). Building theory in the sciences of the artificial. In V. Vaishnavi & S. Purao (Eds.), *Proceedings of the 4th International Conference on Design Science Research*. ACM. https://doi.org/10.1145/1555619.1555625

Gregor, S., & Jones, D. (2007). The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, *8*(5), 312–335.

Gruhn, V., & Laue, R. (2006). Complexity Metrics for Business Process Models. In W. Abramowicz & H. C. Mayer (Eds.), *9th International Conference on Business Information Systems (BIS 2006)* (pp. 1–12).

Hallerbach, A., Bauer, T., & Reichert, M. (2008). Managing Process Variants in the Process Life Cycle. *10th Int'l Conf. on Enterprise Information Systems (ICEIS'08)*, 154–161.

Hallerbach, A., Bauer, T., & Reichert, M. (2010a). Capturing variability in business process models: The Provop approach. *Journal of Software Maintenance and Evolution*, *22*(6–7), 519–546. https://doi.org/10.1002/smr.491

Hallerbach, A., Bauer, T., & Reichert, M. (2010b). Configuration and Management of Process Variants. In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1* (pp. 237–255).

Hanson, W. E., Creswell, J. W., Plano Clark, V. L., Petska, K. S., & Creswell, J. D. (2005). Mixed Methods Research Designs in Counseling Psychology. *Journal of Counseling Psychology*, *52*(2), 224–235.

Herhausen, D., Binder, J., Schoegel, M., & Herrmann, A. (2015). Integrating Bricks with Clicks: Retailer-Level and Channel-Level Outcomes of Online-Offline Channel Integration. *Journal of Retailing*, *91*(2), 309–325. https://doi.org/10.1016/j.jretai.2014.12.009

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2), 87–92. http://community.mis.temple.edu/seminars/files/2009/10/Hevner-SJIS.pdf

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information systems research. *MIS Quarterly*, *28*(1), 75–105.

Homann, U. (2006). *A Business-Oriented Foundation for Service Orientation. Microsoft Corporation*. https://cdn.ymaws.com/www.businessarchitectureguild.org/resource/resmgr/homann_article_on_capabiliti.pdf

Iivari, J. (2007). A Paradigmatic Analysis of Information Systems As a Design Science. *Scandinavian Journal of Information Systems*, *19*(2), 39–64. https://doi.org/10.1.1.218.2636

Iivari, J. (2020). Editorial: A Critical Look at Theories in Design Science Research. *Journal of the Association for Information Systems*, *21*(3), 502–519.

Indulska, M., Green, P., Recker, J., & Rosemann, M. (2009). Business Process Modeling: Perceived Benefits. *International Conference on Conceptual Modeling*, *November*, 458–471. https://doi.org/10.1007/978-3-642-04840-1

Indulska, M., Recker, J., Rosemann, M., & Green, P. (2009). Business Process Modeling: Current Issues and Future Challenges. *International Conference on Advanced Information Systems Engineering*, *June*, 501–514. https://doi.org/10.1007/978-3-642-02144-2

Javidan, M. (1998). Core competence: What does it mean in practice? *Long Range Planning*, *31*(1), 60–71. https://doi.org/https://doi.org/10.1016/S0024-6301(97)00091-5

Jonnavithula, L., Antunes, P., & Cranefield, J. (2015). Organisational Issues in Modelling Business Processes: An Activity-Based Inventory and Directions for Research. *PACIS*, 184.

Kajornboon, A. B. (2008). *Using interviews as research instruments*. https://doi.org/Available at http://www.culi.chula.ac.th/research/e-journal/bod/annabel.pdf (Accessed 12 November 2017)

Kesari, M., Chang, S., & Seddon, P. (2003). A content-analytic study of the advantages and disadvantages of process modelling. *ACIS 2003 Proceedings*, *2*(November), 1–11.

Khider, H., Hammoudi, S., & Meziane, A. (2020). Business Process Model Recommendation as a Transformation Process in MDE: Conceptualization and First Experiments. *International Conference on Model-Driven Engineering and Software Development*, 65–75. https://doi.org/10.5220/0009155600650075

Kim, Y.-G. (1995). Process Modeling For BPR: Event-Process Chain Approach. *16th International Conference on Information Systems*, *11*, 109–121.

Koppenhagen, N., Gaß, O., & Müller, B. (2012). Design Science Research in Action - Anatomy of Success Critical Activities for Rigor and Relevance. *20th European Conference on Information Systems (ECIS 2012), Barcelona, Spain, June 10-13, 2012 ; Poster Presentation*, 1–12. https://madoc.bib.uni-mannheim.de/31687/

Koschmider, A., Fellmann, M., Schoknecht, A., & Oberweis, A. (2014). Analysis of process model reuse: Where are we now, where should we go from here? *Decision Support Systems*, *66*, 9–19. https://doi.org/10.1016/j.dss.2014.05.012

Krogstie, J., Dalberg, V., & Jensen, S. M. (2006). Increasing the Value of Process Modelling. *ICEIS*, *3*, 70–77.

Kumar, A., & Yao, W. (2012). Design and management of flexible process variants using templates and rules. *Computers in Industry*, *63*(2), 112–130. https://doi.org/10.1016/j.compind.2011.12.002

La Rosa, M., Jin, T., Wang, J., ter Hofstede, A., & Wen, L. (2010). Efficient Querying of Large Process Model Repositories. *International Conference on the Move to Meaningful Internet Systems*, 402–409. https://doi.org/10.1016/j.compind.2012.09.008

Lane, D. C. (2008). The emergence and use of diagramming in system dynamics. *Systems Research and Behavioral Science*, *25*, 3–23. https://doi.org/10.1002/sres.826

Larsen, K. R., Lukyanenko, R., Mueller, R. M., Storey, V. C., Robinson, J. M., Parsons, J., Vandermeer, D., & Hovorka, D. S. (2020). Validity in Design Science Research. *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry: 15th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2020*, 272–282.

Lewis, J., Whysall, P., & Foster, C. (2014). Drivers and Technology-Related Obstacles in Moving to Multichannel Retailing. *International Journal of Electronic Commerce*, *18*(4), 43–67. https://doi.org/10.2753/JEC1086-4415180402

Lockie, W. (2014). Delivering an effective click-and-collect strategy - A retailer case study. *Journal of Digital & Social Media Marketing*, *2*(2), 139–152.

Macintosh, A. L. (1993). The need for enriched knowledge representation for enterprise. *AI (Artificial Intelligence) in Enterprise Modelling, IEE Colloquium on (Digest No.078)*, 1–3.

Makni, L., Haddar, N. Z., & Ben-Abdallah, H. (2018). An automated method for the construction of semantic business process patterns. *Int. J. Process Management and Benchmarking*, *8*(3), 263–290. https://doi.org/https://doi.org/10.1504/IJPMB.2018.092889

Malinova, M., Leopold, H., & Mendling, J. (2013). An Empirical Investigation on the Design of Process Architectures. *Wirtschaftsinformatik.*

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 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 Quarterly*, *32*(4), 725–730.

Markovic, I., & Pereira, A. C. (2007). Towards a Formal Framework for Reuse in Business Process Modeling. *International Conference on Business Process Management*, 484–495. https://doi.org/10.1007/978-3-540-78238-4_49

Marshall, M. N. (1996). Sampling for qualitative research. *Family Practice*, *13*(6), 522–525. https://doi.org/10.1093/fampra/13.6.522

Menrad, M. (2020, June 22). Systematic review of omni-channel banking and preview of upcoming developments in Germany. *Innovative Marketing*, *16*(2), 104–125. https://doi.org/10.21511/im.16(2).2020.09

Nagm-Aldeen, Y., Abdel-Fattah, M. A., & El-Khedr, A. (2015). A Literature Review of Business Process Modeling Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(3), 43–47.

Narendra, N. C., Ponnalagu, K., Gangadharan, G. R., Truong, H. L., Dustdar, S., & Ghose, A. K. (2012). Effective reuse via modeling, managing and searching of business process assets. *Services Computing, (SCC), 2012 IEEE Ninth International Conference On*, *3*, 462–469. https://doi.org/10.1109/SCC.2012.46

Nolte, A., Bernhard, E., & Recker, J. (2013). You’ve modelled and now what?"-exploring determinants of process model re-use. *24th Australasian Conference on Information Systems (ACIS)*, 1–11. https://doi.org/10.3127/ajis.v18i3.1094

Nolte, A., Bernhard, E., Recker, J., Pittke, F., & Mendling, J. (2016). Repeated use of process models: The impact of artifact, technological and individual factors. *Decision Support Systems*, *88*, 98–111. https://doi.org/10.1016/j.dss.2016.06.002

Nunamaker, J. F., Chen, M. C., & Purdin, T. D. M. (1990). Systems Development in Information Systems Research. *Journal of Management Information Systems*, *7*(3), 89–106. https://doi.org/10.1287/isre.

Nuthmann, C. (1994). Using human judgment in system dynamics models of social systems. *System Dynamics Review*, *10*(1), 1–19. https://doi.org/10.1002/sdr.4260100102

Ouyang, C., Dumas, M., ter Hofstede, A. H., & van der Aalst, W. M. (2008). Pattern-based Translation of BPMN Process Models to BPEL Web Services. *International Journal of Web Services Research*, *5*, 42–61.

Peffers, K., Rothenberger, M. A., Tuunanen, T., & Vaezi, R. (2012). *Design Science Research Evaluation* (pp. 398–410). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29863-9_29

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Bragge, J., & Virtanen, V. (2006). The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. *Journal of Management Information Systems*, *24*(3), 24. https://doi.org/10.2753/MIS0742-1222240302

Peffers, K., Tuure, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Polyvyanyy, A., Pika, A., & ter Hofstede, A. H. M. (2020). Scenario-based process querying for compliance, reuse, and standardization. *Information Systems*, *93*. https://doi.org/10.1016/j.is.2020.101563

Poulin, J. S. (1997). *Measuring software reuse: principles, practices, and economic models*. Addison-Wesley.

Prahalad, C. K., & Hamel, G. (2003). *The Core Competence of the Corporation*. www.hbr.org

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for Design Science Research Evaluation. *ECIS 2008*. http://aisel.aisnet.org/ecis2008

Radgui, M. a, Saidi, R. b, & Mouline, S. a. (2013). Design for reuse in business process: Method and experiments. In *International Journal of Enterprise Information Systems* (Vol. 9, Issue 4, pp. 12–27). https://doi.org/10.4018/ijeis.2013100102

Raduescu, C., Tan, H. M., Jayaganesh, M., Bandara, W., zur Muehlen, M., & Lippe, S. (2006). A Framework of Issues in Large Process Modeling Projects. *ECIS*, *2006*, 1594–1605.

Recker, J. C. (2006). Process Modeling in the 21st Century. *BPTrends*, *5*, 1–6. https://doi.org/10.1016/0026-2714(85)90617-1

Recker, J. C., Rosemann, M., Indulska, M., & Green, P. (2009). Business process modeling: a comparative analysis. *Journal of the Association for Information Systems*, *10*(4), 333–363.

Reijers, H. A., Mans, R. S., & van der Toorn, R. A. (2009). Improved model management with aggregated business process models. *Data & Knowledge Engineering*, *68*(2), 221–243. https://doi.org/10.1016/j.datak.2008.09.004

Rosemann, M. (2006). Potential pitfalls of process modeling: part B. *Business Process Management Journal*, *12*(3), 377–384. https://doi.org/10.1108/14637150610668024

Rosemann, M., Green, P., Indulska, M., & Recker, J. (2009). Using Ontology for the Representational Analysis of Process Modelling Techniques. *International Journal of Business Process Integration and Management*, *4*(4), 251–265. https://doi.org/10.1504/IJBPIM.2009.032282

Rylander, A. (2012). Pragmatism and design research–An overview. *Ingår i Designfakultetens Serie Kunskapssammanställningar*.

Saarsen, T., & Dumas, M. (2016). Factors Affecting the Sustained Use of Process Models. *International Conference on Business Process Management*, 193–209.

Saunders, M., Lewis, P., & Thornhill, A. (2023). *Research methods for business students* (9th ed). Pearson.

Savary-Leblanc, M., Le Pallec, X., & Gérard, S. (2024). Understanding the need for assistance in software modeling: interviews with experts. *Software and Systems Modeling*, *23*(1), 103–135. https://doi.org/10.1007/s10270-023-01104-6

Sayuri, C., Amaral, T., Rozenfeld, H., Mascarenhas, J., Costa, H., De Andrade Magon, M. de F., & Mascarenhas, Y. M. (2011). Improvement of radiology services based on the process management approach. *European Journal of Radiology*, *78*, 377–383. https://doi.org/10.1016/j.ejrad.2010.12.025

Seck, A. M. (2013). The Issue of Multichannel Integration, a Key Challenge for Service Firms in a Context of Multichannel Services Distribution. *International Business Research*, *6*(2), 160–167. https://doi.org/10.553

Seck, A. M., & Philippe, J. (2011). Service encounter in multi-channel distribution context: virtual and face-to-face interactions and consumer satisfaction. *The Service Industries Journal*, *33*(April 2015), 1–15. https://doi.org/10.1080/02642069.2011.622370

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly*, *35*(1), 37–56. http://www.jstor.org/stable/23043488

Shahzad, K., Elias, M., & Johannesson, P. (2010). Requirements for a business process model repository: A stakeholders' perspective. *Lecture Notes in Business Information Processing*. https://doi.org/10.1007/978-3-642-12814-1_14

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). MIT Press.

Smirnov, S., Weidlich, M., Mendling, J., & Weske, M. (2012). Action patterns in business process model repositories. *Computers in Industry*, *63*(2), 98–111. https://doi.org/10.1016/j.compind.2011.11.001

Smith, K. J., & Weistroffer, H. R. (2016). Systems Theory : Should information researchers even care? *SAIS 2016 Proceedings*, 1–6.

Soffer, P., Reinhartz-berger, I., & Sturm, A. (2007). Facilitating Reuse by Specialization of Reference Models for Business Process Design. *8th Workshop on Business Process Modeling, Development, and Support (BPMDS07), Proceedings of CAiSE'07 Workshops*, 1–9. http://lams.epfl.ch/conference/bpmds07/program/Soffer_5.pdf

Sola, D., Meilicke, C., Van Der Aa, H., & Stuckenschmidt, H. (2021). A Rule-based Recommendation Approach for Business Process Modeling. *International Conference on Advanced Information Systems Engineering*, 328–343.

Sola, D., & Stuckenschmidt, H. (2021). Towards a Rule-based Recommendation Approach for Business Process Modeling. *International Conference on Service-Oriented Computing*, 25–31. https://doi.org/https://doi.org/10.1007/978-3-030-76352-7_4

Sonnenberg, C., & vom Brocke, J. (2012). Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. *International Conference on Design Science Research in Information Systems*, 381–397. https://doi.org/10.1007/978-3-642-29863-9_28

Terry, C. (1993). *Analysis and implementation of software reuse measurement*. Virginia Polytechnic Institute and State University, Master's Project and Report.

The Federation of Enterprise Architecture Professional Organizations. (2017). *The Federation of Enterprise Architecture Definitions*. https://cdn.ymaws.com/www.businessarchitectureguild.org/resource/resmgr/docs/feapo_adopted_architecture_d.pdf

Trabelsi, F. Z., Khtira, A., & El Asri, B. (2021). Towards an Approach of Recommendation in Business Processes Using Decision Trees. *Proceedings - 2021 International Symposium on Computer Science and Intelligent Controls, ISCSIC 2021*, 341–347. https://doi.org/10.1109/ISCSIC54682.2021.00068

Tran, H. N., Coulette, B., & Vu, D. T. (2011). Automatic Reuse of Process Patterns in Process Modeling. *Proceedings of the 2011 ACM Symposium on Applied Computing*, 1431–1438. https://doi.org/10.1145/1982185.1982494

Vaishnavi, V., & Kuechler, B. (2004). *Design Science Research in Information Systems*. https://doi.org/10.1007/978-1-4419-5653-8 van der Aalst, W. M. P. (2013). Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, *2013*. https://doi.org/10.1155/2013/507984

Van Eijndhoven, T., Iacob, M. E., & Ponisio, M. L. (2008). Achieving business process flexibility with business rules. *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, 95–104. https://doi.org/10.1109/EDOC.2008.23

Van Nuffel, D., & De Backer, M. (2012). Multi-abstraction layered business process modeling. *Computers in Industry*, *63*(2), 131–147. https://doi.org/10.1016/j.compind.2011.12.001

Veitch, R., & Seymour, L. F. (2019). Measuring Business Process Model Reuse in a Process Repository. In Di Francescomarino C., R. Dijkman, & U. Zdun (Eds.), *Business Process Management Workshops. BPM 2019. Lecture Notes in Business Information Processing, vol 362*. Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-030-37453-2_58

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2012). A comprehensive framework for evaluation in design science research. *International Conference on Design Science Research in Information Systems*, 423–438. https://doi.org/10.1007/978-3-642-29863-9

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2017). Choosing a design science research methodology. *ACIS 2017 Proceedings*, *112*. https://aisel.aisnet.org/acis2017/112/

Verhoef, P. C., Kannan, P. K., & Inman, J. J. (2015). From Multi-Channel Retailing to Omni-Channel Retailing. Introduction to the Special Issue on Multi-Channel Retailing. *Journal of Retailing*, *91*(2), 174–181. https://doi.org/10.1016/j.jretai.2015.02.005

Weaver, W. (1948). Science and Complexity. *American Scientist*, *36*, 536–544.

Web Services Architecture Working Group. (2004, February 11). *Web Services Glossary*. W3C Working Group Note 11 February. https://www.w3.org/TR/ws-gloss/

Weber, B., Reichert, M., Mendling, J., & Reijers, H. A. (2011). Refactoring Large Process Model Repositories. *Computers in Industry*, *62*(5), 467–486.

Whittle, R., & Myrick, C. B. (2016). *Enterprise business architecture: The formal link between strategy and results*. CRC Press.

Yongchareon, S., Liu, C., & Zhao, X. (2020). Reusing artifact-centric business process models: a behavioral consistent specialization approach. *Computing*, *102*(8), 1843–1879. https://doi.org/10.1007/s00607-020-00798-6

Zhao, W., Hauser, R., Bhattacharya, K., Bryant, B. R., & Cao, F. (2006). Compiling business processes: untangling unstructured loops in irreducible flow graphs. *International Journal of Web and Grid Services*, *2*(1), 68–91. https://doi.org/10.1504/IJWGS.2006.008880

Zhu, R., Huang, Y., Liu, L., Zhou, W., Zhang, X., Chen, Y., & Cai, L. (2024). Business Process Retrieval from Large Model Repositories for Industry 4.0. *IEEE Transactions on Services Computing*, *17*(1), 306–321. https://doi.org/10.1109/TSC.2023.3348294

zur Muehlen, M., Wisnosky, D. E., & Kindrick, J. (2010). Primitives: Design Guidelines and Architecture for BPMN Models. *Proceedings of the 21st Australasian Conference on Information Systems, ACIS 2010, Brisbane, Australia, December 1-3*.

10 Appendices

11 Appendix 1: Repository Data Structure

This research project was carried out in a large South African bank that uses ARIS as the process repository. In the ARIS modelling tool, all process information is stored in a relational database. This section describes the ARIS database schema and the associated terminology.

Repository Database Schema

A simplified database schema of a repository with key relationships is shown in Figure 11.1, where both models and Functions (steps in the model) are stored as objects in the database. The database also stores a set of relationships between these objects, which defines both the hierarchy (relationships between Functions and models) and the process flow (relationships between Functions).

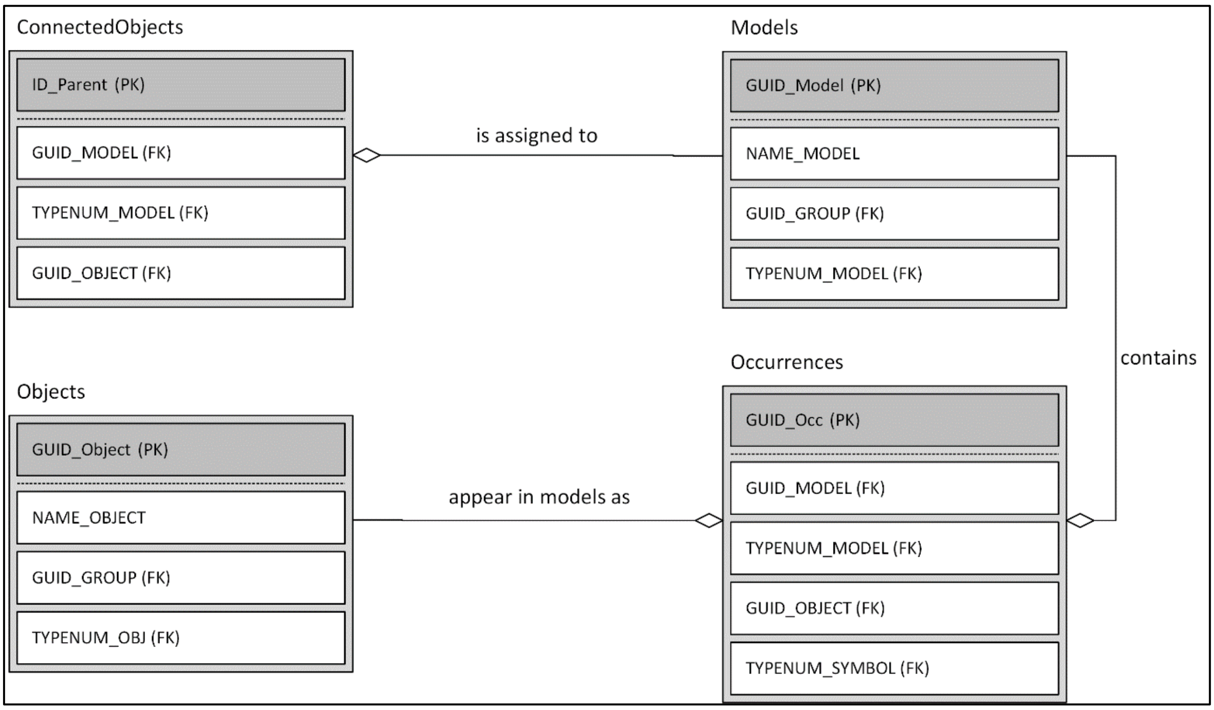


Figure 11.1 Simplified process repository ERD

The terminology used in this research must be explained as it forms the basis of the discussion and approach to measuring the reuse of business process models:

- A Function exists as an object of type "Function" in the process repository. A Function can be an individual activity or task at a granular level (e.g., capture customer name) or an abstraction of another process (e.g., authenticate customer). If a Function is an abstraction of another process model, that process model is said to be "assigned to" that Function (Figure 11.2a). Alternately, it can be said that Function F2 decomposes into M11. In terms of Figure 11.1, such a Function is considered to be a "Connected Object".

- A model represents a process and consists of Functions, which can be individual activities/tasks at a granular level or abstractions of other processes.
- An object (e.g., a Function) contains all the attributes and attribute values for a specific object. For example, the attribute "Name" contains the name of the object.
- An object (e.g. a Function) appears in a model as an "Occurrence" (Figure 11.2b). The occurrence in a model is a pointer to the object in the repository. It is essential to understand that a specific object can occur multiple times in the same model or even in multiple models.
- All occurrences of Functions in a model are represented by a symbol selected by the PM. The same Function can occur as different symbols in different models.
- The parent object of a model is the object to which a model is assigned. In Figure 11.2a, object F2 is the parent of model M11.

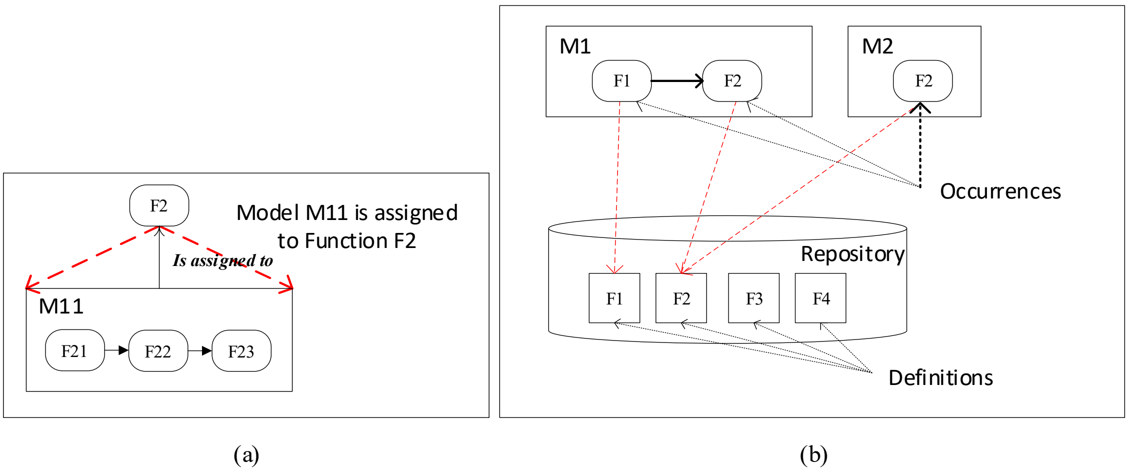


Figure 11.2. Objects and Occurrences

Significant reuse opportunities arise in repositories which have this structure because updating a process model M, which is assigned to a Function F, will result in all other process models in which Function F occurs, also adopting that change. In Figure 11.3, by changing the Function F5 in model M1, the object F5 in the repository is changed. It can be seen that model M2 also has an occurrence of the object F5, and therefore, the change made to F5 is also reflected in model M2.

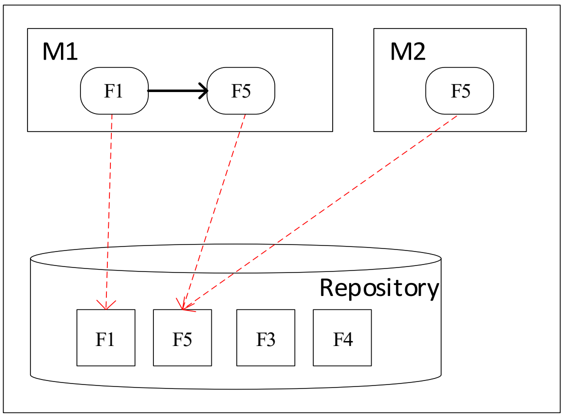


Figure 11.3 Object reuse in models

The PA typically defines the high-level process grouping found in the organization and provides guidelines regarding the decomposition of such processes into levels of increasing granularity (Dijkman et al., 2016; Malinova et al., 2013; zur Muehlen et al., 2010). The number of levels of decomposition adopted depends on each organization, but typically, 3-5 levels of decomposition are used, as shown in Figure 11.4 for three levels of decomposition (Van Nuffel & De Backer, 2012). Level 0 is the highest level (most abstract), while Level 5 would contain the most granular process models.

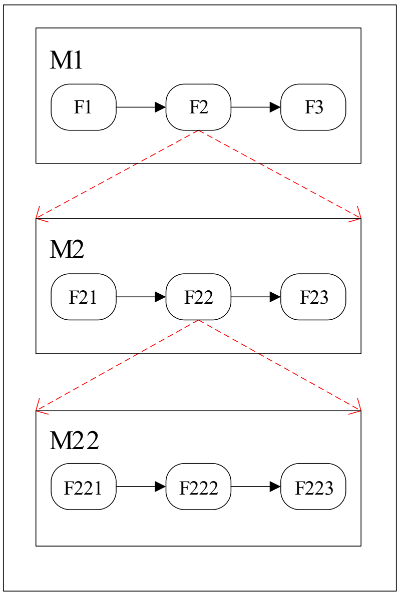


Figure 11.4. Business Process Architecture Hierarchy

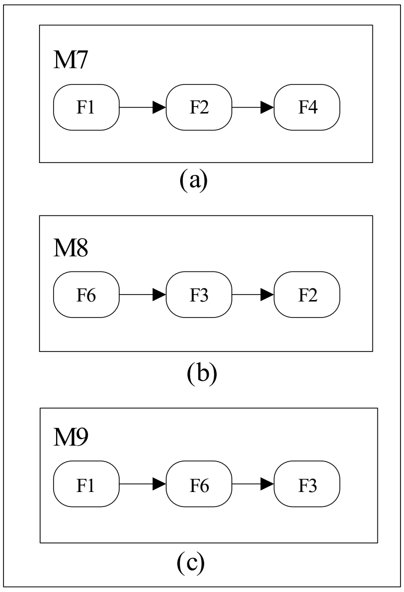


Figure 11.5. Complete process model reuse

Using Figure 11.4 and Figure 11.5, the concept of complete process model reuse is illustrated. While Function F2 is an abstraction of process model M2, F2 is used in both models M7 and M8. Similarly, F1 is used in M7 and M9. A change to the underlying model M2 will change the meaning of F2, and therefore, the models M7 and M8 will also be impacted. Therefore, the number of times M2 has been used can be calculated by determining the number of occurrences of F2 in models in the repository.

Types of Models used by the research organisation.

While the modelling tool provides a wide range of model types that can be used by an organisation, most organisations only use a subset of the model types that are available. The model types (and the number of such models) used in the research organisation are indicated in Table 11.1. The focus of this research project is BPM and, specifically, control flow models, which depict the series of activities required to produce the desired outcome. The relevant control flow models of interest to this study are also indicated in Table 11.1. All model types and counts are reflected because they were used to validate the data transfer to SQL Server.

Table 11.1: Model Types and number of models in the repository

Model Type	Repository Count	Control Flow Model?
Access diagram	34	
Application collaboration diagram	1	
Application system type diagram	11	
BPMN collaboration diagram (BPMN 2.0)	1691	Yes
Business controls diagram	1734	
Business rule architecture diagram	2	
Class diagram	1	
Customer journey landscape	5	
Customer journey map	19	
Customer segmentation map	2	
Customer touchpoint allocation diagram	4	
Enterprise BPMN collaboration diagram	450	Yes
EPC	20214	Yes

Model Type	Repository Count	Control Flow Model?
EPC (column display)	1877	Yes
EPC (row display)	13	Yes
Function allocation diagram	35295	
Function tree	108	
IE Data model	3	
Knowledge structure diagram	9	
KPI allocation diagram	19	
Matrix model	11	
Objective diagram	6	
Organizational chart	90	
Process selection diagram	255	
Process selection matrix	12	
Product tree	1	
Product/Service tree	356	
Quick model	125	
Risk diagram	50	
Screen design	1	
Screen navigation	20	
Service allocation diagram	139	
Service architecture diagram	5	
Service collaboration diagram	17	
Spreadsheet	5	
Task allocation diagram	1	
Technical terms model	11	
UML 1.4 Use case diagram	1	
Value-added chain diagram	248	Yes
Total	62846	

Types of Objects used by the research organisation.

Similarly, the modelling tool provides a wide range of object types, of which most organisations use only a subset of those available. The object types used (and the number of such objects) are indicated in Table 11.2. This study is concerned with only one type of object, known in the modelling tool as a **Function**. A Function object type is used to represent an activity or step in the modelling tool in all relevant model types. All object types and counts are reflected because they were used to validate the data transfer to SQL Server.

Table 11.2: Object types and number of objects in the repository

Object Type	ARIS Object Count	Relevant to this study
Application system	2	
Application system class	3	
Application system type	1000	
Business rule	6729	
Capability	305	
Class	1028	
Cluster/Data model	5449	
Component	24	
Customer journey	22	
Customer journey step	66	
Customer lifecycle stage	21	
Customer segment	1	
Customer touchpoint	62	
Distribution channel	49	
Documented knowledge	743	
Emotion	277	
ERM attribute	12	
Event	176644	
Function	185598	Yes
Group	518	
Information carrier	18174	

Object Type	ARIS Object Count	Relevant to this study
IT function type	5	
KPI instance	1250	
Lane	2169	
List	12	
Location	43	
Main process	12	
Message	163	
Objective	103	
Organizational unit	2756	
Participant	824	
Person	143	
Persona	6	
Policy	1	
Position	11	
Product/Service	948	
Questionnaire template	3	
Quick object	2542	
Requirement	872	
Risk	2874	
Risk category	79	
Role	6325	
Rule	80761	
Screen	5087	
Screen design	242	
Service type	495	
Structural element	1045	
Success factor	47	
System organizational unit	1	

Object Type	ARIS Object Count	Relevant to this study
Task	37	
Technical term	4433	
Text annotation	1338	
Total	511354	

Symbols

In the modelling tool, an object occurs in a model as a **Symbol**. The symbol used in a particular model depends on the purpose of the object in that model and on the modelling method adopted by the organisation. For example, a Function may appear in a BPMN Process model as a User Task or a Script Task. In a Value Chain model, that same Function would appear as a Value-chain symbol.

12 Appendix 2: Model Hierarchy and Model Reuse

Figure 12.1 illustrates the concept of complete process model reuse and PA level.

The organisation where the research was conducted uses a 5-level PA, with level 5 process models being the most detailed. In Figure 12.1, Model C would be considered a Level 5 process. Model C is assigned to an object C, which in turn can be used in another process model. In Figure 12.1, it can be seen that object C (representing Model C) forms part of Model X. As a result, Model X is considered a Level 4 process model architecturally. Extending the explanation, Figure 12.1 also shows that Model X is assigned to object X, which in turn forms part of Model Z. Architecturally, Model Z is considered a Level 3 process model.

An important point to note is that objects/models belonging to a higher PA level represent larger amounts of "reuse" than those at a lower level. For example, the object X that is used in Model Z represents the reuse of processes A, B and C, and therefore, when measuring complete process model reuse, this should be accounted for (Daneva, 1999, 2000)

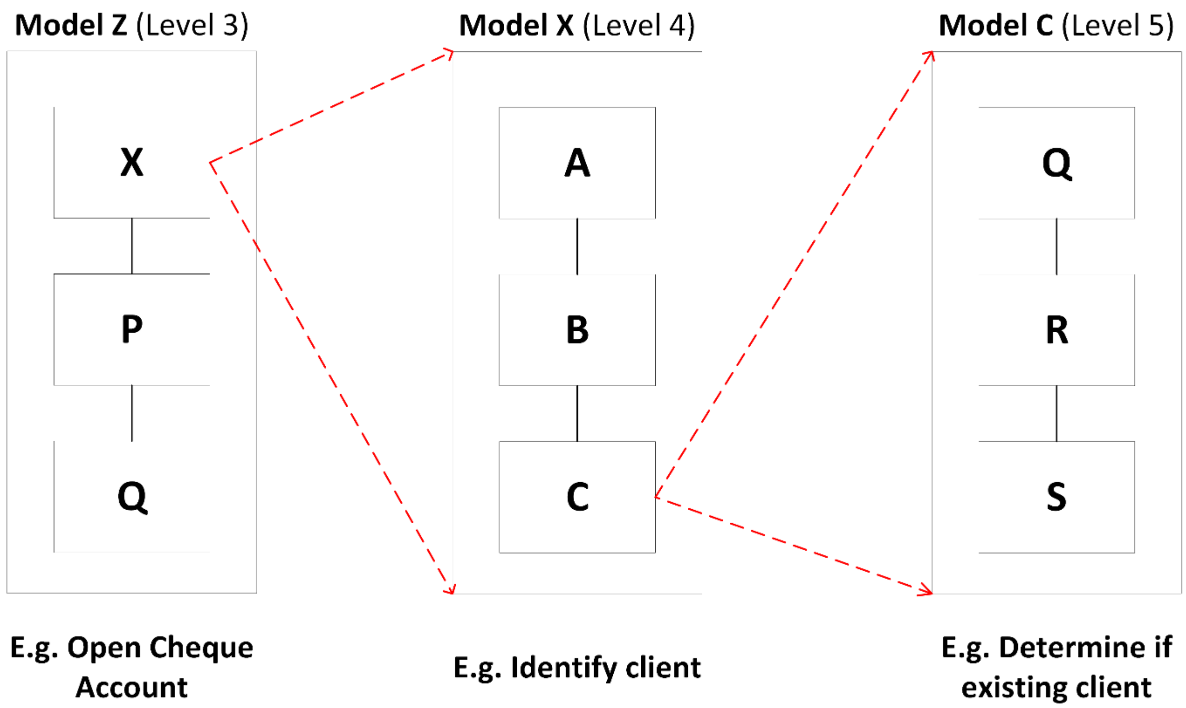


Figure 12.1: Architecture levels and model reuse

13 Appendix 3: Quantitative Data Generation Process

Figure 13.1 depicts the process used to generate and analyse the quantitative data relating to process model reuse.

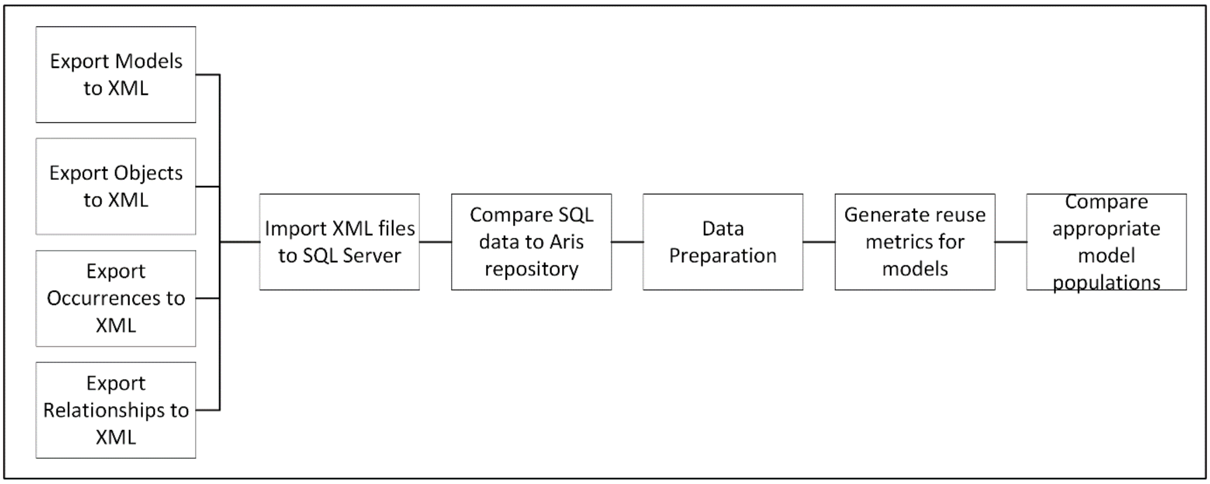


Figure 13.1: Data processing process used for quantitative data generation and analysis.

Table 13.1 describes the steps followed to generate the quantitative data in further detail.

Table 13.1: Quantitative model reuse data generation

Step Name	Activity	Comment
Export models to XML	Export the model data from the repository into XML files using in-built scripting functionality.	All the models in the repository were exported, including the following key information related to each model: Model Name, Model Group, Model GUID, Model Type Number, and Model Type. The structure of the XML file generated is depicted in the XSD file in Appendix 10
Export objects to XML	Export the object data from the repository into XML files using in-built scripting functionality.	All the objects in the repository were exported, including the following key information related to each object: Object Name, Object Group, Object GUID, Object Type Number, and Object Type. The structure of the XML file generated is depicted in the XSD file in Appendix 11
Export occurrences to XML	Export the occurrence data from the repository into	All the occurrences in each model were exported, including the following key information related to each occurrence: Model Name, Model GUID,

Step Name	Activity	Comment
	XML files using in-built scripting functionality.	Model Type Number, Model Type, Object Type Number, Object Type, Object Name, Symbol Type Number, and Symbol Type. The structure of the XML file generated is depicted in the XSD file in Appendix 12.
Export relationships to XML	Export the relationship data from the repository into XML files using in-built scripting functionality.	All the connected objects (objects which are "parents" of a model) were exported, including the following key information related to each connected object: Model Name, Model GUID, Model Type Number, Model Type, Parent Type Number, Parent Type, Parent Name, and Parent GUID. The structure of the XML file generated is depicted in the XSD file in Appendix 13.
Import XML files to SQL server	Import the XML data files into SQL server.	The XML files generated were imported into Microsoft SQL Server using SQL Server Integration Services (SSIS).
Compare SQL data to the ARIS repository.	Validate the integrity of the import by comparing model and object counts	The data imported into SQL Server was validated by comparing the following: • Models: Total number of models (per model type) in ARIS to SQL Server • Objects: Total number of objects in ARIS to SQL Server • Occurrences: The occurrences of a sample of objects were manually counted in ARIS and compared to SQL Server data • Parent Objects: The parent objects of a sample of models in ARIS were identified and compared to SQL Server data The results of these validations are shown in Table 13.2.

Step Name	Activity	Comment
Data Preparation	Count the number of times each model is used in a different model	The number of times a model is reused is central to this research project. The approach used to determine this value for each process model is explained in Section 5.2.
Generate reuse metrics for models.	Categorise model populations based on reuse comparisons required (e.g., business area, architecture level, old/new modelling method)	The tables containing the lists of models and objects were modified to include additional fields which were used to categorise the model for analysis (e.g., architecture level, business area, modelling method used). These attributes were then populated based on the analysis required.
	Calculate reuse metrics for each model population	The frequency of model reuse was calculated
Compare appropriate model populations.	Compare model populations to determine whether there is a significant difference in reuse between populations.	The frequency of reuse is not expected to be normal (e.g., it would be expected that most models are reused only once, and the reuse decreases consistently.

SQL Server Models' Validation

Models are stored in the repository as objects of type Model. Furthermore, Models have additional attributes such as Type (e.g. Organisational Chart, BPMN Collaboration Diagram, Class) and Group (Models are stored in Groups which are the equivalent of Folders). Model data was exported to a set of XML files, which were then imported to a Microsoft SQL Server database, as depicted in Figure 13.2. The integrity of the data transfer was verified by comparing the number of models of each model type transferred to SQL Server to the number of models of each type in the repository. Table 13.2 indicates that all models were transferred correctly.

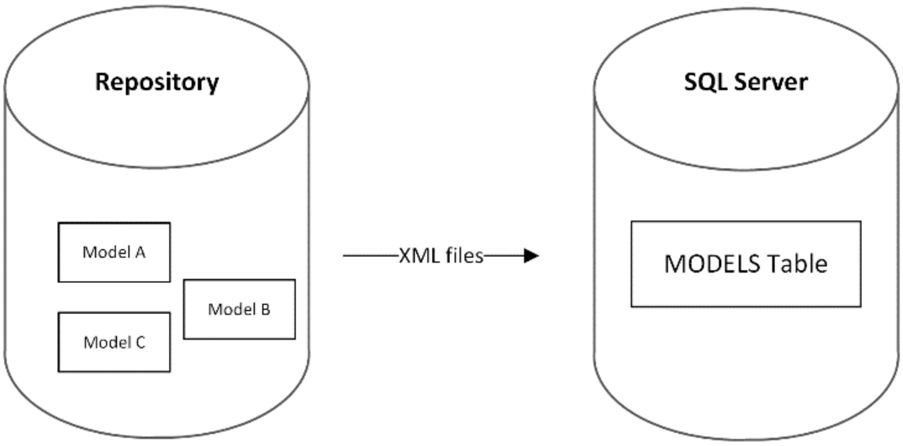


Figure 13.2: Data processing of Models in the repository

Table 13.2: Count of Model Types: Repository vs SQL Server

Model Type	ARIS Count	SQL Count
Access diagram	34	34
Application collaboration diagram	1	1
Application system type diagram	11	11
BPMN collaboration diagram (BPMN 2.0)	1691	1691
Business controls diagram	1734	1734
Business rule architecture diagram	2	2
Class diagram	1	1
Customer journey landscape	5	5
Customer journey map	19	19
Customer segmentation map	2	2
Customer touchpoint allocation diagram	4	4
Enterprise BPMN collaboration diagram	450	450
EPC	20214	20214
EPC (column display)	1877	1877
EPC (row display)	13	13
Function allocation diagram	35295	35295
Function tree	108	108
IE Data model	3	3

Model Type	ARIS Count	SQL Count
Knowledge structure diagram	9	9
KPI allocation diagram	19	19
Matrix model	11	11
Objective diagram	6	6
Organizational chart	90	90
Process selection diagram	255	255
Process selection matrix	12	12
Product tree	1	1
Product/Service tree	356	356
Quick model	125	125
Risk diagram	50	50
Screen design	1	1
Screen navigation	20	20
Service allocation diagram	139	139
Service architecture diagram	5	5
Service collaboration diagram	17	17
Spreadsheet	5	5
Task allocation diagram	1	1
Technical terms model	11	11
UML 1.4 Use case diagram	1	1
Value-added chain diagram	248	248
Total	62846	62846

SQL Server Objects' Validation

Objects are stored in the repository as entities of type Object. Furthermore, Objects have additional attributes such as Type (e.g. Function, Role, Event, Screen) and Group (Objects are stored in Groups which are the equivalent of Folders). Object data was exported to a set of XML files, which were then imported to a Microsoft SQL Server database, as depicted in Figure 13.3. The integrity of the data transfer was verified by comparing the number of objects of each object type transferred to

SQL Server to the number of objects of each type in the repository. Table 13.3 indicates that all objects were transferred correctly.

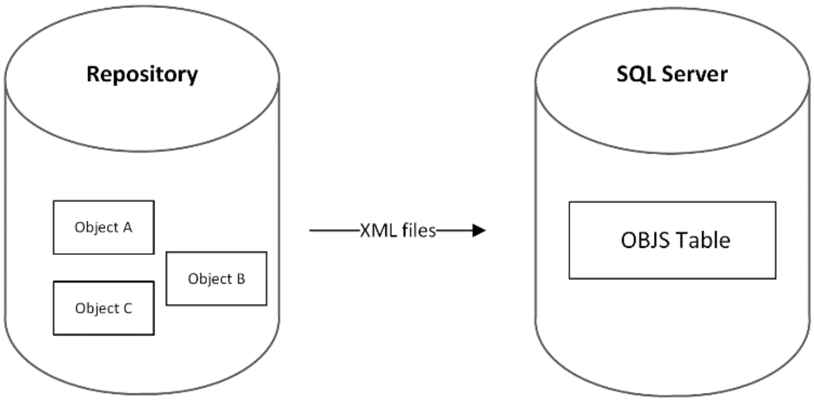


Figure 13.3: Data processing of Objects in the repository

Table 13.3: Count of Object Types: Repository vs SQL Server

Object Type	ARIS Object Count	SQL Server Object Count
Function	185598	185598
Event	176644	176644
Rule	80761	80761
Information carrier	18174	18174
Business rule	6729	6729
Role	6325	6325
Cluster/Data model	5449	5449
Screen	5087	5087
Technical term	4433	4433
Risk	2874	2874
Organizational unit	2756	2756
Quick object	2542	2542
Lane	2169	2169
Text annotation	1338	1338
KPI instance	1250	1250

Object Type	ARIS Object Count	SQL Server Object Count
Structural element	1045	1045
Class	1028	1028
Application system type	1000	1000
Product/Service	948	948
Requirement	872	872
Participant	824	824
Documented knowledge	743	743
Group	518	518
Service type	495	495
Capability	305	305
Emotion	277	277
Screen design	242	242
Message	163	163
Person	143	143
Objective	103	103
Risk category	79	79
Customer journey step	66	66
Customer touchpoint	62	62
Distribution channel	49	49
Success factor	47	47
Location	43	43
Task	37	37
Component	24	24
Customer journey	22	22
Customer lifecycle stage	21	21
ERM attribute	12	12
List	12	12
Main process	12	12

Object Type	ARIS Object Count	SQL Server Object Count
Position	11	11
Persona	6	6
IT function type	5	5
Application system class	3	3
Questionnaire template	3	3
Application system	2	2
Customer segment	1	1
Policy	1	1
System organizational unit	1	1
Total	511354	511354

SQL Server Occurrences' Validation

Models contain occurrences of objects. Using BPMN notation, for example, a BPMN process model contains activities, which in ARIS are represented by an object of type **Function** in the repository. Figure 13.4 illustrates how the objects named A, B, and C occur in the models. In Figure 13.4, it can be seen that object A occurs in two models. Each occurrence of each object in each model in the repository was exported to a set of XML files, which were then imported to a Microsoft SQL Server database, as depicted in Figure 13.4. The integrity of the data transfer was verified by comparing the number of occurrences of each object type transferred to SQL Server to the number of occurrences of each object type in the repository. Table 13.4 indicates that all occurrences were transferred correctly.

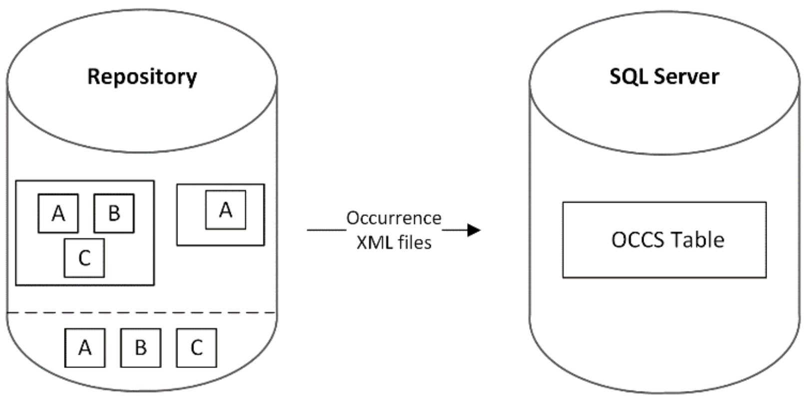


Figure 13.4: Data processing of Object Occurrences in the repository

Table 13.4: Count of object occurrences per object type: Repository vs SQL Server

Object type	ARIS	SQL Server
Function	376642	376642
Event	326595	326595
Role	171072	171072
Rule	118679	118679
Information carrier	74052	74052
Application system type	70158	70158
Organizational unit	24357	24357
Screen	20264	20264
Business rule	13882	13882
Risk	8546	8546
Cluster/Data model	6123	6123
Technical term	6064	6064
Quick object	5253	5253
Lane	4185	4185
Documented knowledge	2858	2858
Person	2855	2855
Group	2850	2850
Product/Service	2731	2731
Location	2728	2728
Class	2634	2634
KPI instance	2407	2407
Structural element	2214	2214
Requirement	2176	2176
Text annotation	1793	1793
Participant	1719	1719
Service type	1536	1536

Object type	ARIS	SQL Server
Objective	1143	1143
Capability	687	687
Screen design	652	652
Emotion	441	441
Risk category	340	340
Message	287	287
Distribution channel	250	250
Customer journey step	150	150
Customer touchpoint	102	102
Task	98	98
Component	82	82
Success factor	59	59
Customer lifecycle stage	35	35
Customer journey	33	33
Position	17	17
List	16	16
Application system	14	14
ERM attribute	12	12
Main process	12	12
IT function type	9	9
Application system class	8	8
Persona	6	6
Questionnaire template	3	3
Policy	1	1
Customer segment	1	1
System organizational unit	1	1
Total	1258832	1258832

SQL Server Parents' Validation

An activity (object type Function) can decompose into a model which depicts further details of that activity. This research project refers to object A as the **Parent** of Model A. Object A is also known as the **Connected Object** of Model A, as illustrated in Figure 13.5. The details of each model and its connected objects (a model is able to have multiple connected objects) were exported to a set of XML files, which were then imported to SQL Server as depicted in Figure 13.5. The integrity of the data transfer was verified by comparing the number of connected objects of each model type transferred to SQL Server to the number of connected objects of each object type in the repository. Table 13.5 indicates that all connected objects were transferred correctly.

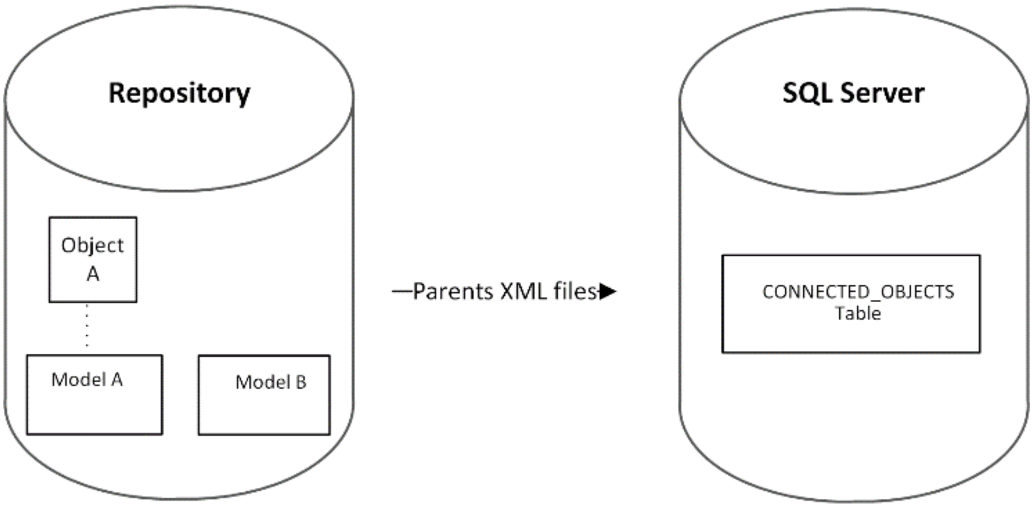


Figure 13.5: Data processing of Parents / Connected_Objects in the repository.

Table 13.5: Count of Model parents per model type: Repository vs SQL Server

Model Type	ARIS Parent Count	SQL Server Parent Count
Application system type diagram	2	2
BPMN collaboration diagram (BPMN 2.0)	852	852
Business controls diagram	1653	1653
Customer journey map	17	17
Customer touchpoint allocation diagram	1	1
Enterprise BPMN collaboration diagram	380	380
EPC	16256	16256
EPC (column display)	1395	1395
EPC (row display)	11	11

Model Type	ARIS Parent Count	SQL Server Parent Count
Function allocation diagram	33655	33655
Function tree	57	57
Knowledge structure diagram	1	1
KPI allocation diagram	20	20
Objective diagram	1	1
Organizational chart	50	50
Process selection diagram	78	78
Process selection matrix	1	1
Product tree	1	1
Product/Service tree	316	316
Quick model	41	41
Risk diagram	20	20
Service allocation diagram	138	138
Task allocation diagram	1	1
Technical terms model	9	9
Value-added chain diagram	162	162
Total	55118	55118

14 Appendix 4: Model reference frequency in the repository

The frequency distributions of the models referenced for three populations of models in the repository were determined. The three identified populations were described in Section 3.5.2. The number of models referenced for each frequency and the resulting distribution calculations are shown in Table 14.1 to Table 14.3.

Table 14.1: Team X (new Prod) model referencing

Frequency	Frequency Count	Object Type	Perc	Cum
0	48	22	3.4%	3.4%
1	594	22	42.5%	45.9%
2	234	22	16.7%	62.7%
3	142	22	10.2%	72.8%
4	62	22	4.4%	77.3%
5	43	22	3.1%	80.3%
6	42	22	3.0%	83.3%
7	50	22	3.6%	86.9%
8	27	22	1.9%	88.8%
9	21	22	1.5%	90.3%
10	10	22	0.7%	91.1%
11	9	22	0.6%	91.7%
12	7	22	0.5%	92.2%
13	9	22	0.6%	92.8%
14	13	22	0.9%	93.8%
15	10	22	0.7%	94.5%
16	10	22	0.7%	95.2%
17	3	22	0.2%	95.4%
18	2	22	0.1%	95.6%
19	1	22	0.1%	95.6%
20	5	22	0.4%	96.0%
21	3	22	0.2%	96.2%
22	3	22	0.2%	96.4%
23	4	22	0.3%	96.7%
24	12	22	0.9%	97.6%
25	3	22	0.2%	97.8%
26	3	22	0.2%	98.0%
27	1	22	0.1%	98.1%
28	2	22	0.1%	98.2%
29	3	22	0.2%	98.4%
30	2	22	0.1%	98.6%
31	2	22	0.1%	98.7%
35	2	22	0.1%	98.9%

36	1	22	0.1%	98.9%
37	2	22	0.1%	99.1%
40	1	22	0.1%	99.1%
41	1	22	0.1%	99.2%
42	1	22	0.1%	99.3%
50	1	22	0.1%	99.4%
53	1	22	0.1%	99.4%
58	1	22	0.1%	99.5%
63	1	22	0.1%	99.6%
67	1	22	0.1%	99.6%
72	1	22	0.1%	99.7%
77	1	22	0.1%	99.8%
106	1	22	0.1%	99.9%
175	1	22	0.1%	99.9%
323	1	22	0.1%	100.0%

Table 14.2: Team X (old) model referencing

Frequency	Frequency Count	Object Type	Perc	Cum
0	363	22	0.393709	39.4%
1	458	22	0.496746	89.0%
2	80	22	0.086768	97.7%
3	15	22	0.016269	99.3%
5	2	22	0.002169	99.6%
6	4	22	0.004338	100.0%
7	0	22	0	100.0%
8	0	22	0	100.0%
9	0	22	0	100.0%
10	0	22	0	100.0%
11	0	22	0	100.0%
12	0	22	0	100.0%
13	0	22	0	100.0%
14	0	22	0	100.0%
15	0	22	0	100.0%

Table 14.3: Team Y (new Prod) model referencing

Frequency	Frequency Count	TYPENUM_OBJ	Perc	Cum
0	2263	22	0.264586	26.5%
1	5310	22	0.620835	88.5%
2	684	22	0.079972	96.5%
3	117	22	0.013679	97.9%
4	59	22	0.006898	98.6%
5	15	22	0.001754	98.8%
6	21	22	0.002455	99.0%
7	15	22	0.001754	99.2%
8	7	22	0.000818	99.3%
9	4	22	0.000468	99.3%
10	2	22	0.000234	99.3%
11	8	22	0.000935	99.4%
12	4	22	0.000468	99.5%
13	2	22	0.000234	99.5%
14	4	22	0.000468	99.6%
15	3	22	0.000351	99.6%
16	3	22	0.000351	99.6%
17	4	22	0.000468	99.7%
20	2	22	0.000234	99.7%
21	2	22	0.000234	99.7%
22	1	22	0.000117	99.7%
23	2	22	0.000234	99.8%
24	1	22	0.000117	99.8%
25	1	22	0.000117	99.8%
26	2	22	0.000234	99.8%
27	1	22	0.000117	99.8%
28	2	22	0.000234	99.8%

Frequency	Frequency Count	TYPENUM_OBJ	Perc	Cum
33	2	22	0.000234	99.9%
36	1	22	0.000117	99.9%
37	1	22	0.000117	99.9%
41	1	22	0.000117	99.9%
47	2	22	0.000234	99.9%
63	1	22	0.000117	99.9%
79	1	22	0.000117	99.9%
82	1	22	0.000117	100.0%
106	1	22	0.000117	100.0%
127	1	22	0.000117	100.0%
138	1	22	0.000117	100.0%
142	1	22	0.000117	100.0%

15 Appendix 5: Permission request to the research organisation

Address1

4 April 2020

Address2

To Whom It May Concern

Permission to use Bank1 and ARIS process repository for Doctoral Research Project

I have been studying towards a Masters post-graduate degree in Information Systems at UCT and was previously granted permission to use Bank1 as my research site as well as the use of the ARIS repository. The topic I have selected relates to the reuse of business process models in a multi-channel / multi-product environment, with the focus being on formalizing the approach that has been adopted/tested in Dept 1 over the past four years. The research topic turned out to be much wider than anticipated, and as a result, UCT has permitted me to upgrade my studies from a Masters degree to a Doctorate.

However, as a result, I need to once again obtain permission from Bank1 to conduct the research at Bank1.

From a formal research perspective, I need to try to obtain a measure of process model reuse historically and then compare it to the levels of reuse resulting from the Dept1 approach. In order to measure historical process model reuse, I would like to analyse historical ARIS databases in an attempt to obtain such a measure. Dept2 have indicated that backups of the ARIS database going back several years exist but that I require formal approval for them to provide me with those backups. Also, UCT has a policy whereby any research conducted must have formal approval from the UCT ethics committee. In order to obtain such approval, formal approval must also be obtained from any organisations involved as it is not permitted to conduct research “under the radar”!

I would also like to mention that as part of the course, we are required to write and submit a paper to a journal or conference as one of the assignments. This paper was a literature survey on this topic and was accepted by the conference (Mediterranean Conference on Information Systems) and was presented in Genoa, Italy, in September 2017. I will also be presenting a paper at the BPM 2019 Conference in Vienna in September 2019. This paper describes how to measure the level of process model reuse in a repository such as ARIS. Such a measurement had never been previously described and tested and was one of the reasons for the upgrade from a Masters to a Doctorate.

There is a clear shortage of research conducted in this field internationally.

Therefore, I would like to request formal permission for the following:

- To conduct my research on this topic at Bank1 in order to formalize the approach academically.
- Access to historical ARIS databases that may be available in order to measure historical and current process model reuse

Additional Information:

I have read and understand the Bank 1 Code of Ethics and Conduct Policy and confirm that I will comply therewith.

In order to meet the ethics requirements of UCT, the following will also be observed:

- Bank1 will not be mentioned by name in the dissertation. Reference would only be made to a "major South African bank". From a dissertation perspective, Bank1 would be anonymous.
- Information confidential to Bank1 (e.g. detailed business processes) will not be shared with any third party. However, it may be required that some discussions be held occasionally with my supervisor (Prof. Lisa Seymour).
- No detailed process information will be disclosed in the dissertation. Any processes discussed will be at a level that would be generic to any financial institution.
- Interviews with individuals will be completely voluntary and interview content will be anonymized in order to protect interviewees. Permission to record any interviews must be obtained from the interviewee concerned.
- All data will be stored on a computer protected by a password. Any data stored in the cloud will be encrypted using an application such as Sookasia.
- Bank1 will be provided with a copy of the final dissertation.

I trust that you will favourably consider my request.

Yours faithfully

16 Appendix 6: Interview Cover Letter

Department of Information Systems
Leslie Commerce Building
Engineering Mall, Upper Campus
OR
Private Bag. Rondebosch 7701
Tel: +27 (0) 21 650 4028 Fax: +27 (0) 21650 2280
Internet: http://www.commerce.uct.ac.za/informationsystems/

13 November 2017

Dear Sir/Madam,

I am a student enrolled in the full-time Business Science specializing in Information Systems at the University of Cape Town. In partial fulfilment of the program, I am required to submit an empirical research dissertation.

The topic I have selected for my research paper is focused on how to increase the reuse of complete process models in organisations. The objective of my research is to develop a modelling method which increases the reuse of complete process models.

The interview will last approximately 45- 60 minutes. Participation is entirely voluntary, and withdrawal from the study can be requested at any time. The research has been approved by the Commerce Faculty Ethics Research Committee. The study ensures complete anonymity as you are not required to provide any personal details. Your participation would be much appreciated.

The findings of this research study will be compiled into a report and presented to the University of Cape Town for academic purposes. Participants will remain anonymous. Participants will have the opportunity to receive a copy of the final report. Should you have any questions regarding the research, please feel free to contact the researcher.

Thank you for your time and participation.

Sincerely,

Ross Veitch
Masters Student
Department of Information Systems
University of Cape Town
Email: vtcros002@myuct.ac.za
Tel: 0833258489

Assoc. Prof. Lisa Seymour
Research Supervisor
Department of Information Systems
University of Cape Town
Email: lisa.seymour@uct.ac.za

"Our Mission is to be an outstanding teaching and research university, educating for life and addressing the challenges facing our society."

17 Appendix 7: Informed Consent Form

Name of researcher:

Title of research project:

By filling out this questionnaire / answering the questions put to me:

- I agree to participate in this research project.
- I have read this consent form and the information it contains and had the opportunity to ask questions about it.
- I understand that I was selected to participate in this study due to my expertise / position.
- I agree to my responses being used for research on condition my privacy is respected. I understand that my responses will be used in aggregate form only, so that I will not be personally identifiable.
- I understand that I am under no obligation to take part in this project.
- I understand I have the right to withdraw from this project at any stage.
- I understand that this research might be published in a research journal or book. In the case of dissertation research, the document will be available to readers in a university library in printed form, and possibly in electronic form as well.

Name of Participant : ______________________

Signature of Participant : ______________________

Date : ______________________

The researcher must supply you with an ***Information sheet*** which provides his / her contact details, outlines the nature of the research and how the information will be used and explains what your participation in the research involves (e.g. how long it will take, participants' roles and rights (including the right to skip

questions or withdraw without penalty at any time), any anticipated risks/benefits which may arise as a result of participating, any costs or payment involved (even if none, these should be stated))				
Has this been provided?	Yes		No	
Have you received verbal confirmation/explanations where needed?	Yes		No	

18 Appendix 8: Interview Guide

This interview specifically is aimed at understanding your experiences of process modelling and process model reuse in this organisation. You will be asked some questions that relate to the following:

- Your background (general information)
- Benefits (or not) of process modelling and reuse
- How you go about creating new models and finding the required information
- Your thoughts on how process modelling can be improved in this organisation and reuse increased.
- Your experiences modelling using the proposed method.
- Any general comments that you may have relating to process model reuse

---

Date of interview:

Stakeholder Name / Research Identifier:

Questions:

General

1. Name, age, role?
2. Could you tell us about the sorts of jobs you have held previously?
3. What is your educational background?
4. How many years process modelling experience do you have?
5. How many years have you worked in this organisation?
6. How long have you been in your current position?
7. What training have you received in process modelling?
8. How would you describe your current job?
9. Can you describe your current project?

Process modelling Benefits

10. Describe process modelling benefits to this organisation
11. Describe process modelling benefits in your area
12. Describe process modelling benefits in projects
13. How could process modelling add more benefits?

Process modelling challenges

14. What problems have you experienced in process modelling:
    a. In your area
    b. On your projects
    c. Within this organisation

Finding information

15. How does your team share information?
16. How do you collaborate with your fellow team members?
17. To what extent do you collaborate with your fellow team members?
18. How do your projects share information?

Process Model Reuse

19. Do you reuse complete process models in your work?
20. Please describe how.
21. Please describe what limits your reuse of complete process models.

Solution Objectives

22. Why do you think that complete process models are not reused much?
23. What would you suggest to increase complete process model reuse?

Evaluation

24. Please describe your experiences modelling using the new method?
25. Describe what you liked about the new method
26. Describe the problems / issues that you experienced
27. Has the new method increased complete process model reuse in your opinion?
28. What changes / improvements would you suggest?
29. How would you improve the training material?
30. Do you think that the new method reduces the amount of modelling required in a project?
31. Can you describe any additional benefits you may have noticed?

Interview conclusion

32. Do you have any other comments or observations that you think may be useful to the research project?

19 Appendix 9: Research Participants

Table 19.1: Participant Details

Participant ID	Process Modelling Experience	Years worked at Bank 01
P01	20	20
P02	10	10
P03	>20	14
P04	11	1
P05	16	11
P06	8	15
P07	12	17
P08	20	20
P09	10	18
P10	10	8
P11	15	21

20 Appendix 10: Model XML File Structure

The xml schema definition for the model export files is as follows:

```
<?xml version=”1.0”?>
<xs:schema attributeFormDefault=”unqualified” elementFormDefault=”qualified”
xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
 <xs:element name=”MODELS”>
  <xs:complexType>
   <xs:sequence>
    <xs:element minOccurs=”0” maxOccurs=”unbounded” name=”MODEL”>
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs=”0” name=”NAME_DB” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_MODEL”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
          </xs:element>
       <xs:element minOccurs=”0” name=”NAME_GROUP”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
          </xs:element>
       <xs:element minOccurs=”0” name=”GUID_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_MODEL” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”DATE_CREATED” type=”xs:string” />
       <xs:element minOccurs=”0” name=”DATE_LASTCHANGED” type=”xs:string” />
```

```
            <xs:element minOccurs="0" name="LASTUSER" type="xs:string" />
            <xs:element minOccurs="0" name="PROCARCHLEVEL" type="xs:string" />
            <xs:element minOccurs="0" name="PROCESSLEVEL" type="xs:string" />
            <xs:element minOccurs="0" name="PROCESSOWNER">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                <xs:maxLength value="2048"/>
                </xs:restriction>
              </xs:simpleType>
                </xs:element>
            <xs:element minOccurs="0" name="MODELLER" type="xs:string" />
            <xs:element minOccurs="0" name="CUSTODIAN">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                <xs:maxLength value="2048"/>
                </xs:restriction>
              </xs:simpleType>
                </xs:element>
            <xs:element minOccurs="0" name="CREATOR" type="xs:string" />
            <xs:element minOccurs="0" name="RCM_APPROVER" type="xs:string" />
            <xs:element minOccurs="0" name="RCM_IN_PRODUCTION" type="xs:string" />
            <xs:element minOccurs="0" name="RCM_LAST_MODIFIED_DATE" type="xs:string" />
            <xs:element minOccurs="0" name="RCM_MODEL_STATUS" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

21 Appendix 11: Object XML File Structure

The xml schema definition for the object export files is as follows:

```
<?xml version=”1.0”?>
<xs:schema attributeFormDefault=”unqualified” elementFormDefault=”qualified”
xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
 <xs:element name=”OBJECTS”>
  <xs:complexType>
   <xs:sequence>
    <xs:element minOccurs=”0” maxOccurs=”unbounded” name=”OBJECT”>
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs=”0” name=”NAME_DB” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_OBJ”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
          </xs:element>
       <xs:element minOccurs=”0” name=”NAME_GROUP”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
          </xs:element>
       <xs:element minOccurs=”0” name=”GUID_OBJ” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_OBJ” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_OBJ” type=”xs:string” />
       <xs:element minOccurs=”0” name=”DATE_CREATED” type=”xs:string” />
       <xs:element minOccurs=”0” name=”DATE_LASTCHANGED” type=”xs:string” />
```

```
<xs:element minOccurs=”0” name=”LASTUSER” type=”xs:string” />
<xs:element minOccurs=”0” name=”PROCARCHLEVEL” type=”xs:string” />
<xs:element minOccurs=”0” name=”PROCESSLEVEL” type=”xs:string” />
<xs:element minOccurs=”0” name=”CREATOR” type=”xs:string” />
<xs:element minOccurs=”0” name=”DESCRIPTION”>
<xs:simpleType>
<xs:restriction base=”xs:string”>
<xs:maxLength value=”4000”/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element minOccurs=”0” name=”INVOLVES”>
<xs:simpleType>
<xs:restriction base=”xs:string”>
<xs:maxLength value=”4000”/>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

22 Appendix 12: Occurrence XML File Structure

The xml schema definition for the occurrence export files is as follows:

```
<?xml version=”1.0”?>
<xs:schema attributeFormDefault=”unqualified” elementFormDefault=”qualified”
xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
 <xs:element name=”OCCS”>
  <xs:complexType>
   <xs:sequence>
    <xs:element minOccurs=”0” maxOccurs=”unbounded” name=”OCC”>
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs=”0” name=”NAME_DB” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”GUID_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_MODEL” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”GUID_OBJ” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_OBJ” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_OBJ” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_OBJ” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_SYMB” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_SYMB” type=”xs:string” />
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

23 Appendix 13: Connected Object (Parent) XML File Structure

The xml schema definition for the parent export files is as follows:

```
<?xml version=”1.0”?>
<xs:schema attributeFormDefault=”unqualified” elementFormDefault=”qualified”
xmlns:xs=”http://www.w3.org/2001/XMLSchema”>
 <xs:element name=”PARENTS”>
  <xs:complexType>
   <xs:sequence>
    <xs:element minOccurs=”0” maxOccurs=”unbounded” name=”PARENT”>
     <xs:complexType>
      <xs:sequence>
       <xs:element minOccurs=”0” name=”NAME_DB” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_MODEL”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
          </xs:element>
       <xs:element minOccurs=”0” name=”GUID_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_MODEL” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_MODEL” type=”xs:string” />
       <xs:element minOccurs=”0” name=”GUID_PARENT” type=”xs:string” />
       <xs:element minOccurs=”0” name=”TYPENUM_PARENT” type=”xs:integer” />
       <xs:element minOccurs=”0” name=”TYPE_PARENT” type=”xs:string” />
       <xs:element minOccurs=”0” name=”NAME_PARENT”>
       <xs:simpleType>
          <xs:restriction base=”xs:string”>
          <xs:maxLength value=”2048”/>
          </xs:restriction>
        </xs:simpleType>
```

```
          </xs:element>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
</xs:schema>
```

24 Appendix 14: Model Update - Model and Object Management

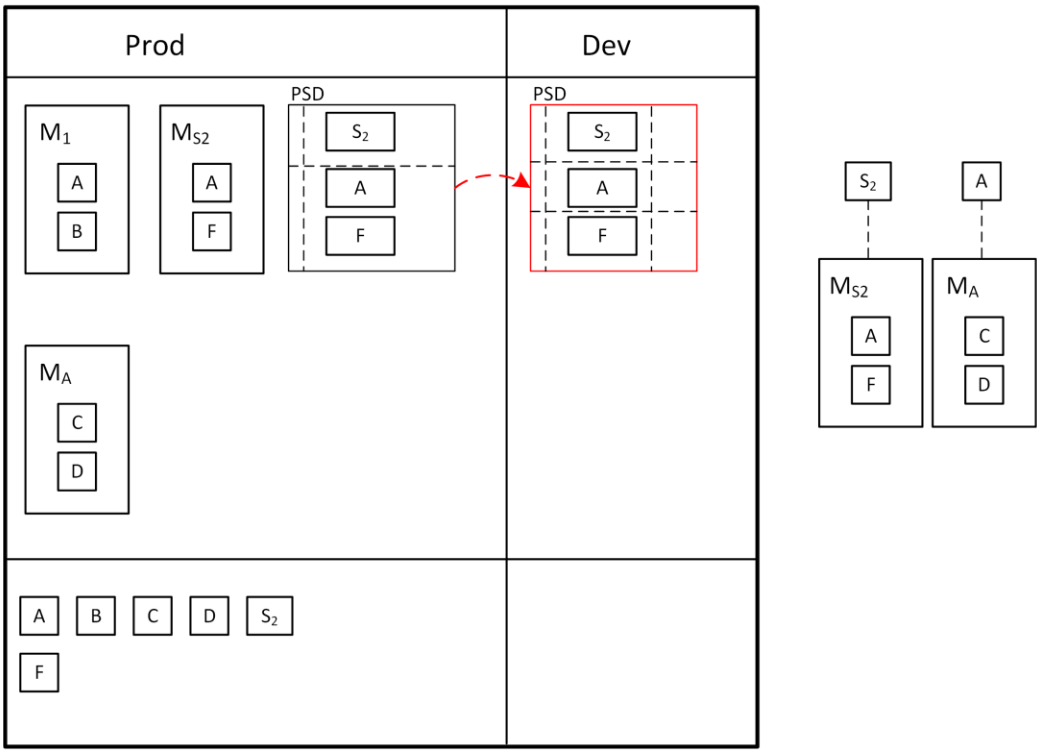
This Appendix describes the detailed steps required to update an existing Production model.

For the purposes of this description the following assumptions have been made:

- There is an existing PSD in Production
- There is an existing Scenario S_2 defined in that PSD.
- The Process M_A needs to be updated.

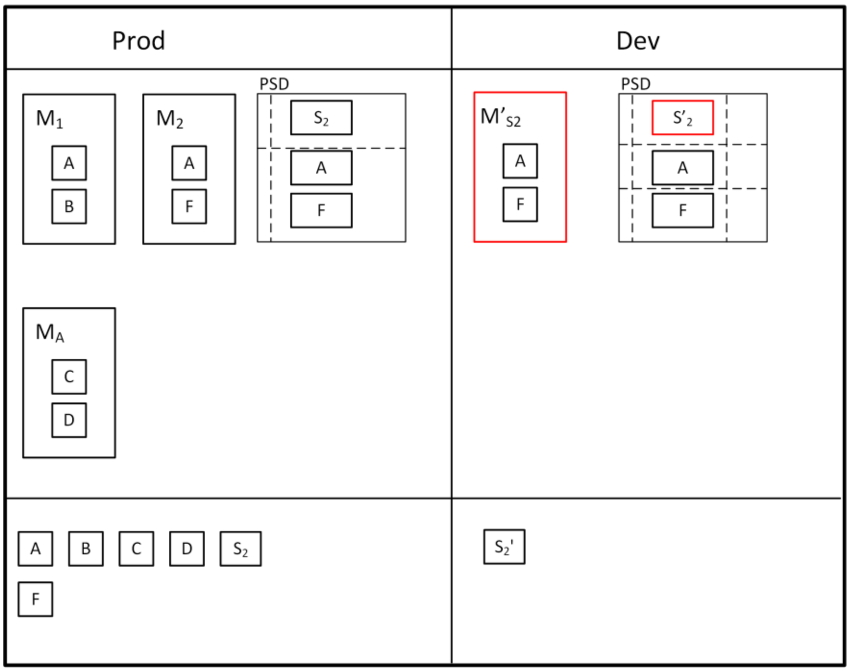
Step 1: Copy the PSD from Production to Development

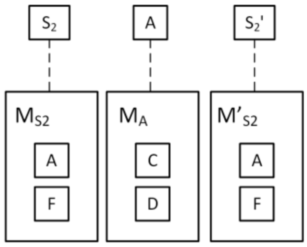
Objective: Update M_A which is used in Scenario S_2



1. S_2 is the parent of M_2
2. Copy the appropriate PSD to Development

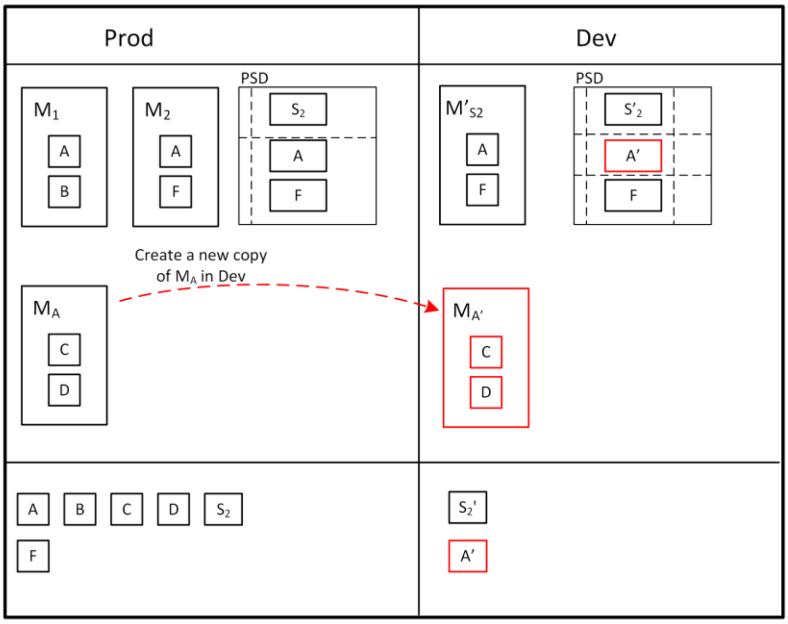
Step 2: Create the Development version of S2 and MS2

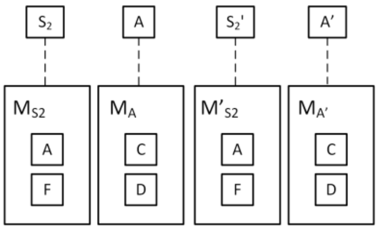




1. Duplicate S_2 (S_2') in the Development PSD
2. Copy M_2 to Development (M_2') and make
S_2' the parent object

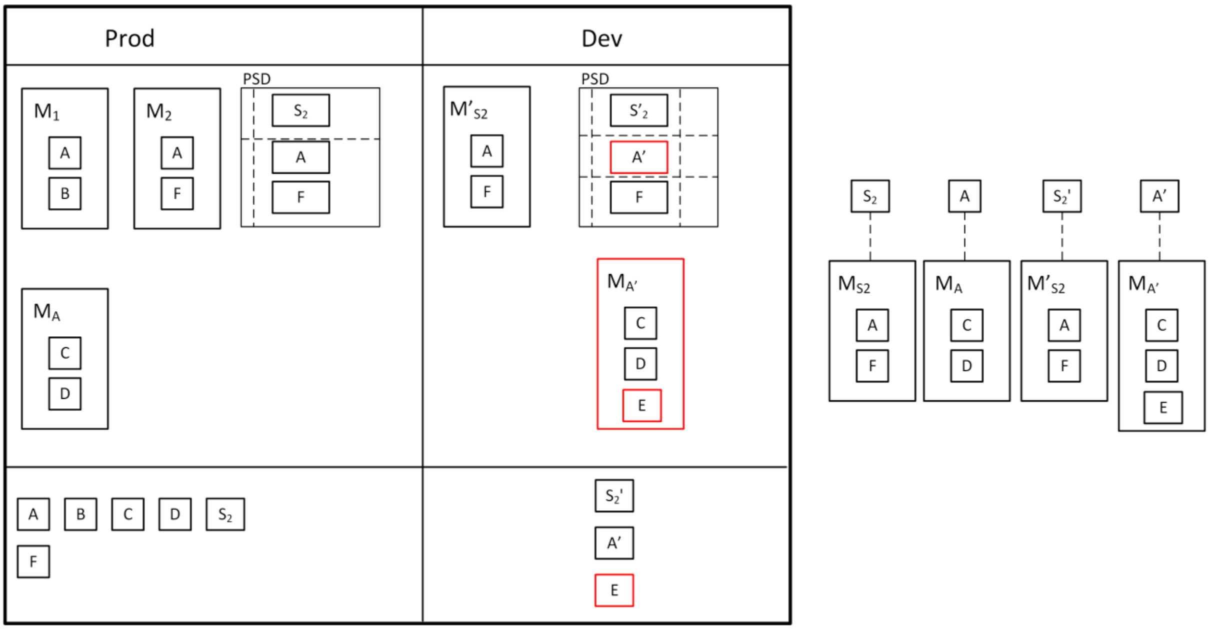
Step 3: Create the Development version of A and MA





1. Duplicate A (A') in the Development PSD
2. Copy M_A to Development ($M_{A'}$)
3. Make A' the parent object of $M_{A'}$

Step 4: Update MA’

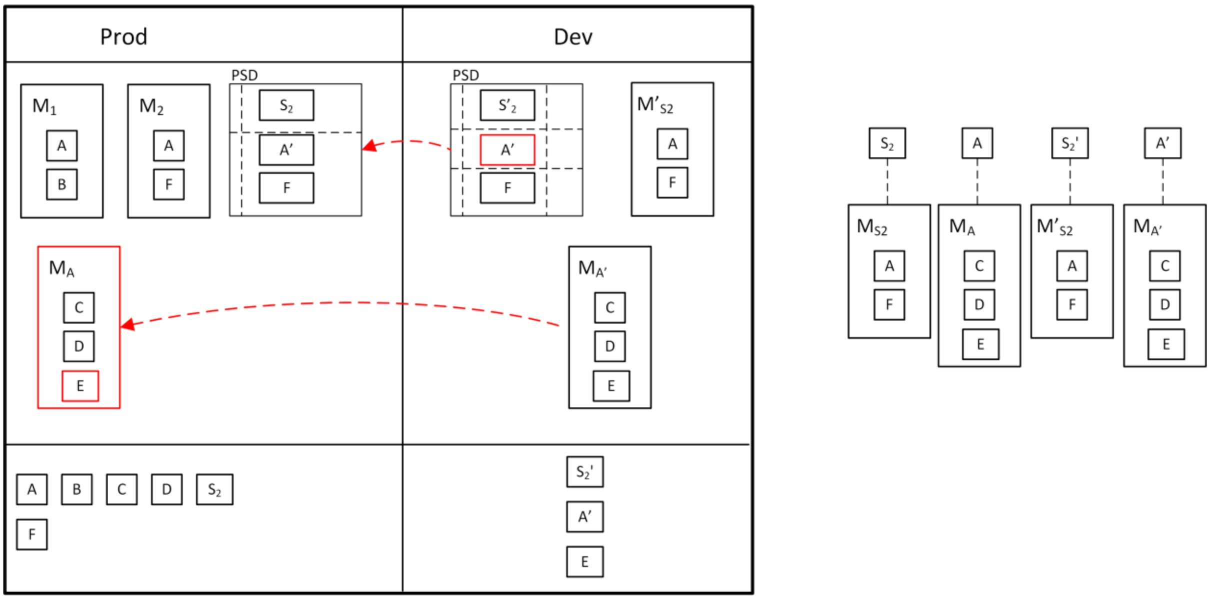


Update $M_{A'}$

1. Edit $M_{A'}$ by adding new object E.

E is located in Dev and can be edited.

Step 5: Update the Production models with the Development versions (A’->A and S’2->S2)

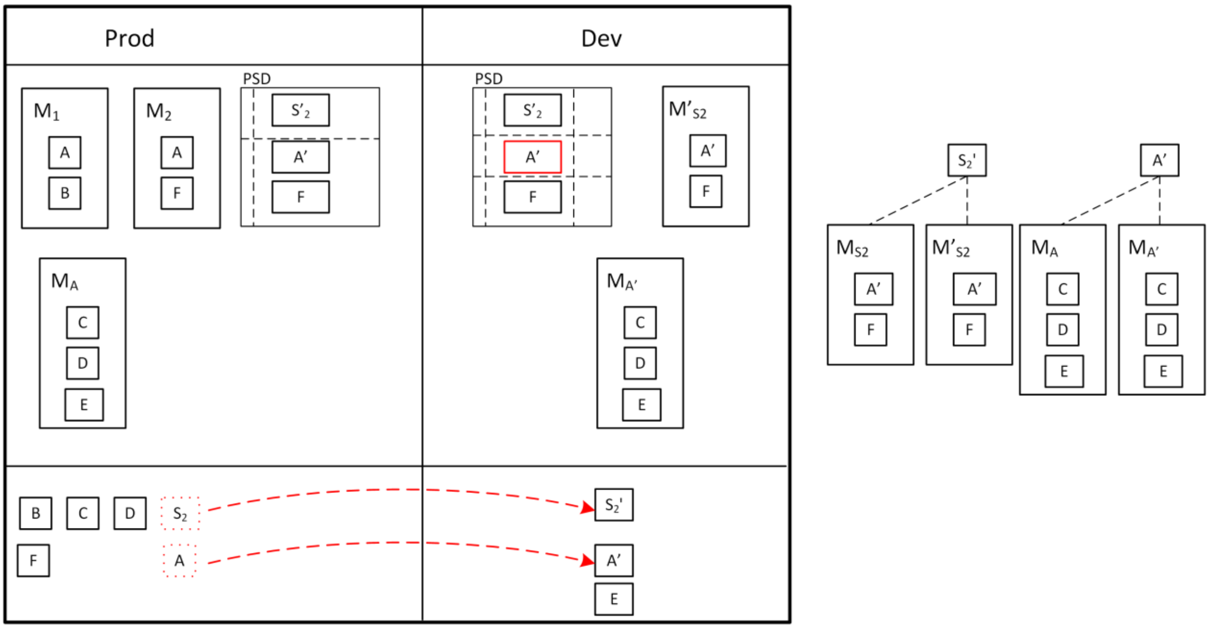


1. Copy PSD content from Development into Production model
2. Copy $M_{A'}$ content into M_A.

Note 1: Objects S’2, A’, and E are still in the Development folders

Note 2: Objects S’2, and A’ are still in the parent objects of M_{S2} and $M_{A'}$

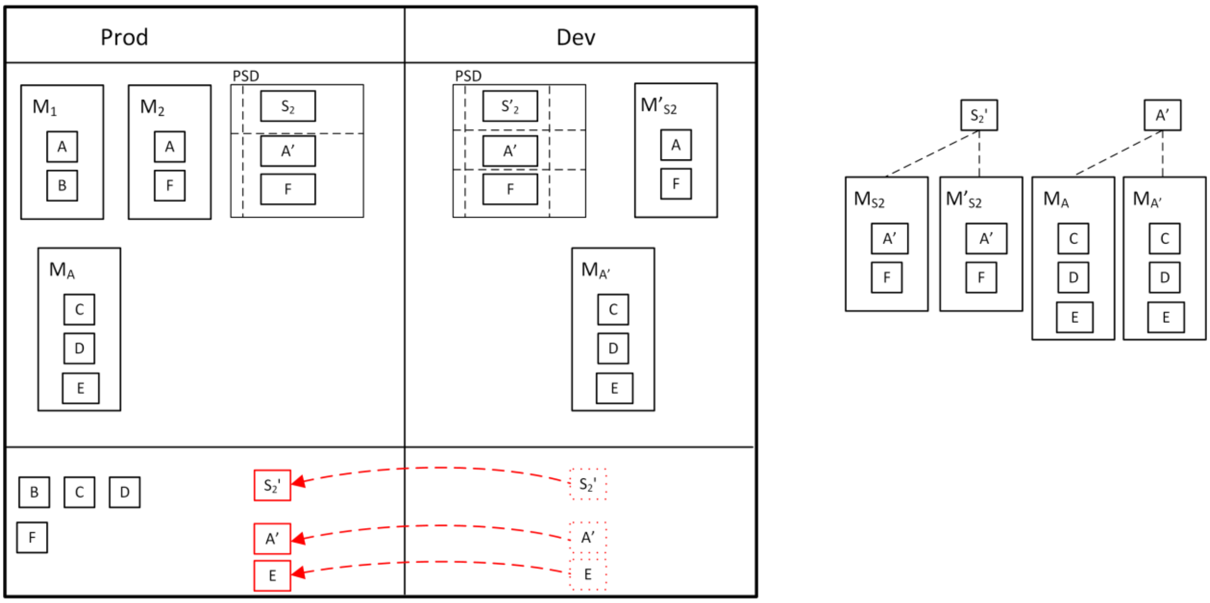
Step 6: Consolidate (merge) the parent objects of each model updated



1. Consolidate S_2 and S'_2, keeping S'_2 as the master
2. Consolidate A and A', keeping A' as the master'

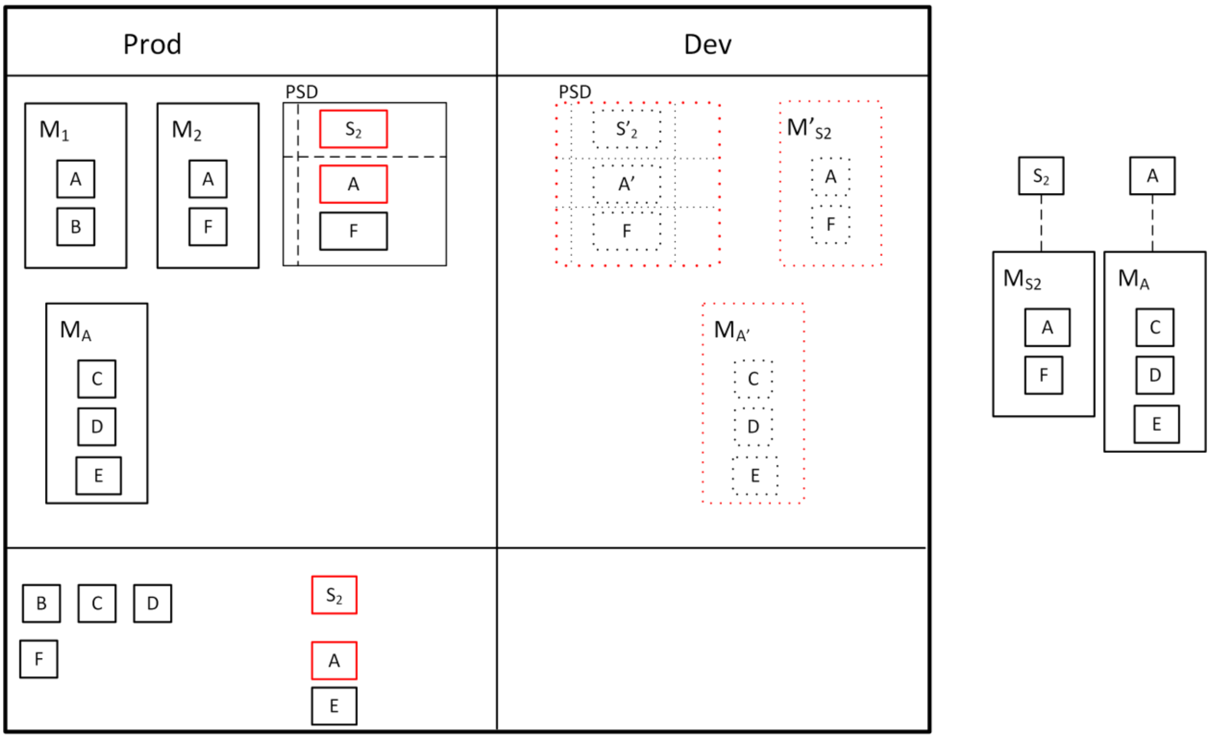
Note 1: S'_2 is still located in the Development folders
Note 2: S_2 no longer exists in the repository
Note 3: S'_2 is now the parent object of both M_{S2} and M'_{S2}
Note 4: A' is still located in the Development folders
Note 5: A no longer exists in the repository.
Note 6: A' is now the parent object of both M_A and $M_{A'}$

Step 7: Move the objects from Development to Production



1. Move S2', A', and E from Development to Production
2. Move S2', A', and E from Development to Production
3. Move S2', A', and E from Development to Production

Step 8: Cleanup



1. Break the assignment from S'_2 to M'_{S2}
2. Delete M'_{S2}
3. Break the assignment from A’ to $M_{A'}$
4. Delete $M_{A'}$
5. Delete the Development PSD
6. Rename S2' to S2
7. Rename A’ to A

25 Appendix 15: New Model - Model and Object Management

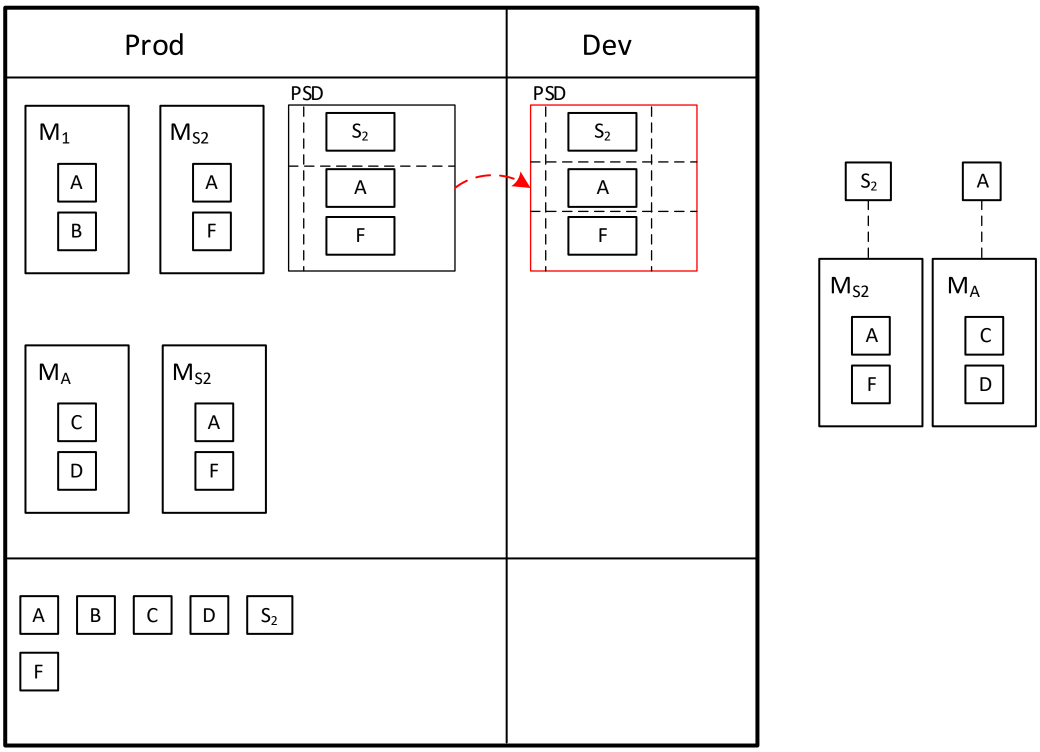
This Appendix describes the detailed steps required for a new Production model.

For the purposes of this description, the following assumptions have been made:

- There is an existing PSD in Production
- There is an existing Scenario S_2 defined in that PSD.
- The Process M_A needs to be updated.

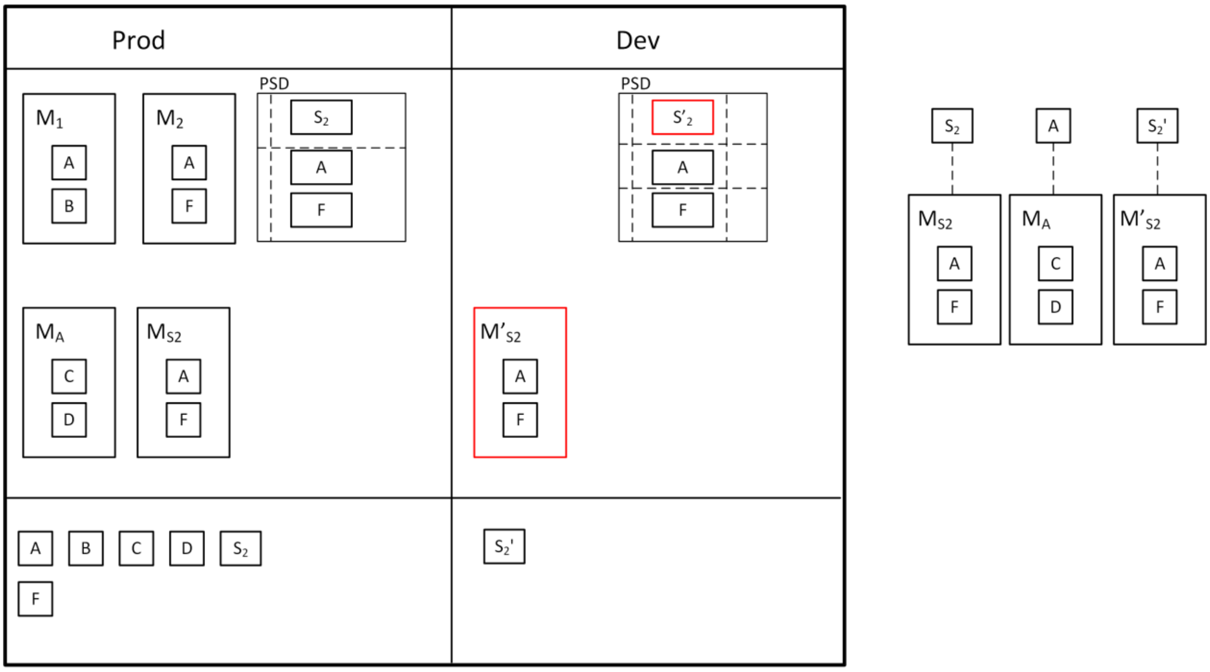
Step 1: Copy the PSD from Production to Development

Objective: Create a new model M_G which is used in Scenario S_2



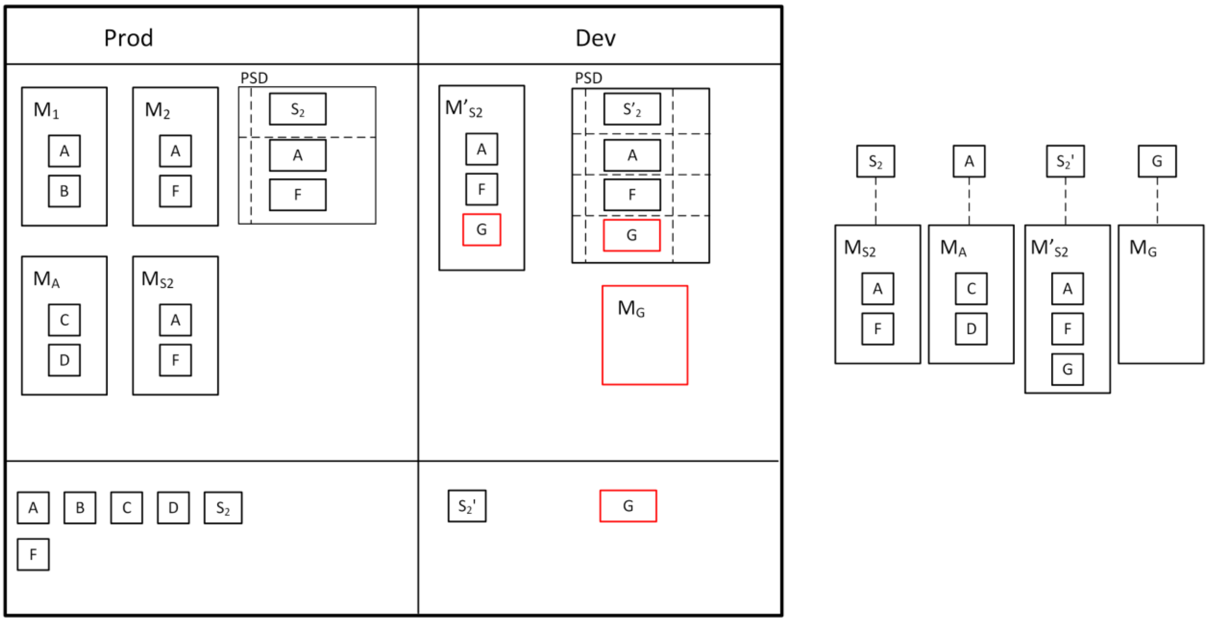
1. S_2 is the parent of M_2; A is the parent of M_A
2. Copy the appropriate PSD to Development

Step 2: Create the Development version of S2 and MS2



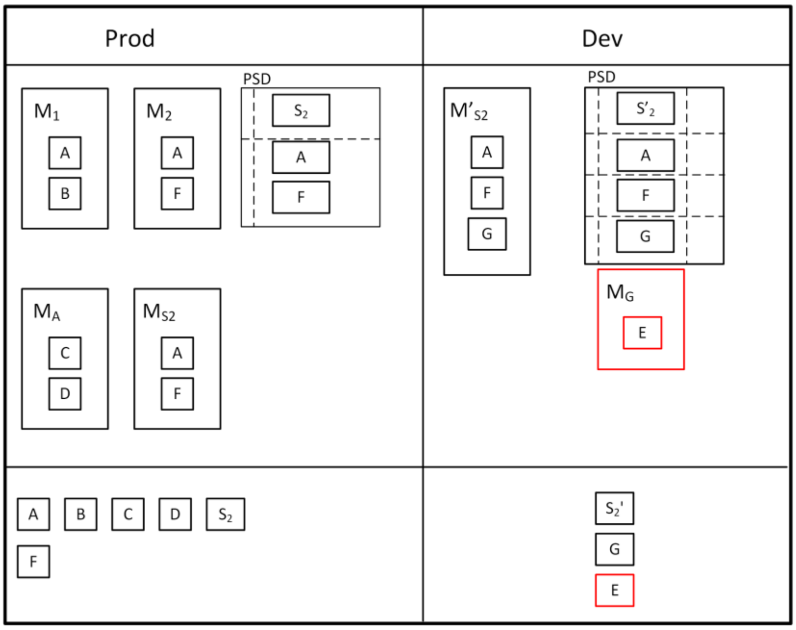
1. Duplicate S_2 (S_2') in the Development PSD
2. Copy M_{S2} to Development (M'_{S2}) and make S_2' the parent object

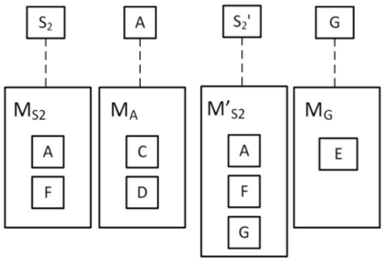
Step 3: Create new parent object G and model M_G



1. Create new Function G in the PSD
2. Create a new empty model M_G and make G the parent function

Step 4: Populate model M_G



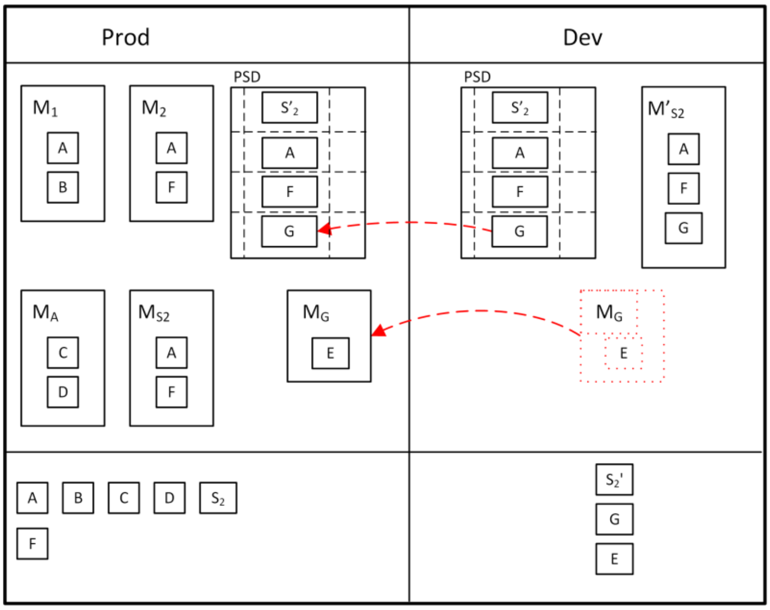


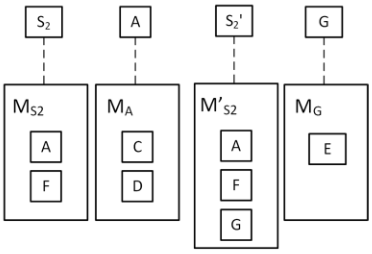
Populate M_G

1. Edit M_G by adding new object E.

E is located in Dev and can be edited.

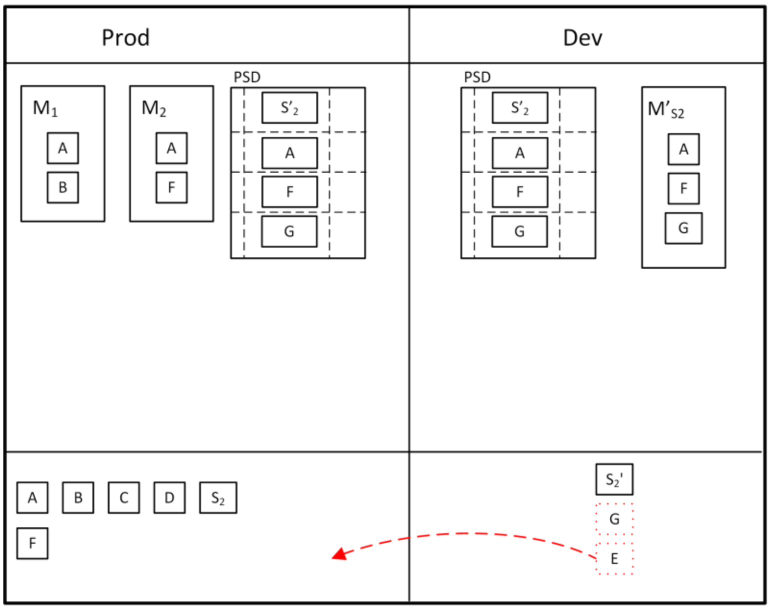
Step 5: Move model M_G from Development to Production

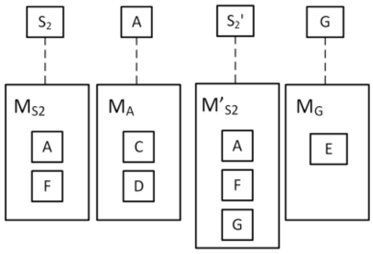




1. Move model M_G from Development to Production
2. For this example, we ignore the updating of the PSD and M_{S2} because these are not a new models and the model and object management follows the process applicable to updating an existing model.

Step 6: Move objects E and G from Development to Production





1.Move model Objects E and G from Development to Production