

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (<https://dspace.lboro.ac.uk/>) under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

**A Novel Workflow Management System for Handling
Dynamic Process Adaptation and Compliance**

By

Mohamad Saiful Haji Omar

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the
award of Doctor of Philosophy of Loughborough University

February, 2014

© by Saiful Omar 2014

Abstract

Modern enterprise organisations rely on dynamic processes. Generally these processes cannot be modelled once and executed repeatedly without change. Enterprise processes may evolve unpredictably according to situations that cannot always be prescribed. However, no mechanism exists to ensure an updated process does not violate any compliance requirements.

Typical workflow processes may follow a process definition and execute several thousand instances using a workflow engine without any changes. This is suitable for routine business processes. However, when business processes need flexibility, adaptive features are needed. Updating processes may violate compliance requirements so automatic verification of compliance checking is necessary. The research work presented in this Thesis investigates the problem of current workflow technology in defining, managing and ensuring the specification and execution of business processes that are dynamic in nature, combined with policy standards throughout the process lifecycle.

The findings from the literature review and the system requirements are used to design the proposed system architecture. Since a two-tier reference process model is not sufficient as a basis for the reference model for an adaptive and compliance workflow management system, a three-tier process model is proposed. The major components of the architecture consist of process models, business rules and plugin modules. This architecture exhibits the concept of user adaptation with structural checks and dynamic adaptation with data-driven checks.

A research prototype - Adaptive and Compliance Workflow Management System (ACWfMS) - was developed based on the proposed system architecture to implement core services of the system for testing and evaluation purposes. The ACWfMS enables the development of a workflow management tool to create or update the process models. It automatically validates compliance requirements and, in the case of violations, visual feedback is presented to the user. In addition, the architecture facilitates process migration to manage specific instances with modified definitions. A case study based on the postgraduate research process domain is discussed.

Dedication

To the loving memory of my late Father:

Haji Omar Abdullah (1920 – 2010),

who has raised me to believe that I can achieve
anything if I set my mind to it.

And to my Mother, Wife and four children for their
Love, Inspiration and Sacrifices that has brought me this
far.

Acknowledgement

I would like to convey my utmost gratitude to numerous people who have provided me with exceptional support, encouragement and wisdom throughout my PhD journey.

First, and foremost, I would like to thank my supervisors Professor Paul Chung and Dr Christian Dawson for their guidance, support and encouragement throughout my PhD programme, without them, my journey would have been much harder. I will always remain grateful for their time, patience and continuous help in reviewing my Thesis and research.

Most of all, I am thankful to my wife: Aziah Abdullah, who endured all the hardship, shared the setbacks and as well, the progresses. And for my four children: Iffah, Syafiq, Farah and Abd Mateen, for understanding the sacrifice of family time within this journey. I would also like to thank my brothers, Hj Yakub, Hj Ismail and Hj Yahya for their tremendous overall support. And also my other siblings and friends for their inspiration and encouragement. And to the administrative and technical support staff at Loughborough University.

Finally, I would like to express my appreciation to Government of Brunei through the Institute Technology Brunei for sponsoring my PhD study at the Department of Computer Science, Loughborough University.

Table of Contents

Chapter 1	Introduction	1
1.1	Introduction	1
1.2	Background	1
1.3	Project Overview.....	3
1.4	Aim and Objectives	4
1.5	Contributions	4
1.6	Thesis Organisation.....	5
Chapter 2	Workflow Technology	6
2.1	Introduction	6
2.2	Workflow System Benefits and Weakness	6
2.3	Workflow System.....	8
2.4	Generic Workflow Models.....	9
2.4.1	Product Implementation Model	10
2.4.2	Workflow Reference Model.....	11
2.5	Workflow Definition Interchange	13
2.5.1	Process Definition	14
2.5.2	Business Process Modelling.....	14
2.6	Integrating Business Process and Business Rules.....	17
2.7	Existing Workflow Development tools	17
2.8	Conclusion.....	20
Chapter 3	Adaptive and Compliance Workflow	21
3.1	Introduction	21
3.2	Dynamic Process Change	22
3.2.1	Introduction	22
3.2.2	Techniques for Supporting Flexible and Adaptive Workflow	22

3.2.3 Process Adaptation	24
3.2.4 Process Adaptation with Business Rules	26
3.3 Compliance supports	26
3.3.1 Introduction	26
3.3.2 Build-Time Compliance Checking.....	27
3.3.3 Run-time Compliance Checking	27
3.3.4 Compliance Auditing	28
3.3.5 Integrating Build-Time and Run-Time.....	28
3.4 Requirements for integrated Process Adaptation and Compliance.....	28
3.5 Adaptive and Compliance Integrated Workflow Development Tools.....	30
3.6 Conclusion.....	32
Chapter 4 Research Methodology	33
4.1 Introduction	33
4.2 Research Methodology and Approach	33
4.2.1 Systems Architecture Design	33
4.2.2 Prototype Development	34
4.2.3 Case Study Research	37
4.3 Case Study Domain Selection	38
4.4 Scenario-Based.....	39
4.5 Conclusion.....	40
Chapter 5 System Design and Implementation	41
5.1 Introduction	41
5.2 Conceptual Design	41
5.2.1 Introduction	41
5.2.2 Process Model.....	42
5.2.3 Business Rules.....	45

5.2.4	Plugin Modules	46
5.2.5	User Adaptation with Structural Check	47
5.2.6	Dynamic Adaptation with Data-Driven Check	54
5.3	Prototype Implementation	56
5.3.1	Introduction	56
5.3.2	Modelling Standard and Process	58
5.3.3	Firing Business Rules.....	60
5.3.4	Client and Server Plugin Modules.....	62
5.3.5	Implementation of User Adaptation with Structural Check.....	64
5.3.6	Implementation of Dynamic Adaptation with Data-Driven Check.....	66
5.4	Conclusion.....	66
Chapter 6	Case Study	68
6.1	Introduction	68
6.2	Evaluation Criteria	68
6.3	Case Study Domain: Postgraduate Research Process	71
6.4	Scenario-Based Testing.....	78
6.4.1	Scenario 1: Apply for Conference Fund with User Adaptation with Completeness Check.....	79
6.4.2	Scenario 2: Redo Progress Review with Dynamic Adaptation	83
6.4.3	Scenario 3. Progress to Extension Year with Adaptive Standard Model.....	86
6.4.4	Scenario 4: Meeting and Training Frequency with Data Check.....	89
6.4.5	Scenario 5: Application for Leave of Absence with User Adaptation.....	93
6.5	Conclusion.....	95
Chapter 7	Conclusions and Future Work.....	97
7.1	Introduction	97
7.2	Thesis Review.....	97

7.3	Contributions	100
7.4	Applicability of the System	103
7.5	Limitations of the Research and System	104
7.6	Recommendation for Future Work	105
7.7	Concluding Remarks	106
	References	107
	Appendix 1 - Basic BPMN Modelling Elements and Description	115
	Appendix 2 – Publication and Recognition Resulted From This Thesis	118
	Appendix 3 – Glossary of Terms	119

List of Figures

Figure 2-1 WfMS Architecture and Characteristics (adapted from WfMC 1999)	9
Figure 2-2 Generic Workflow Product Structure (adapted from WfMC 1999)	10
Figure 2-3 Workflow Reference Model - Components and Interfaces (adapted from WfMC, 1999)	12
Figure 2-4 Business Process related standards time-line (adapted from Chinosi & Trombetta 2012)	16
Figure 3-1 Process Adaptation and Built-in Flexibility (adapted from Weber et al. 2008)	25
Figure 4-1 Prototype Model (adapted from Naumann & Jenkins 1982)	35
Figure 5-1 ACWfMS Architecture	43
Figure 5-2 Three-Tier Non-Static Reference Model	45
Figure 5-3 User Adaptation example showing ASM and UDP	48
Figure 5-4 FlowElement Class Diagram (adapted from OMG 2011)	49
Figure 5-5 Flow Elements Completeness Check algorithm	50
Figure 5-6 InputOutputSpecification Class Diagram (adapted from OMG 2011)	51
Figure 5-7 Data Association Correctness Check Algorithm	52
Figure 5-8 Correctness Check Algorithm	53
Figure 5-9 Dynamic Adaptation example showing UDP	54
Figure 5-10 Business Rules to Execute	55
Figure 5-11 Example of Business Rules Definition	55
Figure 5-12 ACWfMS Technologies Foundation	57
Figure 5-13 Designer User Interface (adapted from JBPM 2012)	59
Figure 5-14 Business Process with Ruleflow	62
Figure 5-15 Business Rule Statements	62
Figure 5-16 Screen shot of ACWfMS - Visual Compliance Check with Instance Tracker	65
Figure 6-1 PGR Standard Model: Core Process	72
Figure 6-2 PGR Standard Model: First and Second Year Sub-Process	73
Figure 6-3 PGR Standard Model: Third and Extension Year Sub-Process	75
Figure 6-4 PGR Standards Model: Conduct Examination Sub-Process	77

Figure 6-5 SM PGR: Request for Leave of Absence Sub-Process.....	78
Figure 6-6 SM PGR: Attend Conference Sub-Process	78
Figure 6-7 Process Flexibility with BPMN Ad-Hoc Sub-Process	81
Figure 6-8 User Adaptation with Compliance Checks	82
Figure 6-9 BPMN flexibility support with selection technique	84
Figure 6-10 Dynamic adaptation with business rules.....	85
Figure 6-11 Inconsistent Two-Tier Model after Specific Instance Dynamic Adaptation	87
Figure 6-12 Consistent Three-Tier Model after Specific Instance Dynamic Adaptation	88
Figure 6-13 Validate Meeting and Training Frequency using BPMN process logic	91
Figure 6-14 Validate Meeting and Training Frequency using ACWfMS hybrid techniques (Business Process and Business Rules)	92
Figure 6-15 Supervisory Meeting Email Reminder	92
Figure 6-16 Attend Training Email Reminder	93
Figure 6-17 User Adaptation without ACWfMS Instance Tracker	94
Figure 6-18 User Adaptation with ACWfMS Instance Tracker Module	95
Figure 6-19 Specific Instance Migration with ACWfMS Instance Migration Module ..	95

Chapter 1

Introduction

1.1 Introduction

The need to react to changes in a quick and flexible way is one of the key challenges facing today's enterprises (Poppendieck & Poppendieck 2006). Most enterprises rely on information systems to support organisational processes. Workflow management systems are one of the technologies that have delivered a great deal of productivity improvements, although they have been designed mainly to support static and repetitive process (Weber et al. 2009). Another design limitation of current workflow technology is the lack of the ability to ensure that the specification and execution of a process are compliant with standard (Chung et al. 2008).

The purpose of this Thesis is to improve the compliance and adaptive workflow that supports the creation and execution of dynamic processes that enable individual process instances to conform against a required standard. In particular, the formal foundation, conceptual and system design, a prototypical implementation of adaptive and compliance workflow management system (ACWfMS) is addressed, and applying postgraduate research process as a domain for testing and evaluating the prototype.

This chapter gives an overview of the Thesis and is structured as follows: Section 1.2 provides the background and Sections 1.3 provides the motivation for this research. Section 1.4 introduces the domain case study and Section 1.5 defines the aim and objectives of the research. Section 1.6 summarises the contributions of this work and Section 1.7 presents the structure of the Thesis.

1.2 Background

A business process is a collection of activities executed following a predefined order to achieve specific business objective or policy goals (Chinosi & Trombetta 2012). A

workflow is a concept of automation of a business process, in whole or part, during which documents, information or tasks are passed between participants according to a defined set of rules (WfMC 1995). A Workflow Management System supports the specification (build-time functions), execution (run-time control functions), and dynamic control of workflows involving humans and information systems (run-time interactions) (Hollingsworth 1995).

Executing non-compliant processes in any organisation are an expensive practice that costs time, effort, reputation and competitive advantage. This can stem from lack of tools to enable the derivation of policy into organisational processes and also treating organisational policy separately from organisational processes (Governatori & Sadiq 2009).

Compliance is about ensuring an organisation performs in accordance with requirements. These requirements derive from laws, regulations, agreed standards, contracts, and organisational governance. Lu et al. (2008) generalise compliance into three distinct perspectives: corrective, detective and preventive that form a collective approach to compliance management.

Automating compliance can be achieved either 'by detection' or 'by design' (Kharbili et al. 2008; Lu et al. 2008; Sackmann & Kahmer 2008). Compliance by detection is based on reporting and monitoring of process executions and data usage (after-the-fact detection), thus making it flexible to adapt to new requirements or processes. This approach will not be able to prevent non-compliant behaviour.

Compliance by design is a preventive approach. It imposes desired behaviour and prevents undesirable events. This approach prevents the actual execution of non-compliant behaviour. It is considered a fool-proof approach if all requirements have been declared within the system.

Workflow is a good contender for exploring the integration of compliance by design. As compliance requirements may change and vary from one domain to another, they are no longer suitable for hard-coding (the practice of embedding business processes or data directly into source code program). Further, making changes may lead to non-compliance.

1.3 Project Overview

In this Thesis, dynamic terms refer to a progressive and continuous change of business processes during build-time or instance processes during run-time. The routine processes such as payroll, processes do not change very often. On the other hand, dynamic processes do not always go according to plan and some processes require incremental progress that makes individual process instances unique. However, existing workflow technology does not adequately address the dynamic changes of individual process instances to conform to certain standard requirements.

Existing approaches to automating compliance checking in workflow technology involve defining a standard (reference) model to assess the degree of compliance of a user- defined process (Chung et al. 2008). This approach works well with the static standard model. However, in a flexible workflow environment, where business process needs to adapt to dynamic situation, this does not work. Hence, another layer of model is introduced, an adaptive standard model that serves as a dynamic reference to cope with evolving workflow instance.

The postgraduate research process is well suited as a case study for this Thesis. The research process is very dynamic in nature and impossible to define a standard workflow to cover the diversities of research processes and activities for different departments, supervisors and students. Furthermore, it is not supported by any automation mechanism to ensure that the specification and execution of the process model are compliant with the policy regulations or codes of practice adopted by any universities.

The common issues that PhD research students encounter are isolation, time management and supervision (Hockey 1994). These issues generally lead to factors such as: lack of progress, lack of confidence, demotivation and withdrawal. De Rezende et al. (2006) proposed a workflow system for Thesis development. However, they identified it is not only difficult to define a standard workflow for different supervisors, but it is also very difficult to follow a single workflow in any Thesis due to the need to answer different questions raised during its development.

Hence, an adaptive and compliance workflow is necessary to support the dynamic nature of PhD process.

1.4 Aim and Objectives

The aim of this research is *“to investigate current workflow technology in order to develop a novel solution of an integrated adaptive and compliance workflow management system architecture that handles dynamic and ad-hoc process modification, as well as automating compliance validation features throughout the process lifecycle”*. Specifically the objectives are to:

- Study the requirements that support dynamic and compliance business processes within workflow technology;
- Identify factors that influence process automation for checking compliance;
- Identify factors that provide adaptive workflow to support an evolutionary and dynamic modification process model both at build-time and run-time while conforming to compliance requirements;
- Develop a system architecture to combine the techniques of adaptive and compliance workflow solutions and adopting the Business Process Modelling Notation (BPMN) standards;
- Implement a prototype system (ACWfMS) that implements the key components of the architecture for testing and evaluation purposes; and
- Evaluate ACWfMS by implementing a postgraduate research regulation compliance process as a case study.

1.5 Contributions

The main contributions of the Thesis are:

- Conceptual design of a novel adaptive and compliance workflow management system architecture for handling process adaptation and compliance features throughout the process lifecycle;
- Propose three-tier reference models: Standard Model, Adaptive Standard Model and User-Defined Process;

- Enhance process validation for non-compliant process through automatic tracking and managing conformance and process execution for specific instances;
- Propose an instance tracker tool that assists process adaptation via process editor; and
- Propose an instance migration tool that allows an updated process instance to continue executing based on the updated process logic (may include control data and application data) and to cope with dynamic changes during run-time.

1.6 Thesis Organisation

The remainder of this Thesis is organised as follows. Chapter 2 reviews the concept of workflow technology and workflow management systems with particular relevance to support the integration of adaptive and compliance workflow system.

Chapter 3 looks at existing techniques for supporting flexible and adaptive workflows and approaches to check process models for compliance requirements. General requirements for the proposed architecture are introduced.

Chapter 4 discusses the overall research methodology adopted by this Thesis. It reviews the importance of system architecture and activities for conducting case study in software engineering. It discusses the rational of adopting Postgraduate Research Process as case study domain.

Chapter 5 proposes a comprehensive system architecture framework for adaptive and compliance workflow management system. It describes the implementation of a prototype: the Adaptive and Compliance Workflow Management System (ACWfMS).

Chapter 6 conducts an evaluation of research outcomes and limitation of proposed architecture in the form of a case study to investigate the use of ACWfMS to support the postgraduate research process domain.

Chapter 7 briefly review the Thesis, summaries the achievements, identifies the limitations, and outlines the needs for future work in some areas.

Chapter 2

Workflow Technology

2.1 Introduction

The two terminologies of Workflow Management (WfM) and Business Process Management (BPM) remain confusing and are used carelessly (Ko 2009). Gartner Research (Hill et al. 2008) describes BPM as a management discipline with workflow management supporting it as a technology. They further claimed that BPM is a process-oriented management discipline while WfM technology is found in BPM systems. Another viewpoint is that the business process life cycle includes: design, configuration, enactment and diagnosis, where WfM has little support on the diagnosis phase (Weske et al. 2004). Since the scope of this Thesis is limited to the design, configuration, and enactment of business process lifecycles, this Thesis will be using the term workflow management throughout.

This chapter provides an overview of generic workflow system architectures. Section 2.2 highlights the benefits and weaknesses of current WfM tools. Section 2.3 describes some important concepts of workflow technology. Section 2.4 reviews the generic workflow model. Section 2.5 reviews the workflow definition interchange. Section 2.6 reviews the integration of business process and business rules. Section 2.7 reviews existing workflow development tools and Section 2.8 concludes this chapter.

2.2 Workflow System Benefits and Weakness

According to Global Industry Analyst (cited by Jose 2011), the global market for business process management is projected to exceed \$5.0 billion by the year 2017, led by organizational needs to improve efficacy, efficiency and strategic value of critical business processes in various scenarios. The introduction of workflow management tools should be seen as an opportunity to improve both on the business and the software development level (Baeyens, 2004; E-Workflow, 2011):

Business level:

- Improves efficiency by eliminating unnecessary steps through business process automation;
- Better process control which improves management of the business process by standardising working methods and availability of audit trails;
- Improved customer service by providing consistency in the process that leads to greater predictability in levels of response to customers;
- Flexibility of software control over processes to enable re-design in line with changing business needs; and
- Business process improvement that focuses on business processes streamlining and simplification.

Software development level:

- Reduced development risk where business analysts use the same language as the developers. Hence, developers do not have to make translations from user requirements to a software design;
- Centralised implementation in which the business process changes, although the implementation of scattered software over various systems development will still be clear; and
- Rapid application development where WfMS free the developer from keeping track of the resources (human or machine) in a process, leading to faster development and code that is more maintainable.

The main weaknesses with current workflow system approaches are (Reijers et al. 2003; Chung et al. 2008; Strijbosch 2011):

- Processes and exceptions need to be declared in advance. Processes that cannot be forecasted are difficult to support and further exception paths are not clear;
- Users are restricted in probable actions, consequently they bypass the system.

- The work that needs to be done is set into activities, but the work itself is more finely grained;
- Routing of work by the WFM is focused on what needs to be done, instead of what can be done. This results in rigid inflexible workflows;
- Due to the complexity of the business process, it needs analysts to make and update the changes. However letting the end user do it themselves fails in most cases; and
- The lack of the ability to ensure that the specification and execution of a process are compliant with the standard.

The benefits of business process automation for improving process control and customer services have attracted enterprise organisations into adopting workflow system. However, as identified above, it does come with limitations and weaknesses especially in the area of handling complexity, uncertainty and compliance of business processes. After a process has been enacted, it is very hard to support changes.

2.3 Workflow System

Workflow has been described as the movement of documents and tasks through a business process (Hee 2004). A business process is an activity or set of activities that can accomplish a specific organizational goal. Workflow is defined by WfMC (1999) as:

“a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications”

A workflow management system (WfMS) helps to separate the business logic represented by the business process from the information system that supports the process. The separation allows business processes to be designed without major amendments to the underlying computing infrastructure (Brien & Wiegand 1998).

After almost two decades the WfMC (1999) framework is still being used as a reference by researchers and developers (Mendling et al. 2008; Mendling 2009; Liu et al. 2011). The WfMC framework provides a convenient platform for describing the capabilities of a WfMS. It supports three functional areas:

- Build-time functions that are concerned with defining, and probably modelling, the workflow process and other parts of its activities;
- Run-time control functions that are concerned with managing the workflow processes in an operational environment and sequencing other activities to be handled as part of each process; and
- Run-time interactions with human users and IT application tools for processing various activity steps.

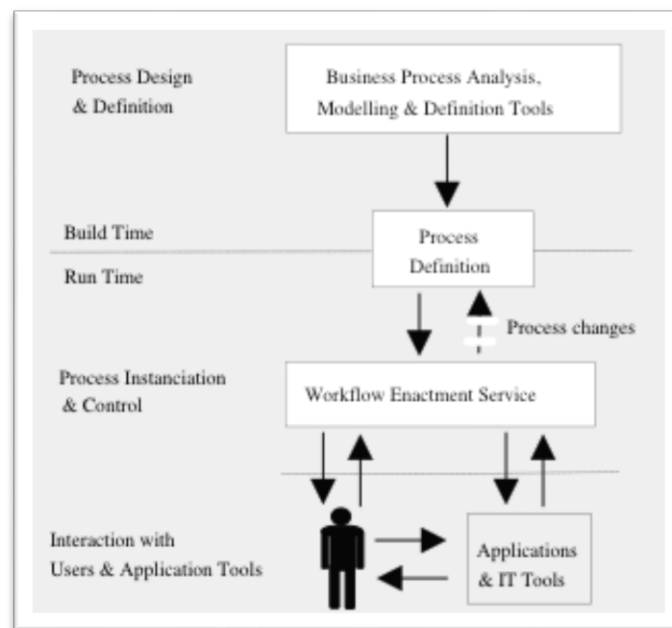


Figure 2-1 WfMS Architecture and Characteristics (adapted from WfMC 1999)

Figure 2-1 illustrates the WfMS architecture and the relationships between functional areas above. The two stages of workflow application are (WfMC 1995): **build time** is the period of time when automated and/or manual workflow process descriptions are defined and/or modified electronically; and **run time** is the period of time during the process is operational, with process instances being created and managed.

2.4 Generic Workflow Models

The WfMC initiatives are aimed towards providing a general framework of a workflow system and come in two abstract models: Product Implementation Model and Reference Model.

2.4.1 Product Implementation Model

As an abstract model, WfMC identifies the main functional components within a workflow system and the interfaces between them. The main functional components in a generic workflow system are illustrated in Figure 2-2. It has three types of components:

- Software components to support various functions within the WfMS (shown in dark fill);
- Various types of system definitions and control data used by one or more software components (shown in unfilled); and
- Applications and databases that are not part of the WfMS but may be invoked during enactment (shown in light fill).

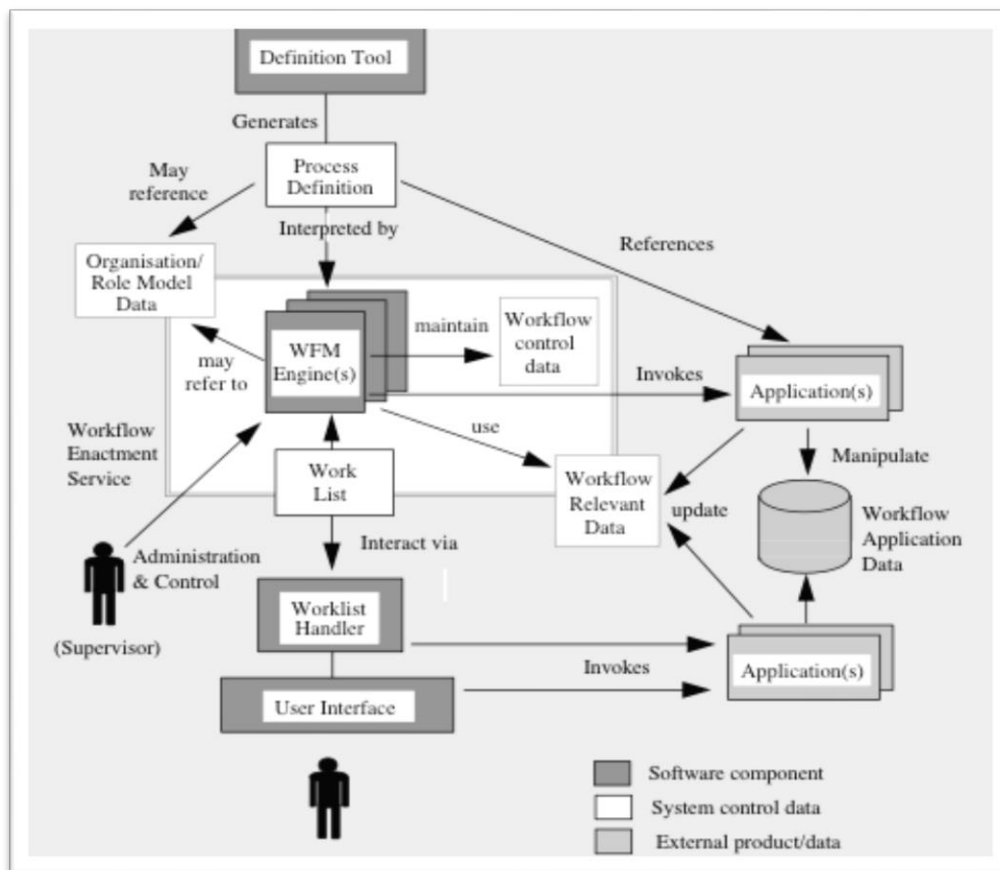


Figure 2-2 Generic Workflow Product Structure (adapted from WfMC 1999)

The descriptions of the major functional components are described below:

- Process definition tools are used to describe process definition in a computer process able form;
- Process definition contains information for the execution of the process by the workflow enactment software;
- Workflow enactment service interprets process description, navigates the sequence of activities, adds work items into a user work list, and invokes application tools as necessary;
- Workflow data can be distinguished into three types: Firstly, workflow control data represents the dynamic state of the workflow system and its process instance, which is managed and accessed by the WfMS. Secondly; workflow-relevant data are used by the WfMS to determine the state transitions of the workflow instance. And thirdly, application data are used strictly by applications supporting the process instance;
- Worklist holds work items assigned by WfMS to the user for attention by the worklist handler;
- Worklist handler manages interactions between the workflow participants and workflow enactment service. It acts as a front end to a worklist and is in charge of prompting the content of a worklist to its owner.

2.4.2 Workflow Reference Model

Based on the generic workflow application structure, WfMC (1999) developed the Workflow Reference Model through identified interfaces that enables products to interoperate at various levels. It puts emphasis on Workflow APIs (WAPI) and interchange formats, which is used to support workflow management functions across the functional areas. Figure 2-3 shows the major components and interfaces within the workflow architecture.

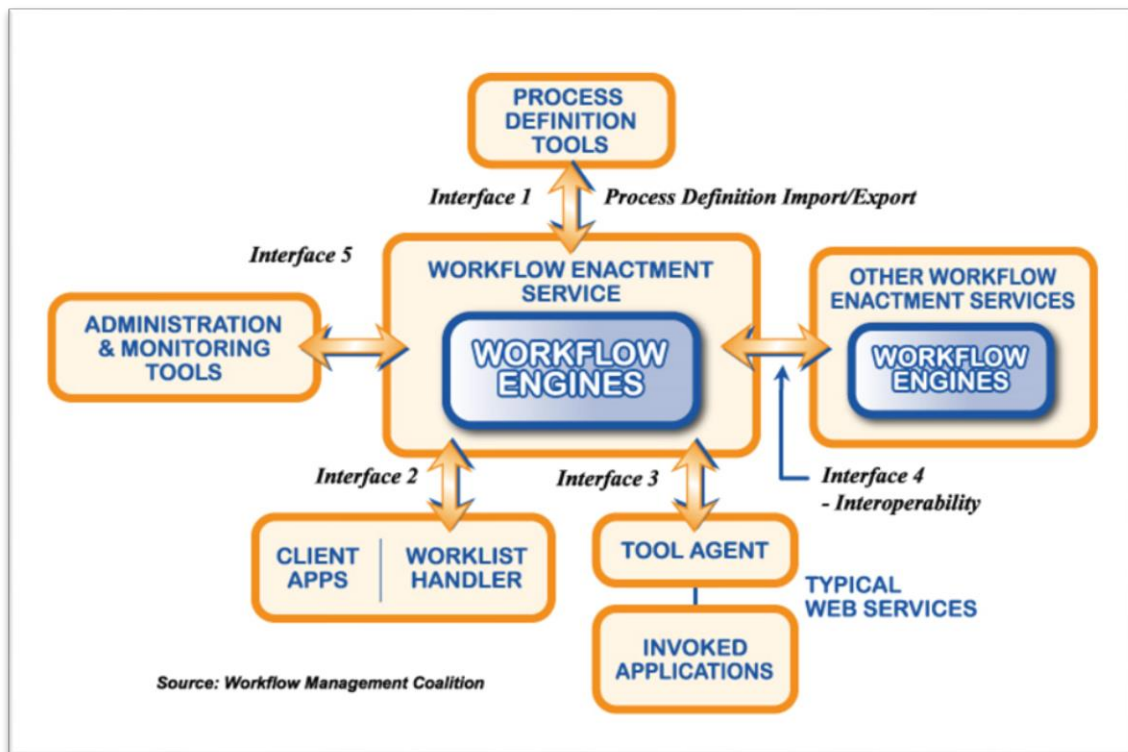


Figure 2-3 Workflow Reference Model - Components and Interfaces (adapted from WfMC, 1999)

The Workflow Reference Model has five interfaces and the description of each interface is described below.

Workflow Definition Interchange (Interface 1) is used to interchange format and API calls between the modelling and definition tools, and runtime workflow management software. It supports the exchange of process definition information over a variety of interchange media.

Workflow Client Application Interface (Interface 2) contains a variety of standard sets of APIs (the WAPI) to provide a consistent manner for access from a workflow application to workflow engine and worklist. Worklists may appear on user screens or other modes of interaction.

Invoked Applications Interface (Interface 3) allows the workflow enactment service to invoke required user applications and transfer data to and from the invoked applications. It is applicable to application agents and workflow compatible applications that could interact

directly with a workflow engine. For non-compatible workflow applications, Tool Agents are used as a bridge to start up and terminate applications, transfer workflow relevant information to and from applications, and control the application's running status. This feature provides an important enterprise integration function.

WAPI Interoperability Functions (Interface 4) aim to define common interface standards that enable workflow interoperability that can pass work items between various workflow products. Nevertheless, WfMC are not enforcing the vendors to follow strictly proposed standards. Interface 4 defines the common interpretation of the process definition, and runtime support for the interchange of various types of control information and to transfer workflow relevant and/or application data between the different enactment services.

Administration and Monitoring Interface (Interface 5) provides a common interface that enables several workflow services to share a range of common administration and system monitoring functions.

Over the years, the usefulness of this approach has become apparent. However, as technology has evolved, a range of different interface specifications have been defined with the technologies of the time (Hollingsworth 2004). The initial interface bindings reflected a relatively low-level programming view of the interfaces that were based on APIs. Subsequently, higher-level programming became accepted, such as IDL and CORBA (for the OMG), MIME email (for process interoperability), Web Services and XML.

Based on the descriptions of these interfaces, none have addressed explicitly the aims of this research – to automate compliance validation and provide adaptive features in workflow solutions. In order to provide these features, a novel system architecture is required to be developed within the scope of WfMC and relevant workflow standards.

2.5 Workflow Definition Interchange

As defined in Section 2.4.2, the WfMC outlined five interfaces of Workflow Reference Model. Interface 1 uses a process definition expression language to describe and/or execute business process in workflow management system for process enactment. A number of process definition standards were established (Ko et al. 2009) with the goals to interchange

business process definition between workflow modelling tools and workflow management systems.

2.5.1 Process Definition

Aalst et al. (2003) introduced workflow patterns that are widely used to describe workflow functionality in a language or system-independent manner. The Workflow Patterns Initiative (2011) identified five types of patterns that cover the following workflow perspectives: control-flow, data, resource, exception handling and presentation.

The control-flow perspective captures parts of control-flow dependencies between various tasks such as parallelism, choice, and synchronization. The data perspective handles the passing of information and scoping of variables. The resource perspective handles the resource to task allocation and delegation. The exception handling perspectives handles the deviations from normal execution arising during the run-time of a business process.

These patterns have been widely used by practitioners, vendors and academics alike in the selection, design and development of workflow systems (Aalst et al. 2003). Börger (2012) criticises the increase in the number of patterns. It started with 20 patterns in 2003, increased to 43 in 2006, and reaching 126 in 2010. Börger was concerned that the patterns quantity may detriment the concept of simplicity, succinctness and de-composition techniques with severe practical consequences.

In most business scenarios processes evolve and not all exceptions can be foreseen during build time or some processes require user intervention to deviate from the predefined process (Weber et al. 2009). Achieving compliance with control-flow and exception handling may increase process complexity that lead to practical concerns as highlighted by Börger above.

2.5.2 Business Process Modelling

There are two basic types of modelling language: graphical and textual (He et al. 2007). Graphical modelling languages use a diagrammatical technique and textual modelling languages use standardised keywords or natural languages. Users find it is very convenient to define process models using graphical tools. It is used as a bridge between the end user and IT developer so they can share a common language in order to describe how a process

achieves specific goals. Standard modelling languages were developed for process definition. Among the prominent standards are the Business Process Model and Notation (BPMN) by OMG (2011), XML Process Definition Language (XPDL) by WfMC (2012) and Web Services Business Process Execution (WS-BPEL) by OASIS (2007).

WS-BPEL is a standard execution language to specify business process behaviour based on Web Services interactions between the process and its partner. However, WS-BPEL does not contain elements to represent the graphical aspects of a process diagram. During the early steps of BPMN diagrams, WS-BPEL was the best choice to execute BPMN model. White (2005) provides an example of partial language mapping from BPMN to BPEL that can be used to generate BPEL code. However, White (2004) claimed that WS-BPEL had too many limitations to be considered the final choice to serialise BPMN diagrams. This is due to the richness of BPMN attributes and properties that had no complete correspondence representation in WS-BPEL.

In contrast, XPDL is a standard that was aimed to interchange business process definitions between different modelling tools and management suites. XPDL is designed to exchange process definitions, within the graphics and semantics of a workflow business process.

XPDL is currently the best file format for exchanging BPMN diagrams. This is because it has been designed specifically to store all aspects of a BPMN diagram. This differentiates XPDL from WS-BPEL which emphasises solely on the executable aspects of the process. Thus, XPDL has been widely adopted as a common standard interchange format for BPMN diagrams.

The first version of BPMN was published in 2004 by the Business Process Modelling Initiative. BPMN aims to provide business process graphical notation that starts from the business analyst that create and improve the process to the technical developer who is responsible for implementing the process, and finally to end users who deploy, monitor and manage such processes (White 2004). The increase of adoptions from companies and individuals caused the adaption of BPMN as OMG standard in 2006 as shown in Figure 2-4.

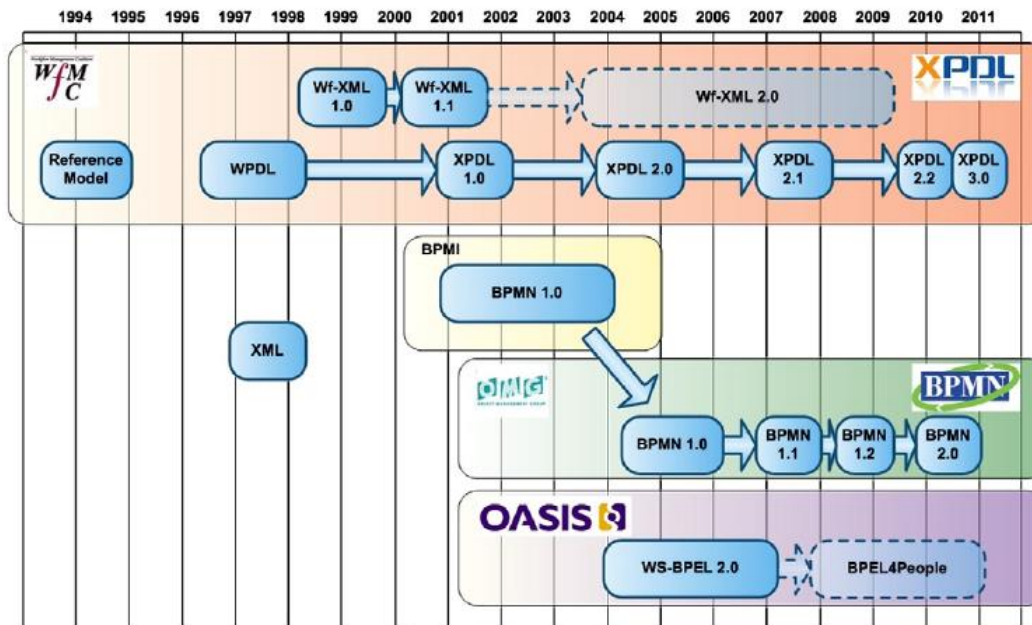


Figure 2-4 Business Process related standards time-line (adapted from Chinosi & Trombetta 2012)

The modelling language for representing the business process landscape changed after the introduction of BPMN version 2.0 (BPMN2), which was published in 2011. BPMN2 standard comes with dual functionality: diagrams to communicate and modelling for execution. BPMN2 provides end-to-end BPMN model: Processes (orchestration for private non-executable, private executable, and public), Choreographies and Collaborations.

Chinosi and Trombetta (2012) claimed that BPMN2 is the de-facto standards in representing business processes. A survey was conducted by BPTrends (Harmon & Wolf 2011) to determine which modelling standards are mostly used (respondent could choose more than once). They found that BPMN (72%) made up the vast majority. The next highest percentage, 33% chose “others” followed by UML (18%), BPEL (6%), and XPDL (4%). Among “others” standards found in the literature are: Petri Net, Yet Another Workflow Language (YAWL), workflow nets, and event-driven process chains (EPCs). For this reason, this Thesis utilises BPMN in defining and executing business process.

BPMN2 provides a graphical notation in order to represent a business process, but, it is also rigorously used as a language that can be used to generate code. Basic BPMN2 consists of five basic elements:

- Connecting or Flow objects: events, activities, gateways;
- Connecting objects: sequence flow, message flow, association;
- Swim lanes: pool, lane;
- Data: objects, inputs, outputs, stores;
- Artifacts: group, annotation

The basic BPMN modelling elements and description is available in Appendix 1.

2.6 Integrating Business Process and Business Rules

BPMN provides a graphical notation of business processes between tasks. The visual representation of BPMN emphasises the workflow process and describes activities of the organisation at an abstract level. However, it does not cover the low-level specification within the business process (Charfi & Mezini 2004). The business rules approach has been used to define precisely the logic of a process task in structured natural language (Knolmayer et al. 2000).

Even though there is a difference in abstraction levels of business process and business rules, combining both can complement each other (Kluza et al. 2012). BPMN model can be used to define the high-level behaviour of the system while the low-level process logic can be described by rules.

2.7 Existing Workflow Development tools

Two main types of products for Workflow Development tools are available: Closed source (proprietary) and Open source. The closed source WfM products are highly competitive. According to Jose (2011), Oracle and IBM are dominant players in the BPM sector. Other players include Adobe Systems Incorporated, Appian Corporation, Fujitsu America Inc., Global 360 Inc., Hewlett Packard Development Company, Intalio Inc., Metastorm Inc., Microgen plc., Microsoft Corporation, Pegasystems Inc., Progress Software Corporation, SAP AG, Software AG, TIBCO Software Inc., and Vitria Technology Inc.

Open source software is computer software that is available in source code form which usually permits users to study, change, improve and distribute the software. For open source WfMS products the software is provided for free. However, most of this software is supported by vendors such as Redhat, Alfresco, Bonisoft, and ProcessMaker. By means of

free software business model, the vendors usually may offer pay support and customisation services.

For the prototype development of this Thesis as a proof-of-concept, Open Source WfM is considered. This is due to the limited access (modification) on proprietary WfM/BPM software. According to Del Rio & Soluciones (2012), the three most prominent Open Source WfM tools are Bonita (Bonitasoft 2013), Activiti (Rademakers 2013) and jBPM (Redhat 2013). Table 2-1 shows the comparison of these tools

Table 2-1 WfM Open Source Software

Open Source WfM	Bonita	Activiti	jBPM
Features			
Process Modelling Language	BPMN2	BPMN2	BPMN2
Process Engine	Based on jBPM3	Based on jBPM4	Based on Drools Flow
Business Rule Integration	Service call	Service call	Instance level
Web-based Process editor	No	Yes (based on Oryx Designer)	Yes (based on Oryx Designer)
Business Activity Monitoring (BAM)	Yes	Yes	Yes
Maturity Level	Current Version 5.7.2 (based on jBPM 3)	Current Version 5.9 (based on jBPM 4) started in 2010	Current version 5.4 (16 Nov 2012), Integration with JBoss Drools
License	GNU General Public License v2.	Apache License V2	Engine: Apache License, Eclipse Designer: EPL, and Web-Based Modeller: MIT
Supported by	Bonisoft	Alfresco	Redhat
Manage adaptation and compliance Validation	No	No	No

To refine the selection process of a prototype development platform, this Thesis put forward a list of criteria, as follows:

- Able to supports basic requirements of project aim and objectives ;
- Business rules integration;

- Supported by reputable organisation;
- High maturity level; and
- Sufficient community supports and documentations.

Based on the criteria above, jBPM match all the criteria especially for business rules integration, where complex business logic can be modelled as a combination of business processes and business rules on instance level. It has an active community and supported by Redhat. Bonita falls short on providing a web-based process editor. Since Activiti is based on jBPM 4, jBPM functionality is still use to provide process and workflow functionality in Activiti. However, none of the existing tools support the integration of adaptive and compliance validation.

2.8 Conclusion

This chapter reviewed key concepts of workflow technology on the generic workflow models, benefits, categories, standards, and development tools. Despite the variety of business process modelling languages available today, BPMN has become the de-facto standard in representing business processes. However, it is also observed that BPMN does not support any compliance features as part of their standards. It is clear from the comparison of the main Open Source WfM tools that none of them support the integration of adaptive and compliance validation.

As a response to the benefits and capabilities of using WfMS, more research is required to fill in gaps that were raised in this review. Particularly these are in the area of the integration of adaptive and compliance WfM capabilities, which is the focus of this Thesis. Adaptive workflow enables process changes demanded by practical situations through user and dynamic adaptation. Compliance in a workflow will ensure an organisation will comply with the requirements. However, both are required to work together, as having adaptive capability on its own may produce disastrous outcomes. Therefore, the following chapters discuss in detail the requirements and architecture of integrating compliance and adaptive features in a WfMS development tool.

Chapter 3

Adaptive and Compliance Workflow

3.1 Introduction

Given the dynamic processes of today's organisations, it is unlikely that some workflow processes can be modelled once and executed repeatedly without any changes. Processes, for example, may evolve to reflect the changing environment or requirements. Hence, there is a strong demand for adaptive workflow management system (WfMS) that allow flexible adaptation of processes (Kumar et al. 2010). However, updating workflow processes without any mechanism to ensure an updated process does not violate compliance requirements during build-time and run-time may lead to executing a non-compliant process. Thus, a novel solution of an integrated adaptive and compliance development tool is necessary to support the workflow process lifecycles.

The remainder of this chapter is structured as follows; Section 3.2 discusses the importance of dynamic processes in WfMS. Section 3.3 reviews the techniques for compliance checks in current WfMS. Section 3.5 identifies the requirements for enhancing the WfMS development tool to handle process adaptation and compliance validation. Section 3.5 discusses related work on integrated adaptive and compliance workflow development tools and evaluates each tool against the requirements identified in Section 3.4. And Section 3.6 concludes the chapter.

3.2 Dynamic Process Change

3.2.1 Introduction

Current WfMS are suitable for routine situations that demand efficient, consistent and accurate execution of standard processes. Workflow management systems (WfMS) have delivered a great deal of productivity improvements, however they have primarily been designed to support static and repetitive business process (Weber et al. 2009).

The issue of managing dynamic processes has been discussed by both academia and industry. This issue relates to the ability of an organisation to respond to changes in an efficient and effective way. Nunes (2011) identified that the concept of process flexibility is related to the need to understand situations that happen while people, systems and resources interact and demand adjustments.

Although there are several success stories on the usage on WfMS for static and repetitive process, they have not had the widespread adoption that was expected (Weber et al. 2009). One of the major reasons for this is the limited support of dynamic changes and the inability to respond to business changes (Mutschler 2008; Aalst & Jablonski 2000). A number of techniques to support flexible and adaptive workflow are identified and discussed in this chapter.

3.2.2 Techniques for Supporting Flexible and Adaptive Workflow

Two main techniques for supporting flexible and adaptive workflow suggested by researchers are *adaptation* and *selection* (Nurcan 2008). Adaptation is the more common approach and is used when the required changes were not anticipated during build-time. Selection, on the other hand, is based on a modelling formalism that offers flexibility through situational changes without any evolution of process definition. The ability to follow different paths thus needs to be incorporated in the process definition during build-time.

It is possible to take advantage of both approaches. By applying both techniques, the WfMS will be able to respond to different kinds of changes. For example, if not all the process paths are clear at build-time, the user may utilise a process adaptation

technique for unseen paths, and if part of the process includes alternative paths that can be defined at build-time then selection could be applied to the process definition. This will reduce time in monitoring and making unnecessary changes in the process definitions.

One of the challenges highlighted by Nurcan (2008) is 'how to deal with the process instances which are currently running? '. To deal with this challenge, migration techniques have been proposed to deal with process instances that are affected. Nurcan (2008) classified these techniques into:

- Cancellation: affected instances are cancelled and new instances are created according to the new process definition. Kradolfer & Geppert (1999) suggested this technique is least desirable due to the loss of information and time;
- With propagation: the modification of the process definition is propagated to the affected instances. This technique requires a transition model by validating the affected instances against a target process definition (Kradolfer & Geppert 1999);
- Without propagation: affected instances continue executing based on the old process definitions while new instances are executed according to the new process definitions. This technique requires version control where the migration is delayed until the old process definition reaches a safe state (Agostini & Michelis 2000; Mathias Weske 2001).

The migration 'with propagation' technique is the most useful of the three techniques. It delivers real-time modification of process definition to support the required changes and brings the affected instances into compliance with the modified process. This technique is viable, if the instance in its current execution state is compliant with the new process definition. The following section discusses process adaptation in more detail.

3.2.3 Process Adaptation

To cope with evolving processes, exceptions and uncertainty, the WfMS needs to be able to deal with structural process adaptation (Weber et al. 2008). To review process adaptation, this Thesis uses the taxonomy presented by Weber et al. (2008), as shown in Figure 3-1.

Process adaptations can be triggered and performed at the process definition (schema) and process instance level (De Leoni et al. 2007). Schema evolution becomes necessary to address the evolving nature of business processes, such as when dealing with changes in organisation requirements. Figure 3-1a illustrates process definition changes where insertion of two additional activities X and Y transform the schema from S to S'. In such situations it is necessary to propagate the changes to ongoing process instances that may run at different stages as reflected in Figure 3-1a. Changes can be propagated and migrated to the running instances if these instances are compliant with schema S'. In this example instances I₁ and I₂; while I₃ has progressed too far and therefore has to be completed based on original schema S.

Ad-hoc situations usually deal with exceptions or unanticipated situations, which result in an adapted instance-specific process schema and do not affect any other ongoing process instances (Reichert et al. 2003). In Figure 3-1b, instance I₄ has been individually changed by inserting activity X and by deleting activity F.

Flexibility can also be achieved by leaving parts of the process definition unspecified at build-time and by adding the missing information during run-time (Aalst, 2009). This built-in flexibility approach is useful in cases of uncertainty by leaving the user to make decisions, during run-time, once information becomes available. Figure 3-1c illustrates a process schema with fragments of placeholder B with four activities S, T, U and V which can be used during run-time to substitute placeholder activity B. Instances I₅ and I₆ constitute two valid changes, which can be created based on process schema S.

Making changes may lead to a non-compliance process and even violate organisation policies. Manually verifying the correctness of a process definition and instances is error prone and infeasible when processes are large and complicated (Kumar et al. 2012). Therefore, there is a need to manage compliance for dynamic and complex processes (Chung et al. 2008). This is discussed in Section 3.3.

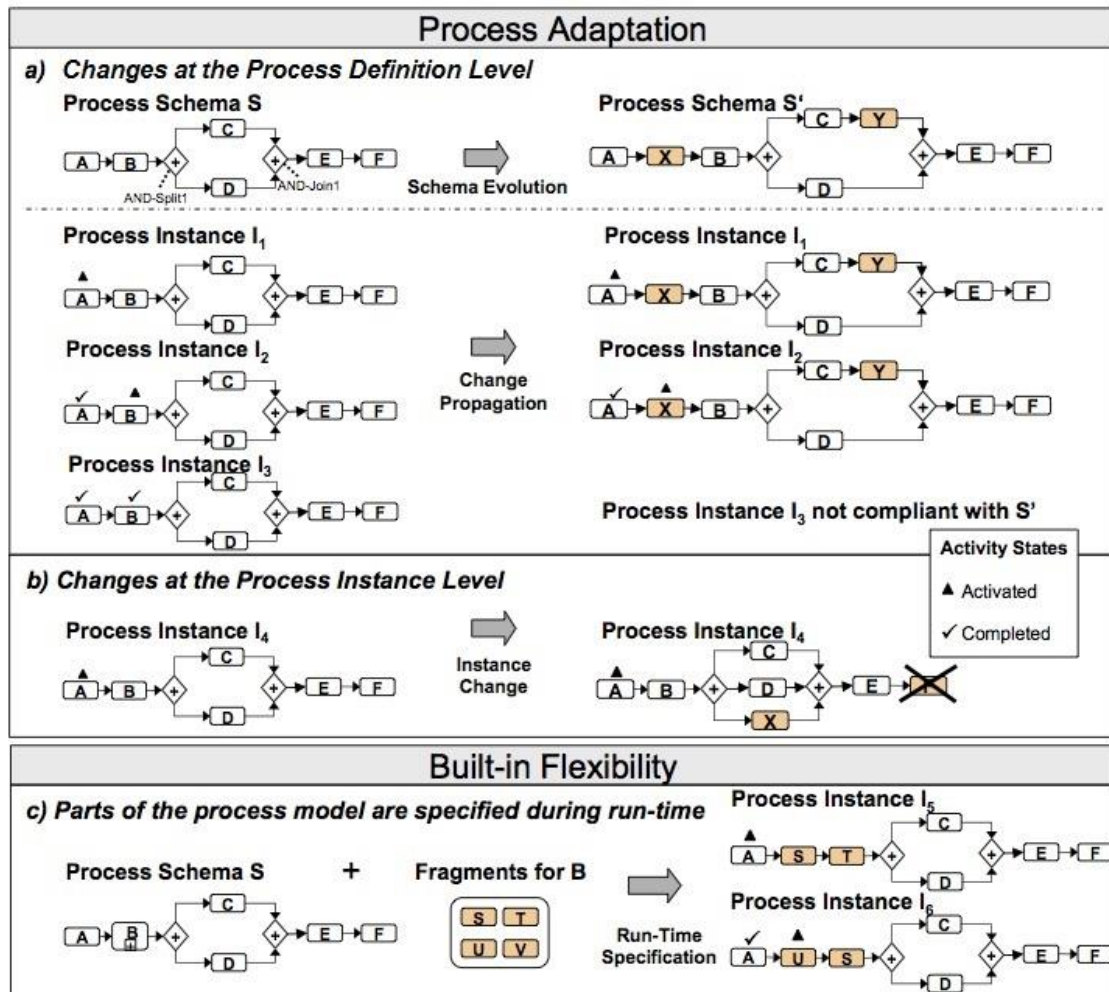


Figure 3-1 Process Adaptation and Built-in Flexibility (adapted from Weber et al. 2008)

The built-in flexibility described above is triggered manually on user decision. In circumstances where logic can be specified, it can be triggered dynamically using rule-based business logic (Weber et al. 2009). Adaptation with business rules is discussed in the following section.

3.2.4 Process Adaptation with Business Rules

Business rules were discussed in Section 2.5; the integration solutions of business process and business rules in a WfMS can complement each other. Further, business rules have been used to improve process flexibility (Goedertier & Vanthienen 2007).

Asuncion et al. (2010) proposed an approach towards the integration solutions by separating the more dynamic aspects of the requirements as business rules while keeping the more stable parts in the business process. Hence, with the separate representation of both solutions, a change between the other solutions does not affect the other adversely. It is important to ensure that process flexibility does not violate compliance requirements. Compliance supports in WfMS are discussed in following section.

3.3 Compliance supports

3.3.1 Introduction

Recently, several techniques have been developed to validate business processes for compliance requirements. As stated in Section 1.2, automating compliance can be achieved either 'by design' or 'by detection'. The Forward (by design) and Backward (by detection) classification of compliance automation that was proposed by Kharbili et al. (2008) is not helpful, as both techniques are more or less implementing the same compliance check. The classification would be more useful if the automation of compliance checks is classified according to the stages when checking is done: compliance validation at build-time, compliance monitoring at run-time and compliance auditing of the completed business process execution.

Build-time compliance checking and run-time compliance monitoring ensures and manages the verification of requirements before and during process instance execution. Thus, these techniques can prevent the actual execution of non-compliant behaviour. However, as for the compliance auditing technique, which is based on the after-the-fact principle, it is found to be less sustainable and ineffective for compliance management (Schumm et al. 2010).

3.3.2 Build-Time Compliance Checking

Build-time compliance checking techniques aim at guaranteeing that process instances are free from compliance violations. Some techniques guide the user during the modelling phase. Other techniques are by definition checking that verifies certain properties exist with existing definitions (Kharbili et al. 2008; Chung et al. 2008). Chung et al. (2008) approached it by using reference model that capture the main elements of compliance requirements as standard model. They suggest compliance checks are dealt with checking a user-defined process against selected standard model. It includes:

- Correctness Check – To ensure the sequence of tasks specified in a user-defined process is in accordance with a selected standard;
- Completeness Check – To ensure all the required tasks within a standard are defined in the user-defined process;
- Capability Check – To ensure the required capabilities of an agent are specified according to a selected standard;
- Recommendation Check – To ensure the recommended techniques, measures, tools or methods for performing a particular task are fully considered.

The reference model technique for compliance check is furthered discussed in Section 3.3.5.

3.3.3 Run-time Compliance Checking

Run-time compliance checking techniques are intended for executable process definitions and subsequently depend on process execution architectures and mechanisms. Generally, this is done by defining compliance regulations either into the process definition (Koehler & Vanhatalo 2007) or it requires run-time information (Fötsch et al. 2006).

According to Charfi & Mezini (2004), defining compliance regulation into the process definition would make process becomes complex and hard to maintain. The process may contain plenty of nested conditional activities to model decision-making point.

To overcome the issue, they propose an integration technique of business process and business rules.

Compliance validation that requires run-time information can be validated with business rules, such as occurrence check e.g. if a customer is frequent customer, he gets a discount of 5%.

3.3.4 Compliance Auditing

Compliance auditing techniques examine completed process instances (Becker et al. 2011). Rozinat & Aalst (2008) developed conformance-checking technique that checks the control flow of process definitions and match them with a certain process instance to show any compliance violation. Whenever non-compliances are detected, the conformance checking provides an indication of where the differences occur by means of graphical notation. The non-compliance indication is restricted to control-flow-related constraints. Thus, no rules involving data fields can be checked.

3.3.5 Integrating Build-Time and Run-Time

For the integration of build-time and run-time compliance supports, Chung et al. (2008) proposed a Compliance Flow system. Their proposed system executes compliance checks during build-time and performs error prevention at run-time in order to identify and prevent the execution of non-complaint tasks.

The work of Chung et al. (2008) provides a useful inspiration for this Thesis. They proposed a two-tier reference model that separates the standard model and the user-defined process. This technique is suitable with user adaptation. However, when dealing with dynamic adaptation, where processes evolve as they progress according to situations that cannot always be prescribed, another layer of the adaptive standard model is needed to propagate the changes.

3.4 Requirements for integrated Process Adaptation and Compliance

Based on the summaries of both process adaptation and compliance techniques above, a number of requirements can be identified to bridge the automation of integrated process adaptation and compliance supports within WfMS. The following is a list of these requirements:

Requirement 1: *Representation for Business Process Modelling and Execution*

Apply standards of business process language (BPMN2) by means of graphical modelling and in turn, interchange to semantic execution for WfMS.

Requirement 2: *Representation of Compliance Requirements into Standard Model*

Capture compliance requirements from standards or policies and translate into the standard model. The standard model should be able to represent compliance requirements and validations at the processes structure level. This requirement will enable consistency in gathering and checking compliance rules.

Requirement 3: *Representation of Compliance Requirements into Business Rules*

Following requirement 2, business rules may represent compliance requirements and validation at tasks and data level.

Requirement 4: *Adaptation with Dynamic change and User Intervention*

Provide adaptation support dynamically (automated) according to business rule logic at run-time or through user intervention both at design-time or run-time and in turn make sure updated processes do not violate compliance requirements.

Requirement 5: *Evolution for Adaptive (non-Static) Standard Model*

Process planning normally starts from an abstract process and becomes more concrete as planning progresses. After a while a user defined process may evolve due to changing needs of the user or case. At each moment in time, a workflow instance is attached to a single adaptive standard model. This requirement ensures the concept of the reference model is always valid for compliance checking within the context of adaptive workflow.

Requirement 6: *Compliance Validation at Structural level.*

Generally, user-defined processes derive from the standard model that acts as a process template. Activity in user-defined processes can be further extended according to user needs. As changes may lead to errors, it is vital to provide support to visually validate structural compliance checks against the adaptive standard model both during build-time and run-time.

Requirement 7: *Compliance Validation at Event and Data Level*

This requirement is intended for checking process instances at run-time. It may deal with real-time data and events, it is important to provide compliance check at this level to ensure process instances comply with compliance requirements.

Requirement 8: *Tracking Instance Support*

To provide a flexible process in the WfMS it is vital to have an efficient tracker to monitor the activity states of instances. The instance tracker should signal the process editor to lock those activities for any further changes to avoid any data and control loss.

Requirement 9: *Migration Instance Support*

Any changes in user-defined processes at run-time that involves running instances requires the running instance to be migrated with the updated user-defined process.

The above requirements take the two techniques of adaptive and compliance workflow that have to be integrated together. The requirements outline an intelligent development tool to automate compliance validation and manage adaptation within a WfMS. This set of requirements will guide the contribution of this Thesis.

3.5 Adaptive and Compliance Integrated Workflow Development Tools

This section discusses related work on adaptive and compliance integrated workflow development tools and will evaluate each against the requirements introduced in the previous section.

ADEPTflex (Reichert et al. 2003) supports users in modifying the structure of WfMS at runtime, while maintaining correctness and consistency. Correctness properties defined by the ADEPTflex model are used to determine whether a specific change can be applied to a given workflow or not. If these properties are violated the change is either rejected or the correctness must be restored by handling the exceptions resulting from the change. However, ADEPTflex uses a manual approach where the user has to decide which events constitute logical failures and which

adaptations have to be performed (Kumar et al. 2012), which does not satisfy Requirement 4 on dynamic adaptation.

DECLARE (Pesic 2007) is a WfMS prototype that uses a constraint-based (rules) process modelling language for the development of declarative models describing loosely-structured process. DECLARE is not particularly suitable for modelling large or highly-structured processes. In both cases, DECLARE would have many constraints, which can easily introduce errors during process development. It is hard for users to understand the whole model during execution and even contribute to performance issues. The declarative technique limits compliance check by enforcing the required constraints among tasks. This does not satisfy Requirements 2 and 3, where compliance representation and check should support both at the process and task level.

SeaFlows (Ly et al. 2011) supports semantic constraints in a process management system. The framework points out the need for a separate constraint (rules) repository. SeaFlows distinguishes itself in support at design time validation and the support of controlling instance adaptation at runtime. To support at design time validation, SeaFlows identifies full compliance, definite violation and conditional violation. However, the framework only provides textual description of violations feedback to user - thus Requirement 6 is not addressed.

ADEPTflex and DECLARE embed compliance requirements within the process definition. Separation of compliance requirements through descriptive business rules and the standard model will increase consistency in checking for compliance violations (Requirements 2 and 3). In addition, validation of process changes has not been addressed (Requirements 6 and 7). Although SeaFlows supports the separation of compliance requirements from the process definition, it does not support adaptation with user intervention (Requirement 4). Further, SeaFlows does not support compliance validation during build-time (Requirement 6).

None of these tools address the requirements defined in the previous section. Therefore, a novel solution that integrates adaptive and compliance workflow in a

single development tool is essential. The development tool should support user and dynamic process adaptation that conform to certain standard requirements.

3.6 Conclusion

This chapter has studied key aspects of integrating adaptive and compliance checking techniques in WfMS. Flexible workflows techniques are categorised either through adaption or selection. Applying both techniques will improve workflow enabling it to be more dynamic, robust and time saving. Adaptation though built-in techniques can also be used to achieve workflow process flexibility by leaving parts of the process definition unspecified at build-time and by adding the missing information during run-time. Migration techniques are classified as cancellation, with propagation and without propagation. The migration 'with propagation' technique delivers real-time impact to the current instances.

The compliance support on forward compliance (by design) and backward compliance (by detection) were reviewed. The classification would be more meaningful according to the timing of when the checking is done. This can be classified as: build time compliance checking, run-time compliance monitoring and compliance auditing. The integration during both stages of build-time and run-time is important in order to identify compliance errors, assist in process specification and prevent non-compliant tasks from being performed accidentally. Further, three-tier models are essential to accommodate dynamic adaptation in making sure the concept of a reference model will still be valid for compliance check within the adaptive workflow environment.

Nine requirements were presented for integrating process adaptation and compliance techniques in a WfMS. These requirements were evaluated against existing adaptive and compliance integrated workflow development tools. There is no technique that covers all compliance validation scenarios and ensures compliance over a dynamic workflow lifecycle. To address these issues, Chapter 5 proposes a novel architecture for an integrated workflow development tool that handles process adaptation and compliance validation.

Chapter 4

Research Methodology

4.1 Introduction

This chapter presents a research methodology that provides the overall approach and strategy used in conducting this research. It is important to use appropriate research design for collecting and analysing data in order to ensure meaningful research results. Further, it helps in producing evidence to evaluate the aim of this Thesis that is on providing novel system architecture to automate compliance validation and adaptive workflow solutions.

The remainder of this chapter is structured as follows: Section 4.2 discusses the research methodology and approach used to achieve the aim and objectives of this research. Section 4.3 introduces the postgraduate research (PGR) process as the case study domain for this research and Section 4.5 concludes this chapter.

4.2 Research Methodology and Approach

The findings from the literature reviews (Chapters 2 and 3) and the system requirements (Section 3.4) are used to design the proposed system architecture. A research prototype - adaptive and compliance workflow management system (ACWfMS) - is developed based on the proposed system architecture to implement core services of the system for testing and evaluation purposes.

4.2.1 Systems Architecture Design

Over the past decade, system architecture has received increasing attention as an important subfield of software engineering (Garlan 2000). Dijkstra (1983) pointed out that it pays to consider how to structure a program, not just how to compute the correct answer. System architecture is often the first design artefact that represents decisions on how requirements of all types are to be achieved (Kazman et al. 1996).

Architecture defines the system elements and how they interact and provides a partial blueprint for development by indicating the major components and dependencies between them (Garlan 2000). Critical evaluation of the architecture provides a clearer understanding of requirements, implementation strategies and potential risks (Boehm et al. 1995).

With these rationales a system architecture approach is considered as the main blueprint of the proposed system. The design of the system architecture is based on the system requirements as identified in Section 3.4 and is translated into the ACWfMS prototype implementation for evaluation purposes.

4.2.2 Prototype Development

The traditional software development lifecycle models can be used in projects where the problem is well defined, the requirements can be clearly elicited and specified, and the technical feasibility of a solution is well understood (Dawson 2005). According to Dawson (2005), in many projects it is often difficult to pin down exactly what is required for a software system at the start of the project, particularly in a developing research area. In this case, Dawson (2005) suggests producing a prototype to explore the requirements of the system with the user and to explore the technical feasibility of a system.

Alavi (1984) conducted a two-phased research project comparing the prototyping approach with the more traditional life cycle approach and found that prototyping facilitates communication between users and designers during the design process. However, the findings also indicate that designers who used prototyping experienced difficulties in managing and controlling the design process. The conclusion is that “Although there are pitfalls and shortcomings, none seem troublesome enough to outweigh the potential benefits”.

This Thesis approaches the prototyping of ACWfMS development by adapting Naumann & Jenkins (1982) four steps interactive process:

- Identifying basic user requirements;
- Developing a working prototype;

- Implementing and using the prototype;
- Revising and enhancing the prototype.

According to Naumann & Jenkins (1982), prototyping is a four step interactive process between user and builder (Figure 4-2). An initial version is defined, constructed and used. At the same time problems are discovered and corrected as revisions and enhancements to the working system.

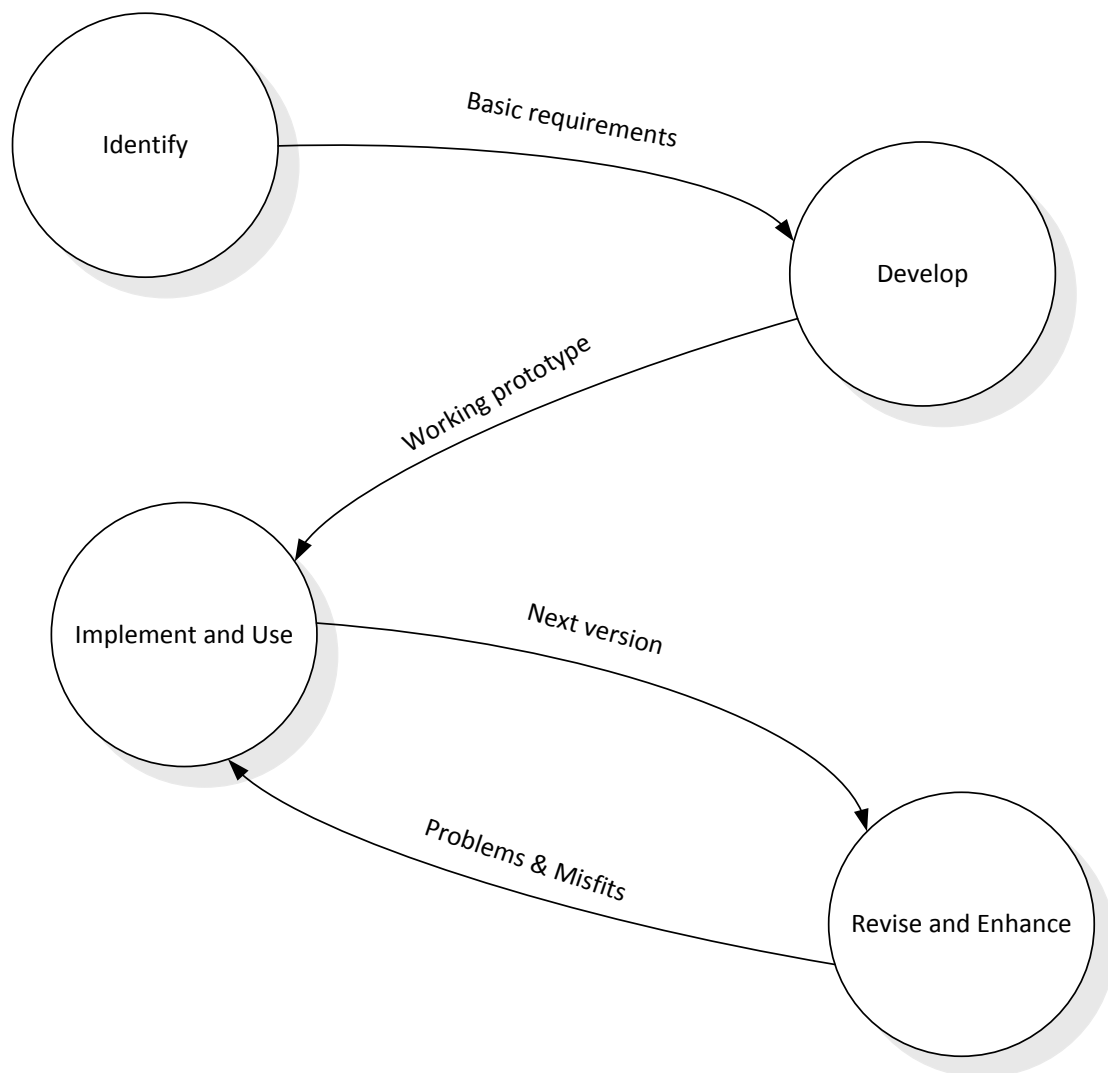


Figure 4-1 Prototype Model (adapted from Naumann & Jenkins 1982)

The first step of prototyping is to identify basic requirements. Nevertheless, this step also identifies features of user requirements that deals with human computer interaction such as providing a graphical interface for modelling process and

feedback supports on non-compliant processes. In order to implement and use the prototype system (Step 4), a business process and rules need to be defined and modelled. At this stage the process and compliance requirements are gathered based on the identified case study domain (PGR process). The process and compliance requirements are compiled from organisation policy.

The second step is to develop a working prototype. As discussed in Section 2.7, jBPM WfMS is considered for the prototyping development. Since jBPM is an open source WfMS, the source code is available to be used and modify according to the system and user requirements. Once a working prototype is ready, the user has a tangible system to experience and critique, in which the builder gets responses based upon that experience (Naumann & Jenkins 1982).

The third step is to implement and use the prototype system. According to Naumann & Jenkins (1982), hands-on use of the system provides experience, understanding and evaluation. Further, Knott & Dawson (1999) pointed out that; “Prototyping provides an effective method for generating feedback about what is good and what is bad about an idea and it is often the only really effective method for doing this”. When the user realises that they can make changes and influence the system, they are willing to participate with the system development with more dedication (Earl 1982) and, in turn, the developer and the user becomes partners. This partnership brings about a more robust system (Schrage 2004). At this stage, business processes and rules can be modelled and defined using the prototype system based on user requirements gathered in Step 1. System requirements are tested with the modelled process.

The fourth step is to revise and enhance the prototype system. Identified errors and missing features from the user and system requirements need to be addressed. It is important to implement and use a working prototype to get feedback and rectify all remaining problems. Naumann & Jenkins (1982) pointed out that steps 3 and 4 must be repeated until the user accepts the system as a good fit to their requirements. Further, all system requirements must fulfil the aim and objectives of this study.

Adapting a prototyping technique is a useful tool to prove a concept. Instead of having users to use and evaluate the prototype, this research uses a case study approach due to time constraint and the limited maturity of the prototype. The case study approach is appropriate for a research project of this nature as detailed below.

4.2.3 Case Study Research

Yin (2003) defines case study research as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.”

According to Runeson & Höst (2009), case study is well suited as a research methodology for software engineering research. However, before choosing a case study approach as the method of choice for evaluation of this Thesis, other major research methodologies were also considered. Among them were: Survey, Experiment, and Action Research. However, it was found that a case study approach excels at bringing the understanding of complex real life issues that involve humans and their interactions with technology (Runeson & Höst 2009). Case studies are commonly used to evaluate programs with the goal of identifying potential explanations for their successes or failures.

There are five major process steps to be considered when conducting case study research (Runeson & Höst 2009):

1. Case study design: objectives are defined and the case study is planned;
2. Preparation for data collection: procedures and protocols for data collection are defined;
3. Collecting evidence: execution with data collection on the studied case;
4. Analysis of collected data; and
5. Reporting.

Robson (2002) classified four types of purpose for research: Exploratory, Descriptive, Explanatory and Improving. This Thesis takes on the improving approach since the main aim of this research is to investigate existing workflow management system with adaptive and compliance capabilities.

The purpose of the data collected is to provide insight into the phenomenon being studied (Lethbridge et al. 2005). In this study, the data need to be analysed in order to provide evidence that the proposed development tool has reached its purpose by validating the results against the identified system requirements (Section 3.4). The result thus determines the effectiveness of the proposed system architecture.

A scenario-based approach is applied to collect evidence for case study analysis. Using scenario-based will enable this research to gain information on how ACWfMS satisfies the system requirements in various user contexts. The scenario-based analysis is discussed further in the Section 4.4.

The report communicates the findings of the study and evaluates the quality of the study. This study adopted the Robson (2002) case study report characteristics:

1. Tell what the study was about;
2. Communicate a clear sense of the build case;
3. Provide a history of the inquiry, to provide information of what was done, by whom and how it was done;
4. Provide basic data in focused form, so the reader can make sure that the conclusions are reasonable; and
5. Articulate conclusions and set into a context they affect.

4.3 Case Study Domain Selection

As stated earlier the Postgraduate Research (PGR) process was identified as the case study domain. The main challenge of implementing PGR with generic WfMS is the nature of the PGR process, that each PGR student has a unique process which cannot be pre-determined and complex. Unanticipated activities, such as student requesting leave-of-absence which might or might not happen or could happen at any time, requires user intervention to be added into the workflow process during run-time. These changes need to be validated to make sure the updates process are still compliant with PGR code of practice.

Further, certain activities need to be dynamically updated to accommodate PGR requirements, such as the student progress review is found unsatisfactory and they

are advised to redo, resubmit or terminate; and the student is ready to submit their Thesis in the second year of their studies. Therefore, the PGR process provides sufficient evaluation scenarios to validate the proposed ACWfMS architecture and evaluate the prototype as a proof of concept, specifically on the adaptation and compliance requirements.

4.4 Scenario-Based

This research evaluate the proposed ACWfMS architecture to determine its fitness through the implementation of an ACWfMS prototype that implements the key components of the architecture for testing and evaluation purposes, specifically on the adaptation and compliance requirements. Kazman et al. (1996) point out that it is too difficult to analyse an architecture based on their abstract qualities which are too vague and provide very little procedural support. To address this problem, an approach that uses scenarios is used (Carroll 2000; Sutcliffe 2003).

Scenarios are brief narratives of expected or anticipated system uses from both user and developer views that provide a look at how the system satisfies quality attributes in various user contexts (Kazman et al. 1996). Several methods to support the analysis of software architecture quality attributes are available. Babar & Gorton (2004) made a comparison and revealed that some methods overlap. They listed five common activities that can form a generic process for system architecture evaluation:

1. Evaluation planning and preparation;
2. Explain system or software architecture approaches;
3. Elicit quality sensitive scenarios;
4. Analyse system or software architecture approaches; and
5. Interpret and present results.

The above activities are used as guidelines to evaluate ACWfMS prototype. Scenario-based evaluation is part of the data gathering method for the case study analysis.

4.5 Conclusion

The details of the research methodology and evaluation methods are presented in this chapter. This chapter discusses the approach of system architecture to design ACWfMS as the main blueprint. ACWfMS is a prototype implementation that is used in a case study for evaluation purposes. The case study approach is particularly useful for the current study in that it brings the understanding of complex real life issues that involve humans and their interactions within workflow technology. A scenario-based approach is used for gathering data for case study analysis. The major steps in conducting a case study was outlined and discussed. Justification for PGR process as the case study domain was covered. The detail process models and scenarios for testing and evaluation are reported in Chapter 6. The following chapter presents the proposed ACWfMS architecture and discusses the implementation of the ACWfMS prototype.

Chapter 5

System Design and Implementation

5.1 Introduction

Chapters 2 and 3 identified gaps which indicated the lack of any integrated development tool covering all compliance validation over the dynamic nature of business processes.

This chapter discusses a novel conceptual design, and prototype implementation of an integrated adaptive and compliance check capability in a workflow development tool. The system requirements identified in Section 3.4 are used to guide the design and implement the proposed Adaptive and Compliance Workflow Management System (referred to here as ACWfMS).

The remainder of this chapter is structured as follows: Section 5.2 discusses the conceptual design of the proposed ACWfMS which includes the detailed concept of the process model, business rules, functionality of each proposed modules and process adaptation concepts. Section 5.3 discusses the ACWfMS prototype implementation and Section 5.4 concludes this chapter.

5.2 Conceptual Design

5.2.1 Introduction

Typical workflow processes may enact a process definition and execute several thousand instances by a workflow engine without any changes. This is suitable for routine business processes. However, when business processes need flexibility,

adaptive features are needed. Updating processes may violate compliance requirements so automatic verification of compliance checking is necessary.

A high-level overview of the proposed ACWfMS architecture is shown in Figure 5-1. The major components of this architecture consist of process models, business rules and plugin modules. This architecture exhibits the concept of user adaptation with structural checks and dynamic adaptation with data-driven checks.

The following sections discuss the detailed functionality of ACWfMS components and underlying concepts that aim to support an integrated adaptive and compliance workflow solution.

5.2.2 Process Model

One of the main problems with a typical workflow system is the lack of ability to ensure that the specification and execution of a dynamic process is compliant with the standards. As discussed in Section 3.3.5, when dealing with dynamic adaptation, another layer of standard model is needed to propagate the changes. Therefore, the proposed ACWfMS architecture approaches this by separating the process into a three-tier model: *Standard Model (SM)*, *Adaptive Standard Model (ASM)* and *User Defined Process (UDP)*.

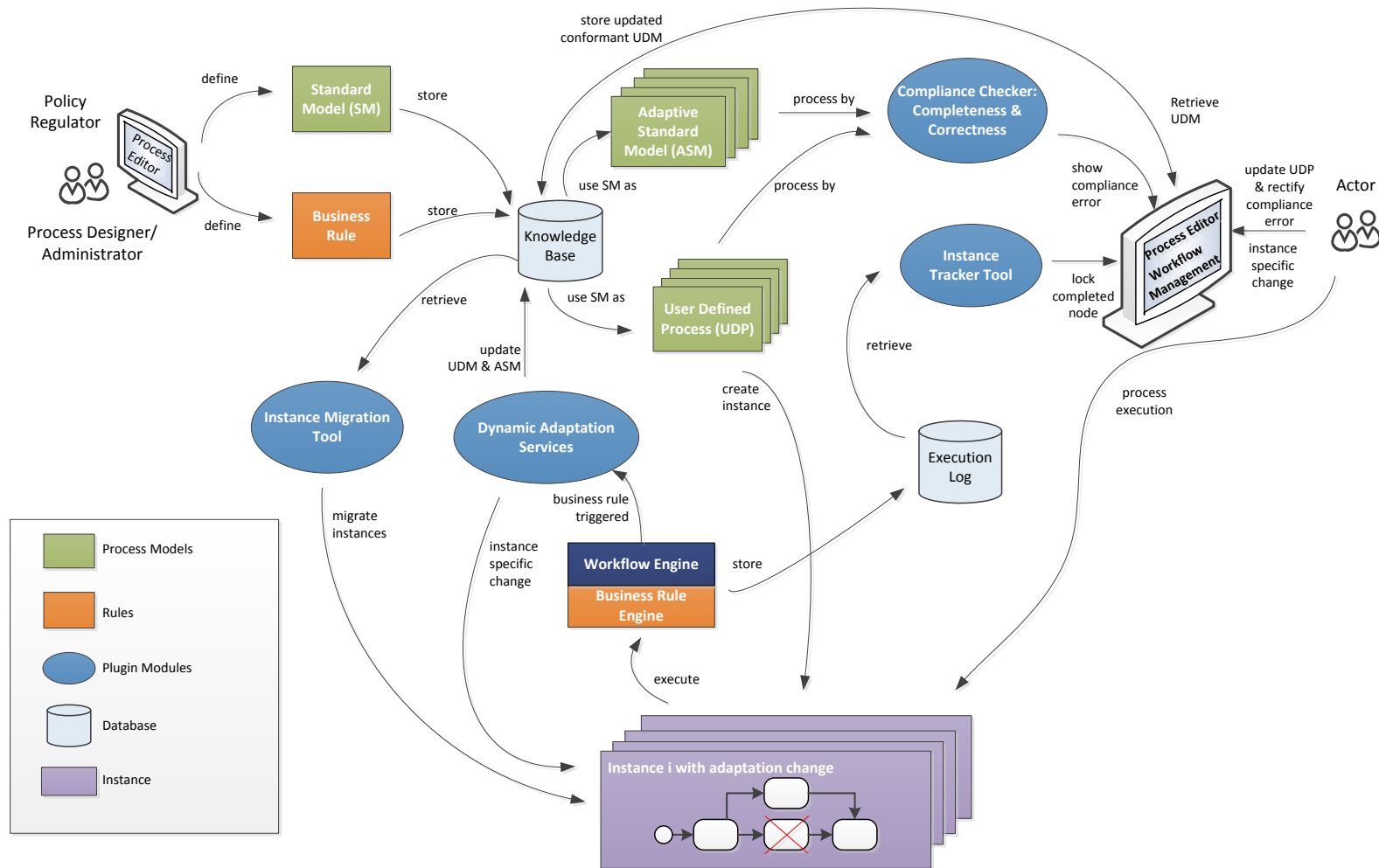


Figure 5-1 ACWfMS Architecture

Compliance requirements are derived from organisational policy. These requirements are usually written in text documents. Such documents could be difficult to understand by most people and cause misinterpretation. Using the graphical *Process Editor* to translate the requirements into a diagrammatical process model in turn provides clarity to understanding the abstract organisational policy concept. BPMN was discussed in detail in Section 2.4. BPMN notation is used to represent the graphical process model that is readily understandable by all users, analysts and technical developers.

To support the user in creating and updating compliant business processes at a structural level, ACWfMS applies the reference model technique. This technique separates the compliance requirement from the actual workflow process. The compliance requirements are modelled and represented as a Standard Model (SM).

Performing a task requires that the pre-conditions are completed successfully in advance. Therefore, the sequence of task executions is constrained by the tasks' dependencies. This principle should be reflected in designing the SM in order to achieve a valid compliance check. Any additional policy should be reflected in the SM to ensure the model is valid for compliance requirements. SM represents compliance requirements at the process structure level. It is used as the initial template for all process models.

Two-tier process models represent a collection of related process instances that may have to be adapted. However, to represent specific instance adaptations, changes to SM would cause other running instances to be non-compliant. The Adaptive Standard Model (ASM) is introduced to ensure the concept of a reference model will still be valid before and after making specific instance changes within the adaptive workflow environment. ASM will provide a consistent reference model and acts as a knowledge base to support compliance checks for specific instance process execution.

User Defined Process (UDP) is the actual business process that is used to manage the workflow during process enactment. It is used to represent each unique *instance(s) with adaptation change*. UDP is an executable BPMN process model. UDP is an

enhancement of the structure outline by SM. Updates during run-time on ad-hoc activities that cannot be anticipated during build-time are reflected in the UDP.

Figure 5-2 shows the concept of ASM as a non-static reference model to adapt as the UDP evolves (process adaptation change) by propagating the changes back to this model. At each moment, a UDP and workflow instance is attached to a single ASM.

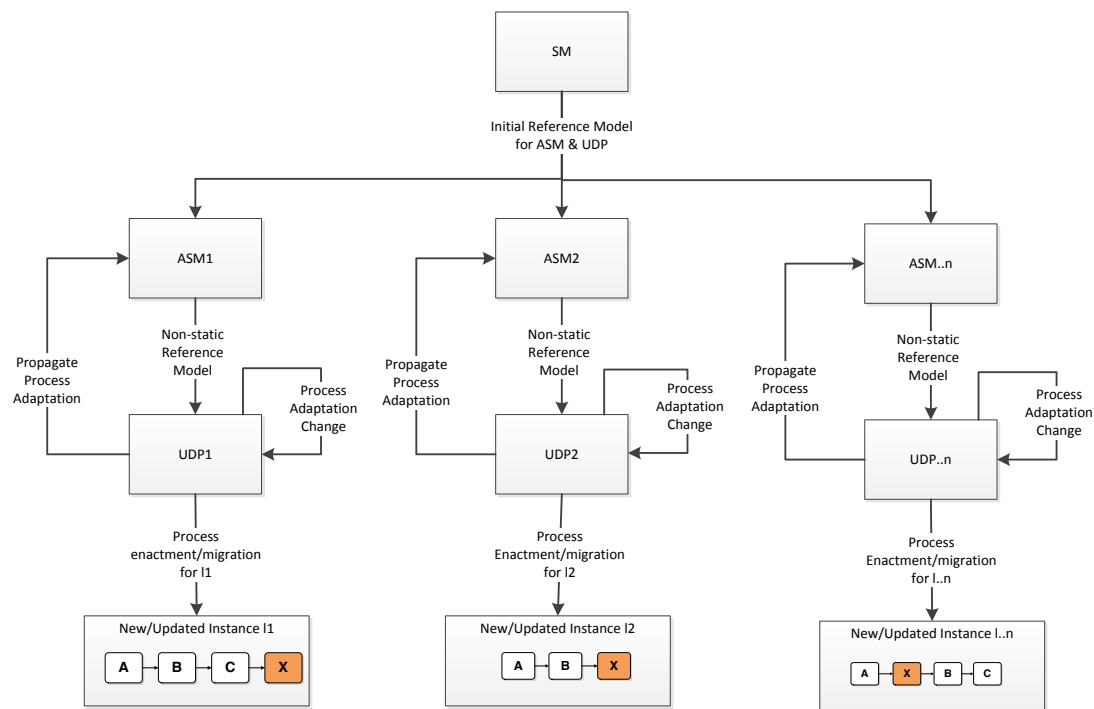


Figure 5-2 Three-Tier Non-Static Reference Model

All models are kept in a repository, which acts as *knowledge base* hub. A knowledge base repository is an important part of the ACWfMS architecture having the ability to undertake compliance checking and dynamically updating the business process.

5.2.3 Business Rules

Business rules were discussed in Sections 2.5 and 3.2.4; the integrated solutions of business process and business rules can complement each other. Compliance requirements may require that certain activities must be executed based on some sequence or logical choices of control and data flow within a business process. This process can be modelled explicitly. Schonenberg & Mans (2008) defined this as an imperative technique. It describes *how* the process is to be undertaken in a rigid manner that may produce a complex and hard to maintain business process. While

using business rules, it focuses on *what* the process should do rather than the *how* – defined as a *declarative technique* by Schonenberg & Mans (2008). It starts with the belief that everything should be allowed, unless explicitly prohibited. Through a declarative technique, the sequence of activities does not need to be determined early, which results in a more flexible process as the execution options (compliance requirements) are stated implicitly.

Drawing on the strength of both business process and business rules techniques, ACWfMS adapts a hybrid technique (Asuncion et al. 2010) to integrate business processes and business rules. The ACWfMS architecture applies business rules to support the enforcement of compliance validation that requires run-time information and supports dynamic adaptation offering run-time flexibility while leaving the more relevant and frequently used paths in the business process. All business rules definitions are kept in the knowledge base repository.

5.2.4 Plugin Modules

ACWfMS extends the WfMC workflow architecture by adding compliance checking and adaptive process functionality through plugin modules as describe below:

Instance Tracker Module: This module uses the Execution log data to track nodes within UDM that have been executed in the process instance. It signals the process editor to lock those executed nodes from any further changes. Thus, this module avoids any control and data losses.

Instance Migration Module: In a typical WfMS there is no link between a workflow instance and its process definition after the workflow instance has started, which implies that changing the process definition does not affect the running workflow instance. The affected instance needs to be migrated with the updated process definition. This module takes the UDP process definition ID and checks with the *Workflow Engine* services for running instances. If the condition is false, then a new instance will be initiated. On the other hand, existing process instances need to be migrated to the new UDP process definition. The migration allows the process to continue based on updated logic and copes with dynamic changes during run-time.

The *Workflow Management* is an interface for managing and executing workflow task with the actor. It is also used for managing process instances (start, stop and inspect). The Instance Migration Module uses the Workflow Management interface to execute a specific instance migration.

Dynamic Adaptation Module: This module dynamically changes process instances as required e.g. adding or deleting an activity. Simultaneously, it propagates the changes back to ASM. This module is triggered by the business rules during the dynamic adaptation process.

Structural Compliance Checker Module: This module validates the correctness and completeness of process flow for UDP against ASM. This module interacts with the *Process Editor* to enable the Process Designer or Actor to verify UDP during process build-time or run-time. Through the process editor, visual compliance feedback is generated to identify non-compliance process behaviour. UDP may be further modified until non-compliance is resolved.

5.2.5 User Adaptation with Structural Check

User adaptation is a technique of updating UDP by means of user intervention and, in turn, making sure updated processes do not violate compliance requirements. This technique accommodates process adaptation both during build-time and run-time. During build-time the initial UDP may be improved to suit an individual case. During run-time the user adaptation may support ad-hoc or unanticipated situations.

Figure 5-3 shows an example of user adaptation progression. In this example, changes are made on UDP by swapping the sequence of T0 and T1. D2 and T6 are deleted and T8 is added before T3. After changes are made, a compliance check for completeness and correctness is performed - the algorithms are discussed in the following sub-sections. If the compliance check is successful, the updated UDP is passed to the *Instance Migration Module*.

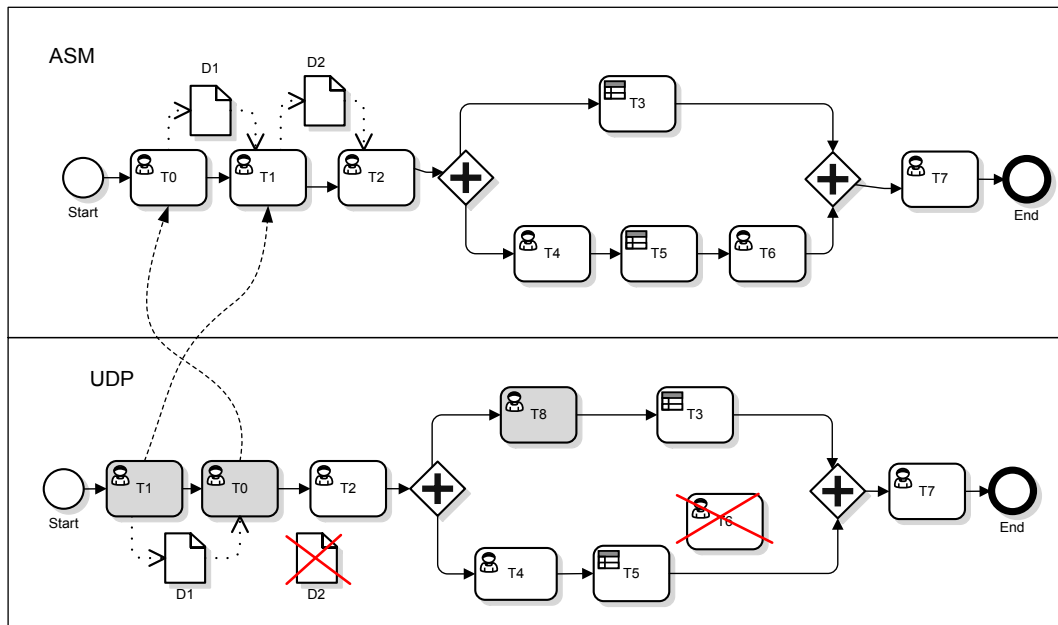


Figure 5-3 User Adaptation example showing ASM and UDP

Completeness check

In the example shown in Figure 5-3 the completeness check fails because Task T6, Data Object D2, Data Association T1 and T2, and Data Properties for T4 are missing in the UDP specification. The completeness check verifies that all ASMs are defined in UDP. If all specifications are found then the completeness check succeeds. If any violation occurs, the compliance checker signals back to the editor to produce appropriate error messages. The user needs to rectify errors in order to achieve the compliance standards.

The Flow elements completeness check algorithm which validates the Activities and Data Object is shown in Figure 5-5. It uses the FlowElement class (Figure 5-4) to collect Activities and Data Objects specifications of ASM and UDP into respective array lists. The ASM list is matched against the UDP list - unsuccessful matching of nodes are stored into an error array list.

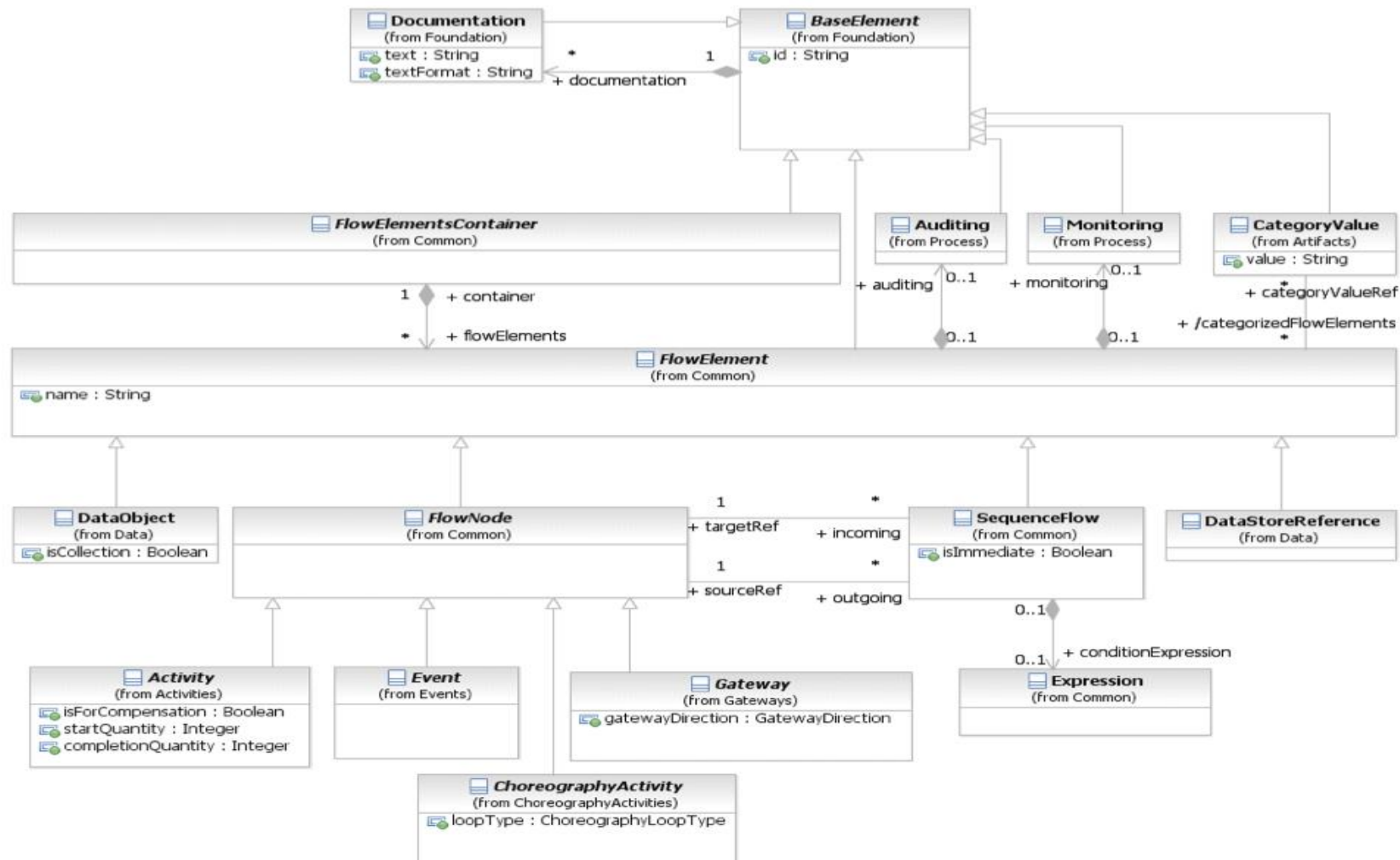


Figure 5-4 FlowElement Class Diagram (adapted from OMG 2011)

```

Function CompletenessCheck(udpDef, asmDef)
    //udpDef: process definition in a user-defined process for which the correctness
    //check is applied to.
    //asmDef: definition of the selected adaptive standard model.

    udpFlowList[...] = getFlowElements(udpDef) // flow nodes and data objects
    asmFlowList[...] = getFlowElements(smDef) // flow nodes and data objects

    For each i : asmFlowList[...]
        If Not udpFlowList[...] contains i Then
            errorList[..] = i
        End If
    Next i

    Return errorList[..]
End Function

```

Figure 5-5 Flow Elements Completeness Check algorithm

According to the BPMN OMG (2011) specification, process and activities require data in order to execute. These elements are aggregated in a BPMN InputOutputSpecification class as shown in Figure 5-6. Data Inputs and Outputs may have incoming and outgoing data associations. Only Data Inputs and Outputs that are contained by activities or events may be the target of Data Association in the model.

The Data Association completeness check algorithm, shown in Figure 5-7, captures incoming and outgoing data associations of ASM and UDP extracted from the 'DataInputAssociation' and 'DataOutputAssociation' classes into respective array maps. If UDP key matches the ASM key and if UDP value(s) contain an ASM data association value(s) then the data association completeness check is successful. Unavailable data associations are stored into an error array list.

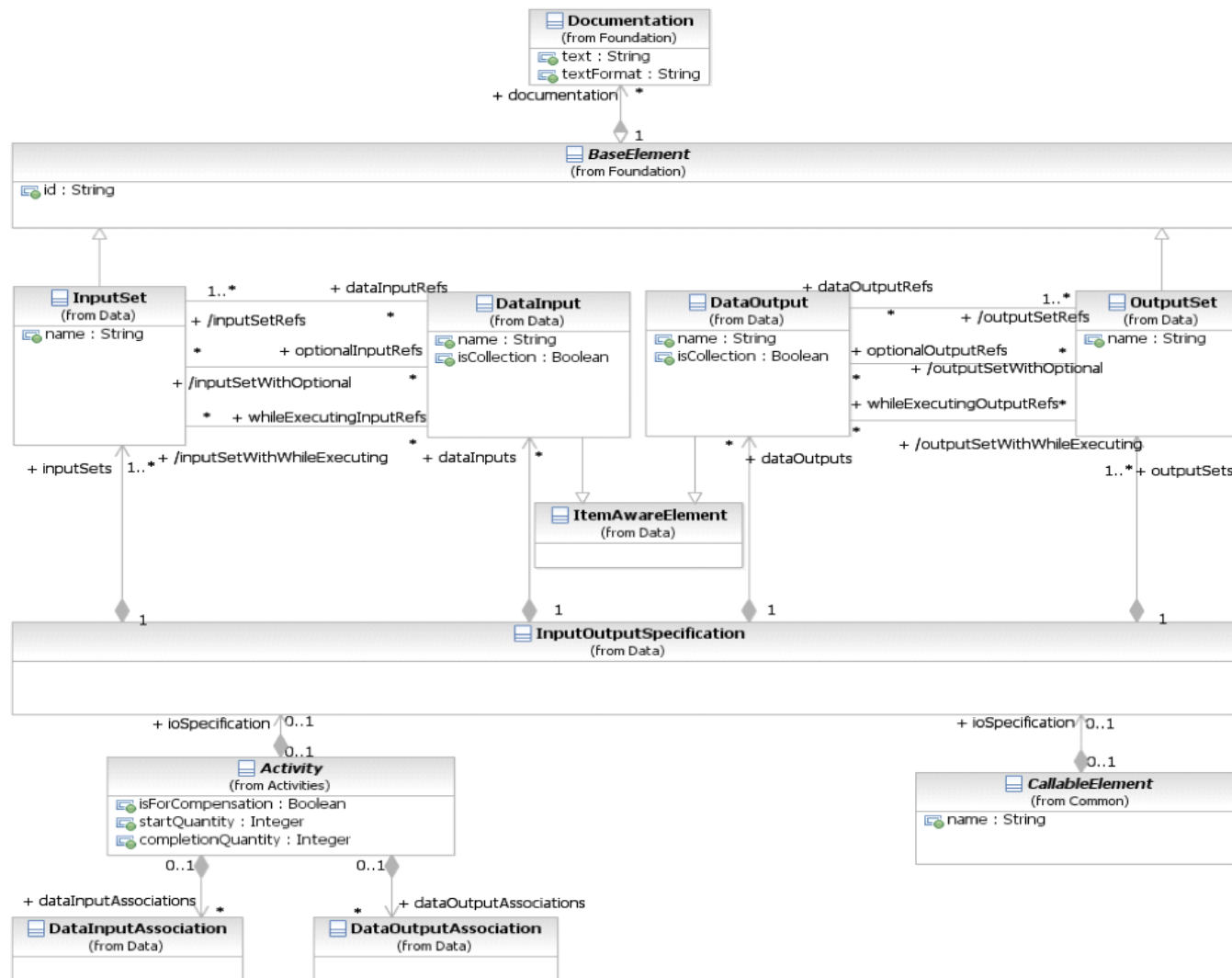


Figure 5-6 InputOutputSpecification Class Diagram (adapted from OMG 2011)

```

Function DataAssociationCheck(udpDef, asmDef)

    asmIncomingDataAssociationMap[...] = GetIncomingDataAssociationArtifacts(asmDef)
    asmOutgoingDataAssociationMap[...] = GetOutgoingDataAssociationArtifacts(asmDef)

    udpIncomingDataAssociationMap[...] = GetIncomingDataAssociationArtifacts(udpDef)
    udpOutgoingDataAssociationMap[...] = GetIncomingDataAssociationArtifacts(udpDef)

    For each i : asmIncomingDataAssociationMap[...]
        keyAsm = GetKkey( i )
        valueAsmList[...] = GetValues( keyAsm )
        if ( udpIncomingDataAssociationMap[...] is empty Then
            For each j : udpIncomingDataAssociationMap[...]
                keyUdp = GetKey( j )
                if keyAsm equals keyUdp Then
                    For each l : GetValues( keyUdp )
                        If Not ValueAsmList[...] contains l Then
                            errorList[...] = l missing association from keyAsm
                        Endif
                    Next l
                Endif
            Next j
        else
            For each m : GetValues( keyUdp )
                errorList[...] = m missing association from keyAsm
            Next m
        endif
    Next i

    For each i : asmOutgoingDataAssociationMap[...]
        keyAsm = GetKkey( i )
        valueAsmList[...] = GetValues( keyAsm )
        if ( udpOutgoingDataAssociationMap[...] is empty Then
            For each j : udpOutgoingDataAssociationMap[...]
                keyUdp = GetKey( j )
                if keyAsm equals keyUdp Then
                    For each l : GetValues( keyUdp )
                        If Not ValueAsmList[...] contains l Then
                            errorList[...] = l missing association from keyAsm
                        Endif
                    Next l
                Endif
            Next j
        else
            For each m : GetValues( keyUdp )
                errorList[...] = m missing association from keyAsm
            Next m
        endif
    Next i

    Return errorList[...]
End Function

```

Figure 5-7 Data Association Correctness Check Algorithm

Correctness Check

The correctness check aims to verify that the execution sequence of Activities in the UDP is correct. In this example, the correctness check fails because Task T1 is missing pre-conditions of Task T0 and Task T7 is missing pre-condition Task T6. However, T2 is still valid as long as the pre-condition is found before Task T2 and not necessarily found immediately. The Correctness Check algorithm is shown in Figure 5-8.

```

Function FlowNodeCorrectnessCheck(udpDef, asmDef)

    //udpDef: definition in a user-defined process for which the correctness check is applied to.
    //asmDef: definition of the selected adaptive standard model.

    udpFlowList [...] = GetFlowElements(udpDef); // flow nodes and data objects
    asmFlowList [...] = GetFlowElements(smDef); // flow nodes and data objects

    updSequenceFlow = getSequenceFlow( udpDef )
    asmSequenceFlow = getSequenceFlow( asmDef )

    udpPreconditionMap[...][...] = getAllPreconditionNodes(updSequenceFlow)
    asmPreconditionMap[...][...] = getImmediatePreconditionNodes(asmSequenceFlow)

    For each i : asmPreconditionList[...][...]
        keyAsm = GetKey( i ) ;
        For each j : udpPreconditionMap[...][...]
            keyUdp = GetKey( j ) ;
            if keyAsm equals keyUdp Then
                For each k : GetValues( keyAsm )
                    foundError = true
                    For each l : GetValues( keyUdp )
                        If k equals l Then
                            foundError = false
                        Endif
                    Next l
                Next k
            If foundError Then
                errorList[...] = node j missing precondition - node i
            Endif
        Endif
    Next j
Next i
Return errorList[...]
End Function

```

Figure 5-8 Correctness Check Algorithm

The Correctness Check algorithm captures pre-condition activities of ASM and UDP extracted from the SequenceFlow class into respective array maps. The ASM map

contains individual nodes as keys with all pre-condition nodes as values. The UDP map contains each node as keys with the immediate pre-condition node as value(s). If the UDP key matches the ASM key and if UDP value contains an ASM value then the Correctness Check is successful, otherwise an unavailable pre-condition value is stored into the error array list.

5.2.6 Dynamic Adaptation with Data-Driven Check

Dynamic adaptation is the technique of updating one specific process instance at run-time by means of business rule logic to accommodate dynamic events of compliance requirements.

Figure 5-9 shows an example of applying dynamic adaptation. In this example, the process dynamically updates the affected process instance through its enacted UDP.

Figure 5-10 illustrates the business rules definition that is defined in separate files. Rule R5 is automatically triggered when Business Rules Task T5 is reached during the execution of the process instance. Rule R5 will validate the condition and, if true, will perform the specified action. In this example it executes the *Dynamic Adaptation Module* to make changes on the enacted UDP to replace User Task T6 with T9. The process instance is updated with the latest modified UDP through a migration technique. The changes also need to be reflected back to the ASM to make sure it is valid as a reference model for the purpose of the structural compliance check.

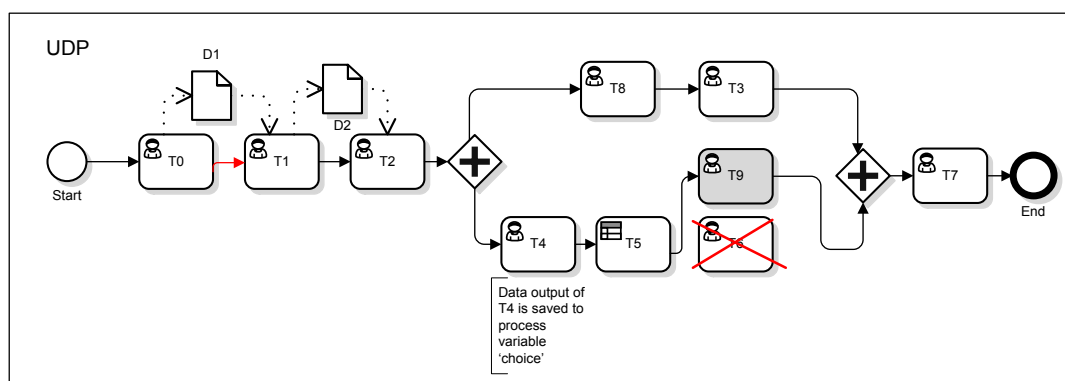


Figure 5-9 Dynamic Adaptation example showing UDP

```
Rules R5

when
    Process Variable 'choice' == 'Yes'
then
    DynamicAdaptation(UDP, replace, T6, T9)
end
```

**Figure 5-10 Business Rules to Execute
Dynamic Adaptation**

Data-Driven Check

The Data-Driven Check is used to monitor relevant workflow data during run-time by using the same business rules logic as above.

Figure 5-11 provides an example showing a business rules definition for business rule task T3 to validate the Process Variable 'choice' value. In this example, if D2 value is more than 10, the system is instructed to send an email to warn the user to take necessary actions or to resolve a potential violation.

```
Rules T3

when
    variables 'choice' < 10
then
    ExecuteEmailReminder();
end
```

Figure 5-11 Example of Business Rules Definition

5.3 Prototype Implementation

5.3.1 Introduction

In order to demonstrate the effectiveness of the ACWfMS architecture, this section discusses the implementation of the ACWfMS prototype. As discussed in Section 2.6, jBPM (2013) met the criteria that this Thesis put forward for the selection of an ACWfMS prototype development platform. However, it was identified that jBPM lacked supporting compliance and adaptive features as a workflow development tool.

jBPM is part of the JBoss community project which is a division of Red Hat. It is a lightweight extensible open-source workflow engine that executes business processes using the latest BPMN 2.0 specification. One of the main reasons for adopting jBPM is its ability for developers to develop new modules. Further, jBPM provides an existing API (Application Programming Interface) that can be used to interact with the knowledge base and internal classes.

jBPM project has community releases from JBoss (Redhat 2013) that come without support. ACWfMS prototype source code is available at the Github site¹, which is an independent variant (i.e. forked) from jBPM-Designer (jBPM 2013).

¹ <https://github.com/saifulomar/process-designer>

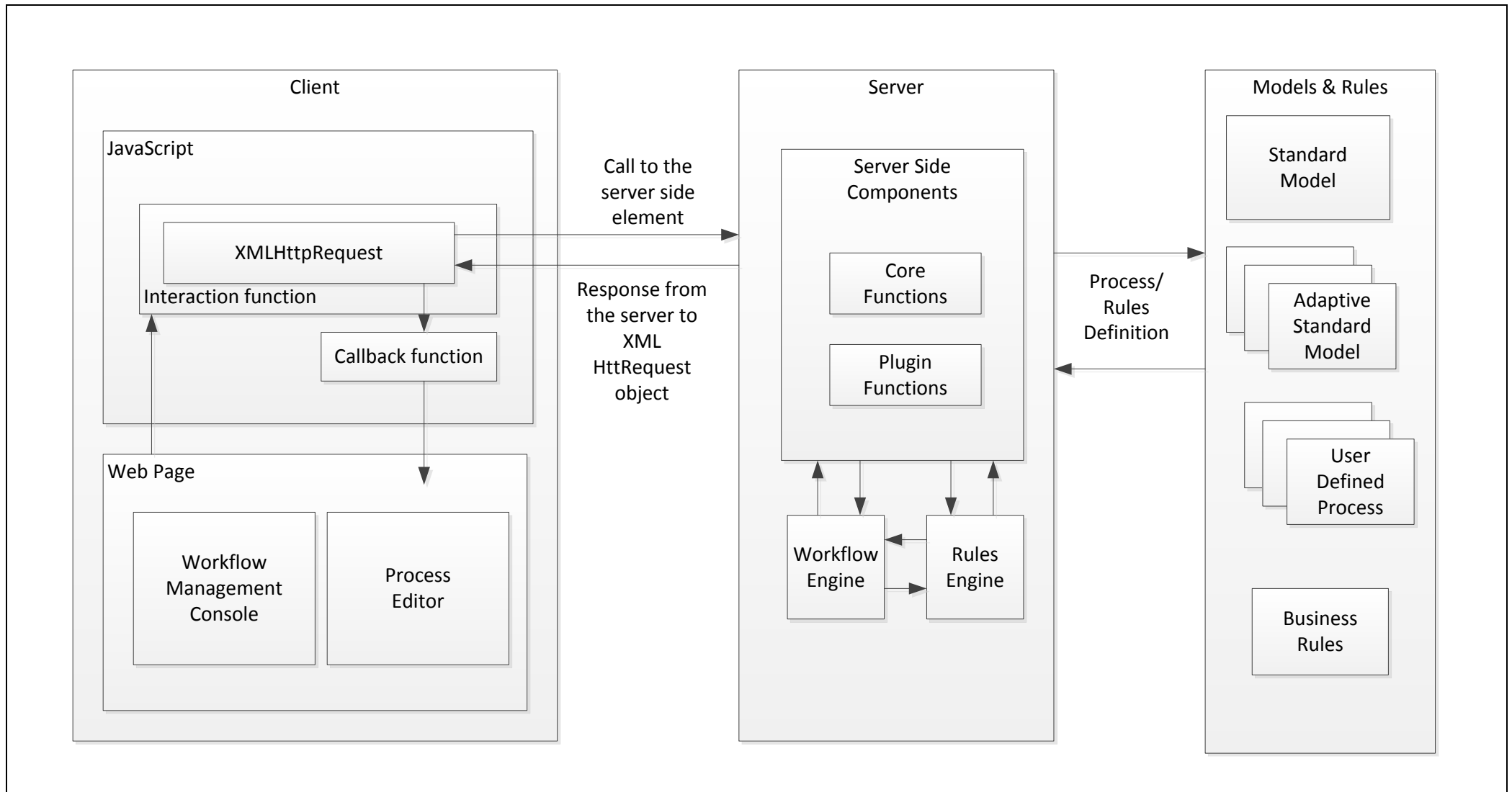


Figure 5-12 ACWfMS Technologies Foundation

The proposed prototype extends the jBPM Designer process editor, workflow management console and server side components by adding plugin modules in order to support adaptive and compliance capabilities. ACWfMS technologies foundation is shown in Figure 5-12. The diagram shows the logical services, data components and underlying technologies to supports ACWfMS.

5.3.2 Modelling Standard and Process

As indicated in Section 5.2.1, ACWfMS applies the reference model technique that separates the compliance requirement from the actual workflow process and introduced three-tier models: SM, ASM and UDP.

Since this study proposes BPMN as a modelling language and jBPM Designer supports BPMN, the ACWfMS prototype uses jBPM Designer as the process editor for modelling SM and UDP. By using jBPM Designer, the Administrator (Process Designer) is able to model organisational compliance requirements as SM.

After SM has been completely drafted and verified by the policy regulator (which checks that SM has been able to represent the compliance requirement), then it is ready to be utilised as a template for ASM and UDP.

From this point, ACWfMS distinguishes itself from other workflow tools. UDP will be used as the actual workflow processes to be executed by the workflow engine. The Actor (Participant) can make improvements to UDP both during build-time or run-time through user adaptation features and, at the same time, make sure changes comply with the requirements. ASM is introduced to reflect changes made during dynamic adaptation.

jBPM Designer is a web-based processes editing tool that can be used to create, view, or update BPMN based processes which are executable in the jBPM runtime environment. jBPM Designer is an independent variant (i.e. forked) of the Oryx project (Decker et al. 2008). The Designer interface is composed of a number of sections as shown in Figure 5-13.

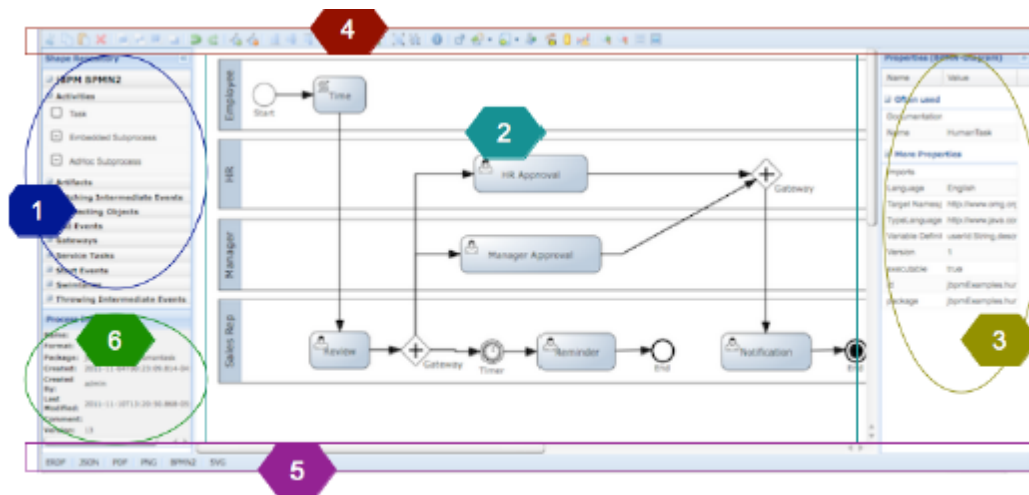


Figure 5-13 Designer User Interface (adapted from JBPM 2012)

Detail of each section are described below (JBPM 2012):

1. The Shape (node) Repository Panel section shows the available BPMN modelling elements that can be used to assemble SM and UDP. The shape can be placed onto the Designer Canvas (2) by dragging and dropping an element onto it.
2. The Canvas section is used as the process drawing board. After dropping different shapes onto the canvas, they can be organised and connected together. Selecting a shape permits one to set its properties in the Properties Window (3)
3. The Properties Panel section is used to set both process and shape properties. When a shape is selected in the Canvas, this section is reloaded to show properties specific to the shape type. If the canvas itself is selected the section shows the general process properties.
4. The Toolbar section contains operations which can be performed on shapes present on the Canvas.
5. The Footer section contains operations to view the source of the process.
6. The Process Information section contains information about the process: name, creation date, version, and others.

5.3.3 Firing Business Rules

As discussed in Section 5.2.3 the ACWfMS architecture applies business rules to support dynamic adaptation and data-driven compliance validation. The integrated solution of business process and business rules as a whole become more adaptable in that processes can be changed dynamically based on compliance requirements defined in a business rules statement. Further, an integrated engine for rules and process is necessary for complex behavioral modeling.

It is convenient for this implementation that jBPM is not just an isolated process engine. jBPM is integrated with a business rule engine supported by the Drools project (2013).

Drools uses Rete, a matching algorithm that was developed by Forgy (1982). Rete is an efficient method for comparing a large collection of patterns to a large collection of objects.

There are four categories of business rule statements (Hay & Healy 2000):

- **Definitions of business terms** - In a business rule the most basic element is the language used to express it. Term is used to describe how people think and talk about things and thus establish a category of business rule;
- **Facts relating terms to each other** – The nature of an organisation can be described in terms of the facts which relate terms to each other. For example, the fact that a student may book a course is a business rule. Facts can be documented as relationships or attributes. A fact is further classified as a base fact or a derived fact. A base fact is simply recorded as given data and remembered in the system. For example, “a student must have a student ID”. A derived fact is an assertion that is constructed from other assertions that may be a computed value or view (principles). For example, with respect to the PGR student progress - “Before a full-time student will be permitted to re-register for their second year of research, he/she will be required to submit evidence of their research progress to date and be assessed”;
- **Constraints** (also called ‘action assertions’) - Organisations impose constraints behaviours in some way that specifies the results that actions can

produce. An action assertion is evaluated using one or more constructs using “If Then ...” where the action would go after “then”. For example, “**If** year one progress review is satisfactory **Then** student progresses to year two”;

- **Derivations** – Business rules state how knowledge in one form may be transformed into other knowledge. Derivation is an algorithm used to compute or infer a derived fact. It must be used to derive at least one and possible more derived facts. For the PGR criteria for assessment the derivations of derived facts can be specified as “the registration for PhD is *inferred* from the research reports that should show evidence of a viable research programme including a plan for completion, a grasp of appropriate research methods, potential for publication, an element of originality and evidence of adequate progress including successful achievement of research targets”.

When using the rule engine there is no control as to when the rules will be fired (executed). Rules become available to fire and the rule engine decides the best order in which to execute them. The fine-grained controls of other languages are missing. According to Browne (2009) this feature is beneficial which makes the individual rules simpler, reusable, and easier to understand. Nevertheless, there are situations where business rules may need to group and control when to fire.

For example, the PGR process may have several business rules, such as: rules that fire for yearly progress review, rules that fire for validating training and supervisory meeting frequency. To overcome the control issue, Browne (2009) suggested using business *ruleflow* that maps process flow graphically to see the sequence of rule firing. It is easier to understand ruleflow in a diagram than deciphering the information buried in individual rules. However this does not mean that ruleflow is a workflow. In BPMN a ruleflow uses rule task elements. When a rule task is reached in the business process, it gives the rule a chance to fire and it is up to the rule engine to decide which rules are the most appropriate to fire. Rules statements are defined using the Drools rule format in a separate file. Rules can become part of a specific ruleflow group using the ruleflow-group attribute in the header of the rule. The integration of business process and business rules makes rules easier to

understand and control. Below is an example of a business process (Figure 5-14) and business rules (Figure 5-14) that cover the above ruleflow concept and business rule statement.

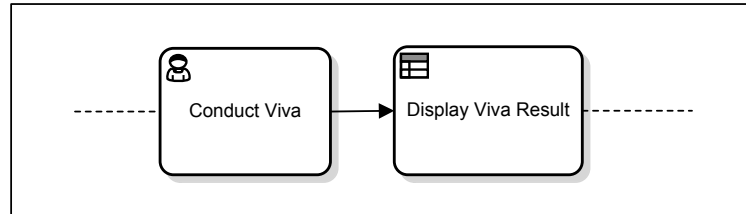


Figure 5-14 Business Process with Ruleflow

```

rule Pass Viva
ruleflow-group "viva"
  when
    variables['vivaResult'] == "passed"
  then
    System.out.println("Congratulation, you passed your
                        PhD viva");
End

rule Pass Viva with Minor Correction
ruleflow-group "viva"
  when
    variables['vivaResult'] == "passed with minor correction" //
  then
    System.out.println("Congratulation, you passed your
                        PhD viva with minor correction");
End
  
```

Figure 5-15 Business Rule Statements

5.3.4 Client and Server Plugin Modules

Section 5.2.4 discussed modules that are needed to support compliance checking and adaptive process functionality. These modules are implemented by extending existing jBPM tools: the Designer process editor and Workflow Management Console. A new service dynamic adaptation module is introduced here to support process change.

Designer

Designer consists of two architectural components - the client and the server side. The integration framework of Designer and ACWfMS is implemented to extract or change process models (SM, ASM and UDP) from the knowledge base repository and track process instances through execution logs. The client side plugins are used to extend the functionality of the editor. Two new modules are added to the editor: *Structural Compliance Checker Module* and *Instance Tracker Module*. With the extended plugins, the editor can be used to ensure compliance during user adaptation - discussed in the following sections. Guidelines on how to develop a plugin can be referenced at the Oryx Project Website².

Console

jBPM Workflow Management Console is used to manage process instances (starting/stopping/inspecting), inspecting the task list and executing those tasks. The ACWfMS prototype extends the Console in order to manage instance migration. When a process gets updated, jBPM will proceed with the running process instance as normal, followed by the process (definition) as it was defined when the process was started. This strategy causes the running process instance to proceed as if the process was never updated. With the addition of this functionality, the process instance is migrated to the updated process definition after committed user adaptation and continues executing based on the updated process logic.

The process instance contains runtime information that includes data (variables) that are linked to the process and the current state of the process. The runtime state is linked to a particular process with unique id references that represent the process logic to be followed when executing the process instance. Therefore, the implementation for process migration of a running process instance is achieved by changing the reference process id to the new id. The *Instance Migration Module* is implemented with the jBPM WorkflowProcessInstanceUpgrader API that upgrades a workflow process instance to a newer process instance.

² <https://code.google.com/p/oryx-editor/>

Dynamic Adaptation Services

In order to support the dynamic adaptation concept (Section 5.2.6), the ACWfMS prototype implements the *Dynamic Adaption Module* as a service class. This class is accessible as a Java API. Its function is to update the UDP definition (adding or deleting) activities and, in turn, migrate effected process instances with updated UDP processes and also to propagate the changes to the ASM.

5.3.5 Implementation of User Adaptation with Structural Check

User adaptation actions can be applied either during build-time or run-time. To execute user adaptations, the *Process Editor* retrieves UDP and the affected instance states by converting the process definition into a graphical business process format. UDP is retrieved from the knowledge base repository. The instance activity state is retrieved from the *Execution log* through the *Instance Tracker Module*.

However, an interesting question arises during the implementation of this prototype: how to explain structural compliance violations to the user? Based on the proposed algorithm that was discussed in Section 5.2.4, the ACWfMS prototype approaches this by using the error list generated from the structural compliance checkers. The error list contains missing activities and references of missing pre-condition activities that are feed to the process editor which visually displays non-compliance activities.

To initiate compliance checking, either during build-time or run-time, the client side plugin is triggered via a toolbar button within the process editor. For compliance checking, the user has to select the Process Compliance Check toolbar button. After pressing the toolbar button, the query is serialized and passed to the server-side plugin. The query servlet instantiates the *Structural Compliance Checker Module* that checks for completeness and correctness and returns an available error list.

Figure 5-16 shows a screen shot of the implemented ACWfMS prototype process editor displaying a non-compliant UDP with instance tracker during run-time.

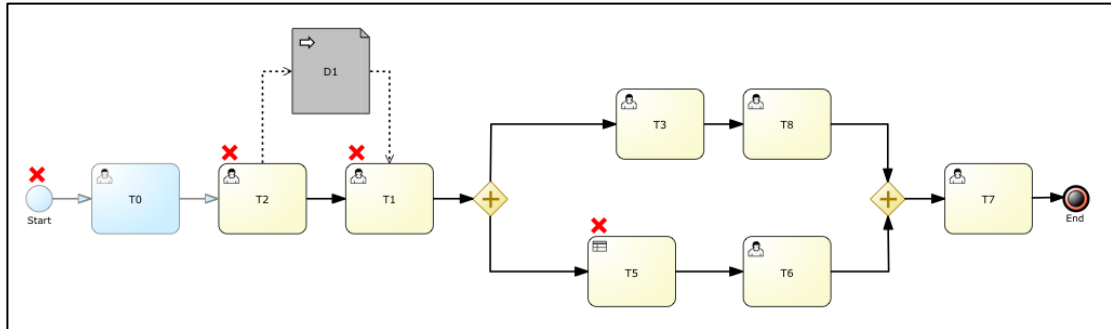


Figure 5-16 Screen shot of ACWfMS - Visual Compliance Check with Instance Tracker

T0 is coloured blue to indicate the nodes are locked from the user making further changes. This lock is activated because of the runtime state of the node that is either activated or completed.

The editor presents red “X” marks next to process nodes that contain compliance violations. Pointing the mouse-over each red “X” presents the violation error description, as follows:

- Start – Missing: Flow Node: T4, Missing: Data Object D2
- T2 – Missing Pre-Condition node: T1, Missing Association from D1
- T1 – Missing Association to D1
- T5 – Missing Pre-condition node: T4

The visual compliance check helps the user to understand the compliance violation affecting unique user defined processes and instances. Automating compliance checks saves the effort of manually identifying such violations. The prototype assumes the activities have unique labels. Names of the same node should be the same through all models and processes.

5.3.6 Implementation of Dynamic Adaptation with Data-Driven Check

Dynamic adaptation is applied during run-time. It is implemented based on the integration of the business process and business rules. Business rules statements are defined according to policy requirements. When a business rule task is reached in the process, it executes the Business Rules Engine to evaluate corresponding rules.

In the event when corresponding rule conditions are true, it triggers the Dynamic Adaptation Module to update affected process instances as required e.g. adding or deleting an activity. Simultaneously, the Dynamic Adaptation Module propagates the changes back to ASM. The evolution properties for ASM ensure the concept of the reference model is continuously valid for compliance checking, in which a process instance is checked against ASM every time a user adaptation is made.

It is not sufficient to enforce all compliance checks at the structural level. Thus, it requires the Compliance Checker: Data checks for run-time monitoring of business process execution. ACWfMS applies the business rules technique to monitor relevant data during the process execution. The target is to identify potential violations as early as possible in order to allow strategies to resolve compliance violation. It validates business rules conditions against instance data. If the condition is true, it triggers an event (e.g. escalation, error) or service task (e.g. email reminder) to take necessary actions or to resolve potential violations.

5.4 Conclusion

This chapter has discussed the specification of system architecture and ACWfMS prototype. The system architecture is designed to cope with any generic business processes that are required by an organisation that needs creativity and flexibility to achieve organisational goals with the ability to ensure any instances comply with the domain-specific process requirements.

Process requirements are translated into a Standard Model (SM) that separates the compliance requirements from the actual workflow process that is represented as a User Defined Process (UDP). A two-tier model is insufficient to maintain and ensure the concept of a reference model is valid throughout the adaptive process. Thus, another model is introduced between SM and UDP i.e. the Adaptive Standard Model

(ASM). This is because the reference model will need to be changed for each instance depending what library sub-processes have been added.

The ASM presented here provides a way to integrate adaptive and compliance workflow solutions in handling individual instance exceptions on rare events and supports evolving process instances through user or dynamic (automatic) process adaptation. Moreover, the proposed concepts ensure that running process instances comply with process requirements before instance migration takes place. Finally, the proposed concept is a preventive approach that prevents actual execution of non-compliant processes in an adaptive workflow solution.

A prototype was implemented as a proof-of-concept on integrating an adaptive and compliance workflow development tool. In particular it demonstrated how the three-tier models used sub-processes to modify an instance during runtime, while at the same time enabling automatic compliance checking. Real world case studies on the Postgraduate Graduate Research (PGR) domain using ACWfMS as a development tool are discussed in the following chapter.

Chapter 6

Case Study

6.1 Introduction

The aim of this chapter is to evaluate the proposed system architecture through the Adaptive and Compliance Workflow Management System (ACWfMS) prototype that implements the key components of the architecture, specifically on the adaptation and compliance standards for dynamic workflows. Section 6.2 introduces the evaluation criteria for the ACWfMS prototype. Section 6.3 presents the case study domain. Section 6.4 discusses the scenarios for the purpose of collecting data and evidence. The scenarios are set up to make comparison between generic WfMS with ACWfMS on supporting process adaptation and compliance standard. Section 6.5 concludes the evaluation that determines the effectiveness of ACWfMS against the required system requirements.

6.2 Evaluation Criteria

Before any evaluation task takes place, an evaluation criteria needs to be set up, which includes the challenges and requirements set out earlier as described in Chapter 3. These criteria cover both the integrated solutions for compliance and adaptive techniques that handle specific dynamic instances within a WfMS.

Recall the key components of the proposed system architecture for this evaluation was introduced in Chapter 5. The adaptive technique should aim at reconciling the two main concerns of process change: ad-hoc changes and evolutionary changes. Ad-hoc changes are supported with a user-adaptation technique that handles rare events. Evolutionary changes are supported with a dynamic adaptation technique that accommodates dynamic events that are based on compliance requirements.

The proposed three-tier model is used to support compliance validation (SM, ASM and UDP). The compliance technique should have the ability to support the dynamic

reference model that supports specific instances in an adaptive workflow environment.

The following are the criterion **used** to evaluate the ACWfMS prototype covering all aspects of ad-hoc and evolutionary changes to a specific instance process:

- Criterion 1:** Use BPMN process for modelling and execution;
- Criterion 2:** Capture compliance requirements with business process;
- Criterion 3:** Capture compliance requirements with business rules;
- Criterion 4:** Perform process change with user adaptation and dynamic adaptation;
- Criterion 5:** Propagate adaptive (non-Static) standard model;
- Criterion 6:** Check compliance at structural level;
- Criterion 7:** Check compliance at data level;
- Criterion 8:** Perform instance tracking; and
- Criterion 9:** Perform instance migration.

Overall, ACWfMS should be able to support dynamic changes on UDP that are based on events by means of business rules. Change should be propagated back to ASM to support individual instances to have a unique non-static reference model for compliance checking. The user should also be able to change the process based on ad-hoc requirements. All ad-hoc changes should be verified for compliance against ASM. It follows with specific instance migration that is based on updated logic and copes with dynamic changes during run-time.

A set of evaluation criteria is summarised in **Table 6-1** together with the scenarios and the targeted aspect to be used for testing.

Table 6-1 Testing Plan

Criteria	Scenario	Targeted Aspect
C1. Use BPMN process for modelling and execution	To be observed in Section 6.3	-
C2. Capture compliance requirements with business process	To be observed in Section 6.3	-
C3. Capture compliance requirements with business rules	To be observed in Section 6.3	-
C4. Perform process change with user adaptation and dynamic adaptation	<p>S1: Apply for Conference Fund with User Adaptation with Completeness Check.</p> <p>S2: Redo Progress Review with Dynamic Adaptation.</p>	<ul style="list-style-type: none"> • Add ad-hoc activities on process instance through user adaptation (S1) • Dynamically change process instance to accommodate compliance requirement by means of business rules (S2)
C5. Evolution of adaptive (non-static) standard model	S3. Progress to Fourth Year with Propagating ASM and UDP.	Propagate dynamic change on process instance back to UDP and ASM for continuity of compliance check validity
C6. Check compliance at structural level	S1: Apply for Conference Fund with User Adaptation with Completeness Check.	Validate UDP against ASM for completeness and correctness check
C7. Check compliance at data level	S4. Meeting and Training frequency check	Examine instance against business rules with data check

C8. Perform instance tracking	S5. Apply leave of absence	Track nodes within UDP that have been executed in the process instance and signals process editor to lock those executed nodes from any further changes
C9. Perform instance migration	S5. Apply leave of absence	Allow change process to continue based on updated logic and copes with dynamic changes during run-time

The criteria C1 to C3 can be observed in the following section where the BPMN standard is applied with graphical representation for specifying compliance requirements and validations at the processes structure level. Business rules represent compliance requirements and validation at tasks and data level.

6.3 Case Study Domain: Postgraduate Research Process

As discussed in Section 4.3, the Postgraduate Research (PGR) process was chosen as the case study domain. The PGR process provides relevant evaluation scenarios to validate the proposed ACWfMS architecture and evaluate the prototype as a proof of concept. The process is derived from the Code of Practice on Research Degree Programmes, Loughborough University (2012) for full-time research students. The SM of PGR core and sub-processes are described as follows:

a) SM for PGR Core Process

The normal period of study for a PhD programme is three years full-time. Within these time scales, students are expected to complete their research and write up their Thesis unless they are granted an extension to their studies for completion of their Thesis. This normal period of study is reflected in the Standard Model (SM) PGR Core Process, which starts with a Student Registration task and a sequence of yearly sub-processes, as shown in Figure 6-1.

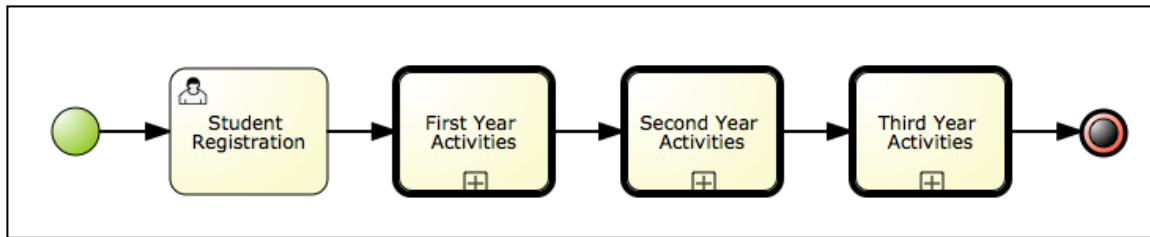


Figure 6-1 PGR Standard Model: Core Process

b) SM for First/Second Year Activities Sub-Process

The minimum frequency of formal meetings between the students and the supervisor(s) will be 12 meetings per annum. Additional meetings can be added as needed. Students are required to undertake research training and maintain a record of their training. Students are expected to re-register annually according to the anniversary of their initial registration, subject to a satisfactory progress review. These requirements are translated into the SM PGR First/Second Year sub-process as shown in Figure 6-2.

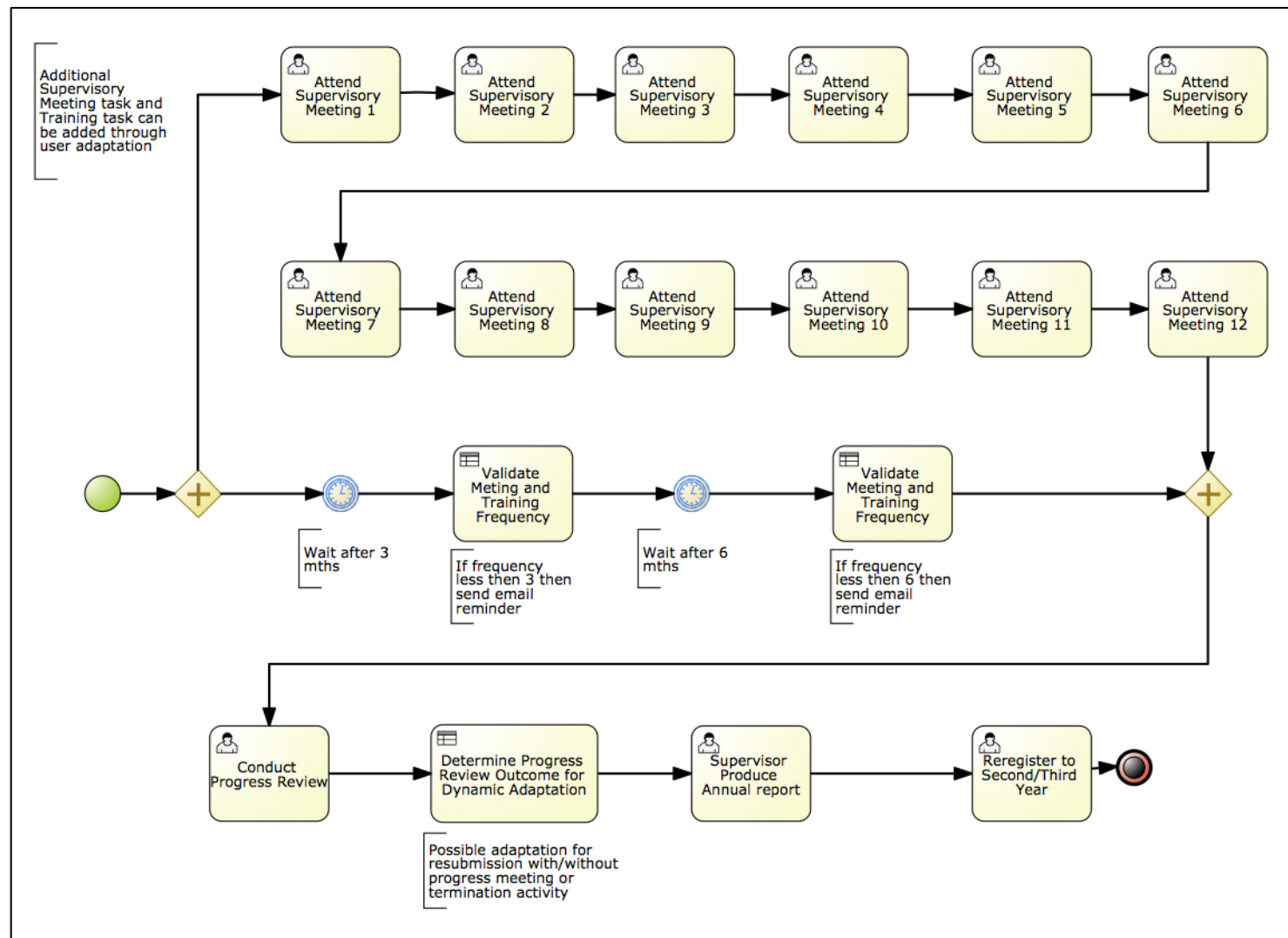


Figure 6-2 PGR Standard Model: First and Second Year Sub-Process

The business rules associated with this model are as follows.

- If progress review is satisfactory then continue to 'Second Year Activities' or 'Third Year Activities';
- If progress review is rewrite and resubmit then add 'Rewrite and Resubmit progress review' activity after 'Conduct Progress Review' activity;
- If recommendation of progress review is rewrite and resubmit with oral examination then add 'Rewrite and Resubmit with oral examination' activity after 'Conduct Progress Review' activity;
- If recommendation of progress review is unsatisfactory then add 'Terminate Registration' activity before reregister to year two/three task;
- If the student wishes to appeal against termination of registration then add 'Appeal Against Termination' activity before reregister to 'Second Year Activities' or 'Third Year Activities';
- If appeal against termination is approved then continue to reregister to 'Second Year Activities' or 'Third Year Activities';
- If appeal against termination is rejected then remove 'Reregister to Second Year or Third Year' task, 'Second or Third Year Activities' and end process;
- If meeting or training frequency is less than three after the third months of yearly registration then send email reminder; and
- If meeting or training frequency is less than six after the sixth months of yearly registration then send email reminder;

c) SM for Third and Extension Period Activities Sub-Process

PhD students are expected to complete their research and Thesis write-up within three years. Thesis submission ideally will take place during their third year. In case a student is unable to complete their research and Thesis write-up during the third year of their study and, subject to approval, an extension period is granted for no more than 12 months. The SM PGR Third and Extension Year Activities sub-process is shown in Figure 6-3 and follows the business rules associated with this model.

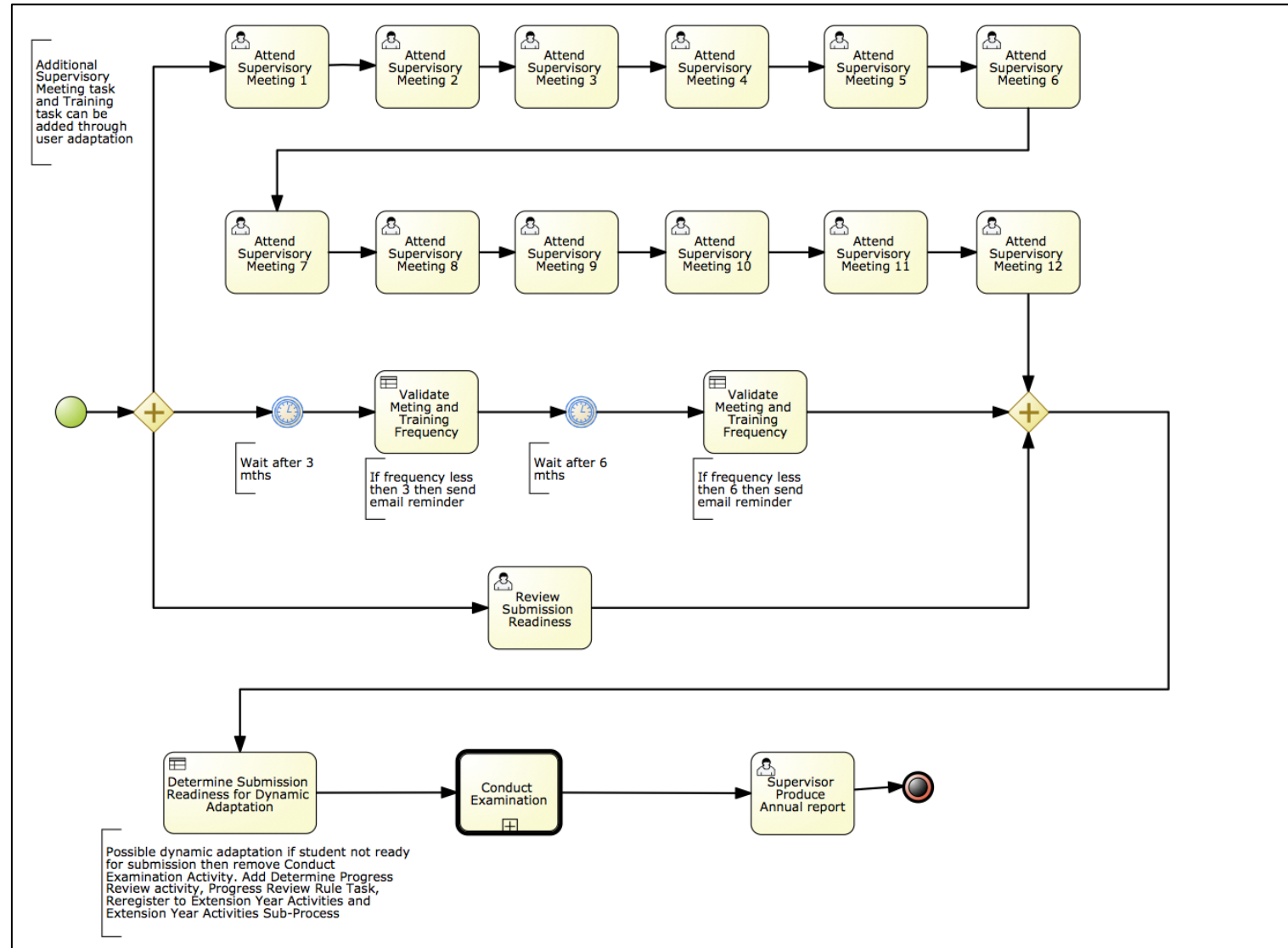


Figure 6-3 PGR Standard Model: Third and Extension Year Sub-Process

The business rules associated with this model are as follows.

- If recommendation of submission review is ready to submit then continue to examination activity;
- If recommendation of submission review is not ready and the student decides not to extend registration then add 'Terminate Registration' activity and end process;
- If recommendation of submission review is not ready and the student decides to extend registration then add 'Apply to Extend Registration' activity;
- If extension is granted then remove 'Conduct Examination' activity. Add 'Determine Progress Review' activity, 'Progress Review' rule task, 'Reregister to Extension Year' activity. And add 'Extension Year Activities' Sub-Process in the PGR Core Process;
- If recommendation of progress review is satisfactory then continue to extension year activities sub-process;
- If progress review is rewrite and resubmit then add 'Rewrite and Resubmit Progress Review' activity after 'Conduct Progress Review' activity;
- If progress review is rewrite and resubmit with oral examination then add 'Rewrite and Resubmit with Oral Examination' activity after conduct progress review activity;
- If progress review is unsatisfactory then add 'Terminate Registration' activity before 'Reregister to Extension Year' task;
- If the student wishes to appeal against termination of registration then add appeal activity before reregister to extension year task;
- If recommendation of appeal against termination is approved then continue with reregister to extension year task;
- If recommendation of appeal against termination is rejected then remove 'Reregister to Extension Year' task and end process;
- If meeting or training frequency is less than three after the third months of yearly registration then send email reminder; and

- If meeting or training frequency is less than six after the sixth months of yearly registration then send email reminder;

d) SM for Conduct Examination Sub-Process

The PGR standard model sub-process for Conduct Examination activities is shown in Figure 6-4 and is followed by the business rules associated with this model.

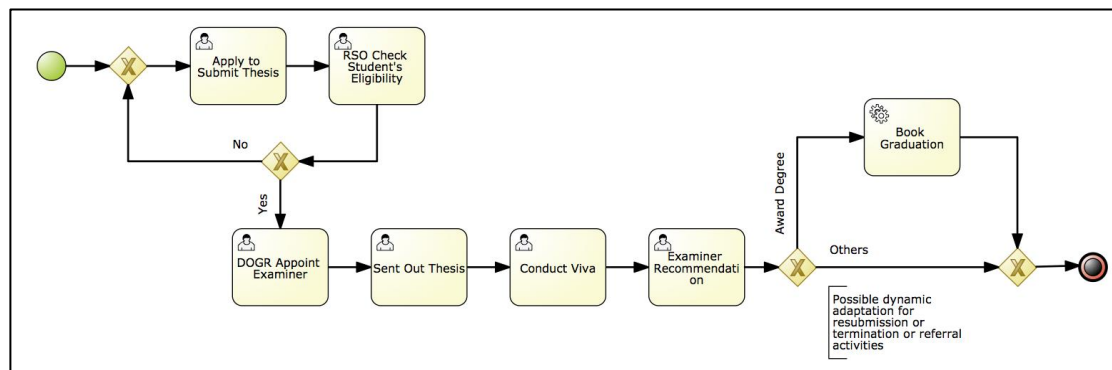


Figure 6-4 PGR Standards Model: Conduct Examination Sub-Process

Business Rules for Conduct Examination activities as follows:

- If examiners recommend that the degree be awarded then add 'Book PhD Graduation' activities and end process;
- If examiners recommend that the Thesis requires correction then add 'Resubmit Thesis Correction' task;
- If examiners recommend that the Thesis is inadequate then add 'Referral Year' sub-process in the PGR Core Process; and
- If examiners recommend that the candidate be failed but offered MPhil then 'Book MPhil Graduation' activities and end process.

e) SM for Request for Leave of Absence Sub-Process

A student may request a leave of absence for personal, health or family reasons during a period of full registration. The SM PGR for Request for Leave of Absence sub-process is shown in Figure 6-5.

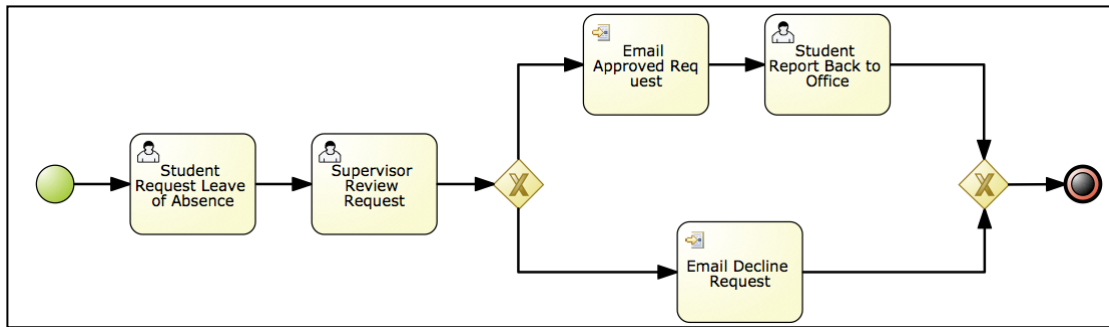


Figure 6-5 SM PGR: Request for Leave of Absence Sub-Process

f) SM for Apply for Conference Fund Sub-Process

The SM PGR for Attend Conference sub-process is shown in Figure 6-6.

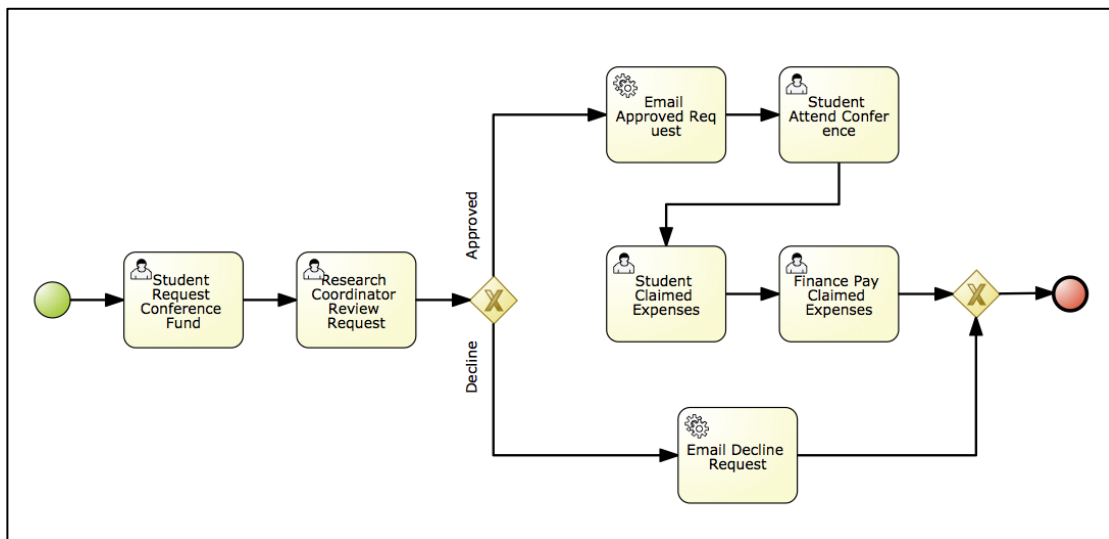


Figure 6-6 SM PGR: Attend Conference Sub-Process

In conclusion it is possible to claim that based on the set of SM above: the technique of the reference model is an efficient way to capture compliance requirements with business process [Criteria C2] and business rules [Criteria C3]; and the BPMN process language is able to represent the workflow process by means of graphical representation [Criteria C1].

6.4 Scenario-Based Testing

As discussed in Section 4.4, a scenario-based approach is used to collect data for the purpose of evaluation on the proposed system architecture through the ACWfMS prototype. The following sections present fictitious scenarios by comparing the generic WfMS approach with ACWfMS development tool to support the PGR process

through user and dynamic adaptation and making sure adapted processes are compliant with policy requirements. The scenarios make use of the SM that were defined in Section 6.3 and follow the plan outlined in **Table 6-1**.

6.4.1 Scenario 1: Apply for Conference Fund with User Adaptation with Completeness Check

This scenario is set-up to evaluate C4 on performing process change with user adaptation and C6 on checking compliance at structural level. In the PGR process, some activities that are known and anticipated can be defined early in sequence of tasks but some tasks are demanded on an ad-hoc basis such as applying for leave of absence, attending a conference, applying for an extension, transferring between universities, terminating studies or appealing against termination. In this scenario, a PhD student is applying for conference funds through their department. This evaluation compares the technique for adding ad-hoc activities of BPMN ad-hoc sub-process against ACWfMS user adaptation.

BPMN provides flexibility with ad-hoc sub-processes. Although a large part of the process is still well structured within the ad-hoc sub-process, users are the one to decide which task should be performed. The user has the ability to add new tasks during that period, which were not defined as part of the process, or repeat tasks with multiple times.

The BPMN approach in providing flexibility with an ad-hoc sub-process may create unstructured business processes or may be based on informal personal communications that result a logistical chaos. As shown in Figure 6-7, the process shows that a student may execute 'Claim Conference Expenses' task before 'Apply Conference Fund'. Further, the 'Finance Pay Conference Expenses' task might also be executed before a request approval is granted from the department. The order of the sequence of the tasks executed does not comply with the requirement's standards. The unstructured process causes confusion that can be unclear as to who is responsible for what and there can be a complete disorder of the tasks executed. Process success or failure depends more on user choice than on a clearly defined process. This approach can be disjointed, that is impossible to monitor a case

progress. Unstructured processes are nearly impossible to optimise - any improvements to the ad-hoc process are based on random judgement.

As discussed in Section 5.2.5, ACWfMS proposed a User Adaptation technique to support flexible processes for ad-hoc activities. For this scenario, the SM for 'Apply Conference Funding Sub-Process', as shown in Figure 6-6, is used as an initial template for ASM and UDP.

The user adaptation is done through a process editor during runtime (process already started) by adding the 'Apply Conference Funding Sub-Process' in between 'Attend Supervisory Meeting 1' and 'Attend Supervisory Meeting 2' tasks in the UDP.

At this point, the user attempts to adapt UDP for 'Apply Conference Funding Sub-Process' to fit their own requirements. The following are the changes made:

- Change existing user task from 'Research Coordinator Review Request' to 'Supervisor Review Request';
- Add an additional user task for 'Student Make Conference Booking' after task 'Email Approved Request'; and
- Change the sequence of 'Student Claim Expenses' to before 'Student Attend Conference'.

After changes take place, the *Structural Compliance Check* is triggered for completeness and correctness check. In this scenario the compliance check results failed as shown in Figure 6-8 with a red 'X' on top of the tasks which affect the whole process. The visual feedback will assist the user in identifying non-compliance activities with ease and make necessary corrections. Only after the process conforms to SM (compliance requirements) it is allowed to proceed with instance migration.

This test confirmed with the targeted aspect of C4 on performing process change with user adaptation and C6 on checking compliance at a structural level of the test plan. Compared with a generic BPMN ad-hoc sub-process, ACWfMS managed to support ad-hoc process change in a structured manner. Further, the compliance check made sure the tasks executed complied with the requirement standards.

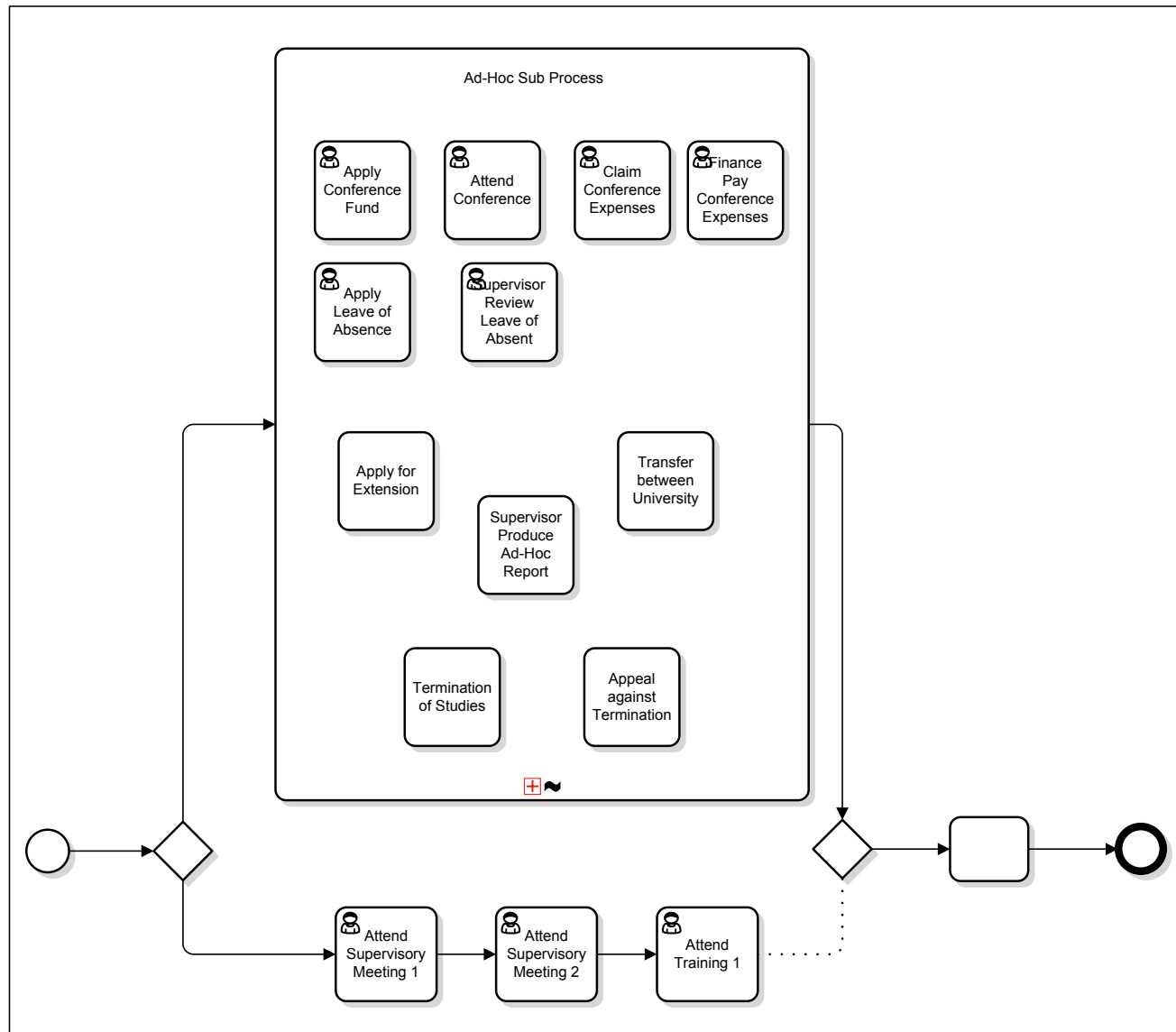


Figure 6-7
Process
Flexibility
with BPMN
Ad-Hoc
Sub-
Process

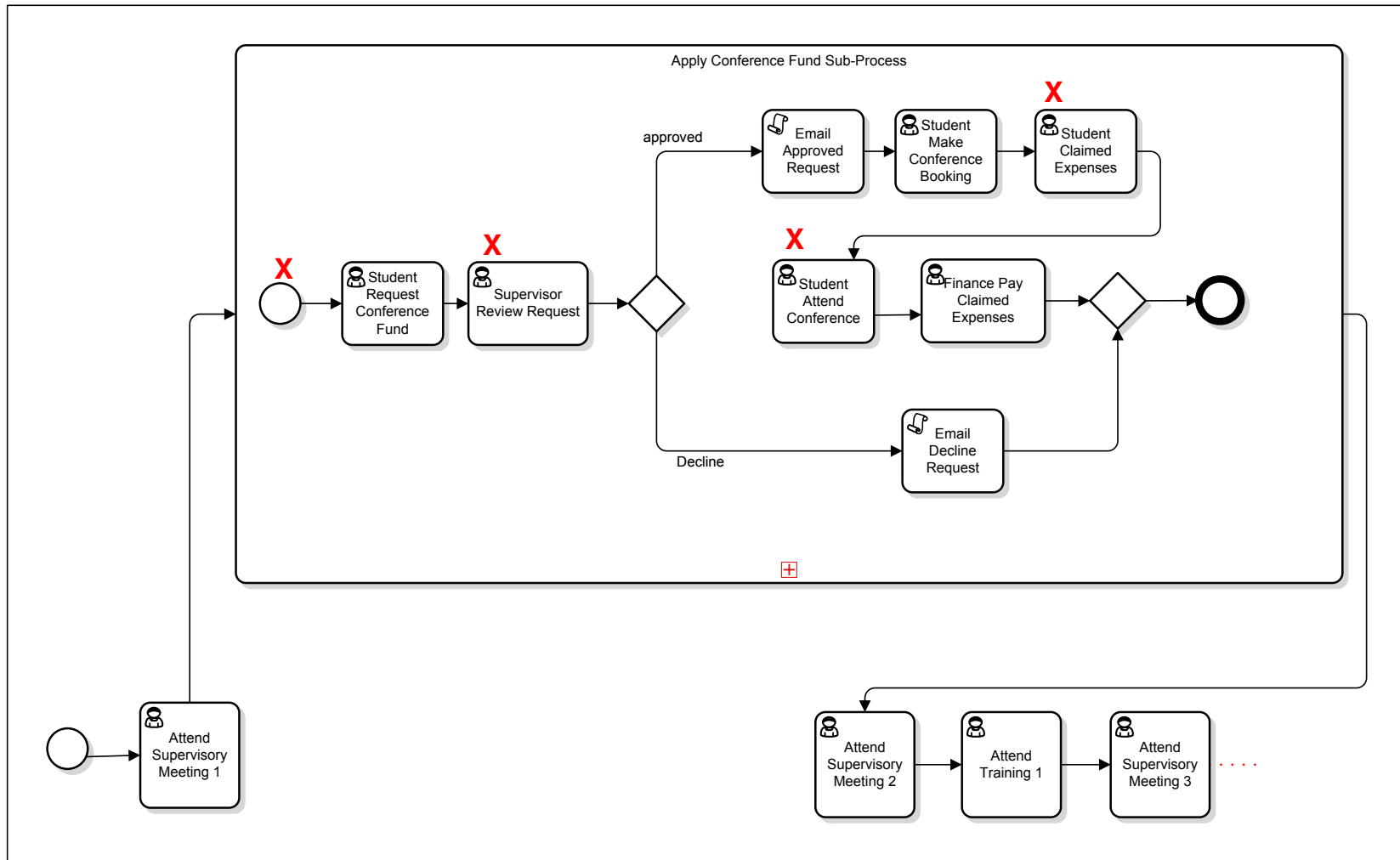


Figure 6-8
User
Adaptation
with
Compliance
Checks

6.4.2 Scenario 2: Redo Progress Review with Dynamic Adaptation

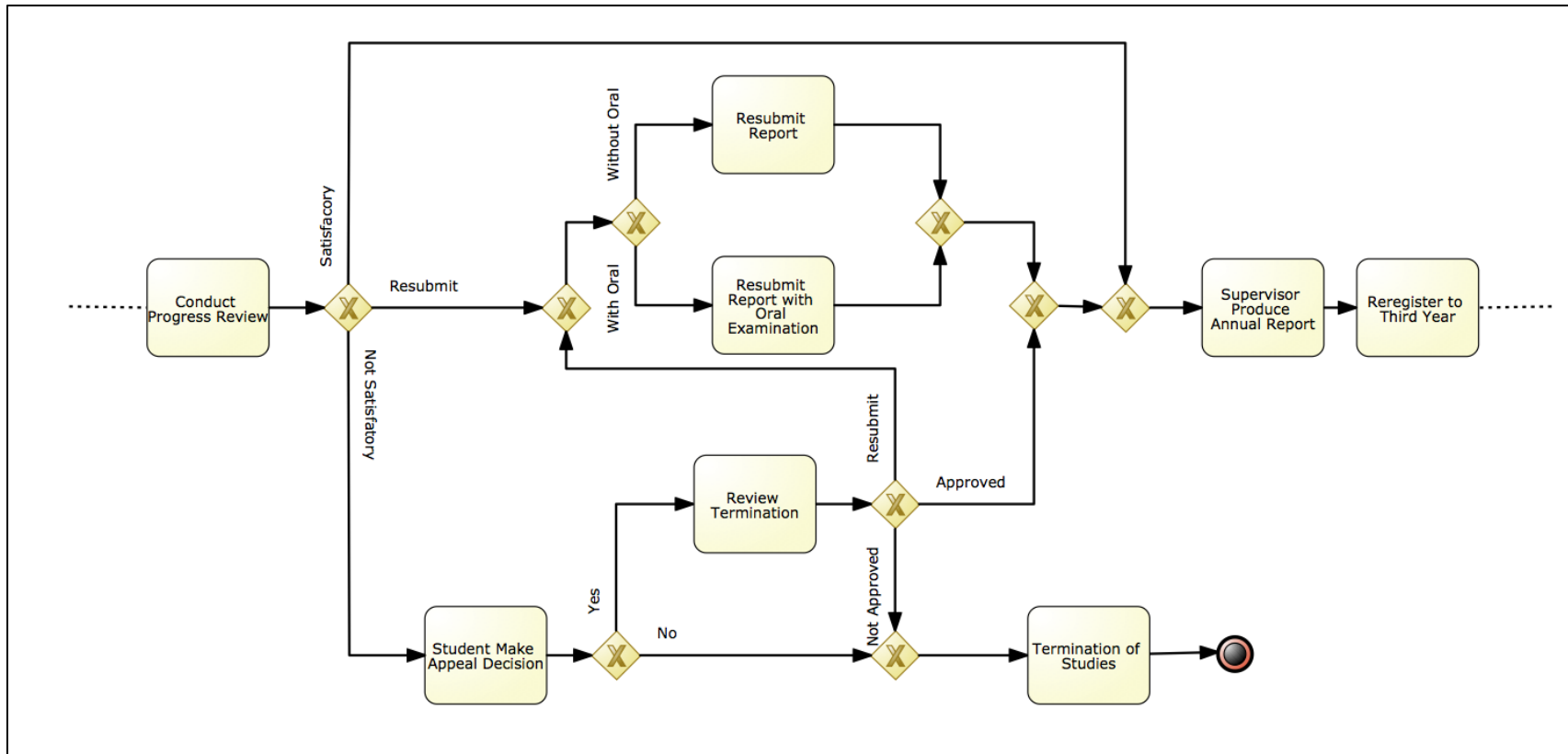
This scenario is set-up to evaluate C4 on performing process change with dynamic adaptation. This scenario is based on student progress where the examiner makes a recommendation to review and resubmit the second year progress report. The SM can be referred to in Section 6.3 (b). As stated in Section 3.2.2, the two main techniques for supporting flexible and adaptive workflow are by selection and adaptation. This evaluation compares the selection and adaptation techniques to change the process instance to accommodate compliance requirement.

Selection technique through events can be applied in a generic WfMS that utilises the BPMN process as shown in Figure 6-9. This process produces a considerable number of logical gateways that produces complex and hard-to-maintain business processes as discussed in Section 5.2.3.

By using the adaptation technique, the process starts with the belief that everything should be allowed, unless explicitly prohibited. The sequence of activities does not need to be determined early, which results in a more flexible process. Dynamic adaptation is performed at the instance level. ACWfMS supports adaptation by means of business rules that instruct the dynamic adaptation module to update the required instance and automatically propagates the changes back to ASM and UDP.

Based on the compliance requirements that are translated into the business rules condition, 'Resubmit Report' task is added dynamically into the process instance. The state of process instance before and after dynamic adaptation is reflected in Figure 6-10.

This test confirmed with the targeted aspect of C4 that ACWfMS supports process change with dynamic adaptation that simplified inflexible and complex process with the integrating of business process and business rules.



**Figure 6-9 BPMN
flexibility support
with selection
technique**

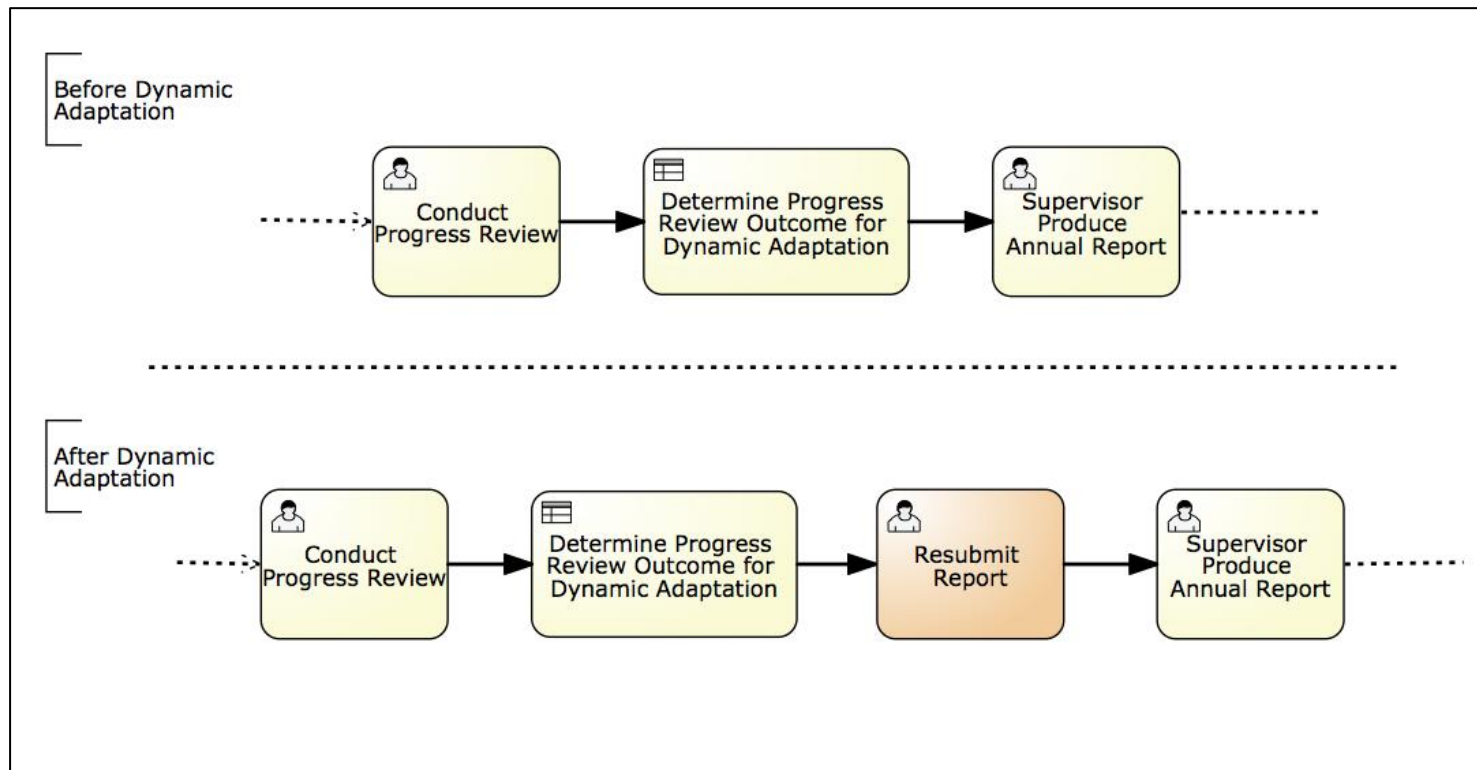


Figure 6-10 Dynamic adaptation with business rules.

6.4.3 Scenario 3. Progress to Extension Year with Adaptive Standard Model.

This scenario is used to evaluate the adaptive (non-static) standard model (ASM) technique that is based on the proposed three-tier model, as discussed in Section 5.2.2. In this scenario, a PhD student could not complete his/her write-up within the three years of full-time registration. However, an extension year is granted to complete his/her Thesis. The SM for PGR Core Process can be seen in Section 6.3 (a) and SM for Third Year Activities Sub-Process in Section 6.4 (c).

This evaluation compares the proposed technique of the three-tier against the two-tier model that was discussed in Section 3.3.5. The two-tier model comprises of SM and UDP layers. The two-tier model is suitable for representing a collection of related process instances that may have to be adapted. However, this test is concerned on specific instance change.

The execution of the Third Year Activities based on this scenario, is triggered by the Business Rules to dynamically adapt the specific instance by adding a 'Conduct Progress Review' task, a 'Determine Progress Review' rule task and a 'Reregister to Extension Year' task. Furthermore, the PGR Core Process instance is also added with the 'Extension Year Activities' sub-Process. To make the adaptation persistent, changes are propagated back to UDP. These changes should also be reflected in the SM for the purpose of compliance checking and consistency. However, with the two-tier model, changes to SM would cause other running instances to be non-compliant. The test result shows the two-tier model to be inconsistent after dynamic adaptation for the specific instance as shown in Figure 6-11. These missing activities in the SM will cause the completeness check to be invalid.

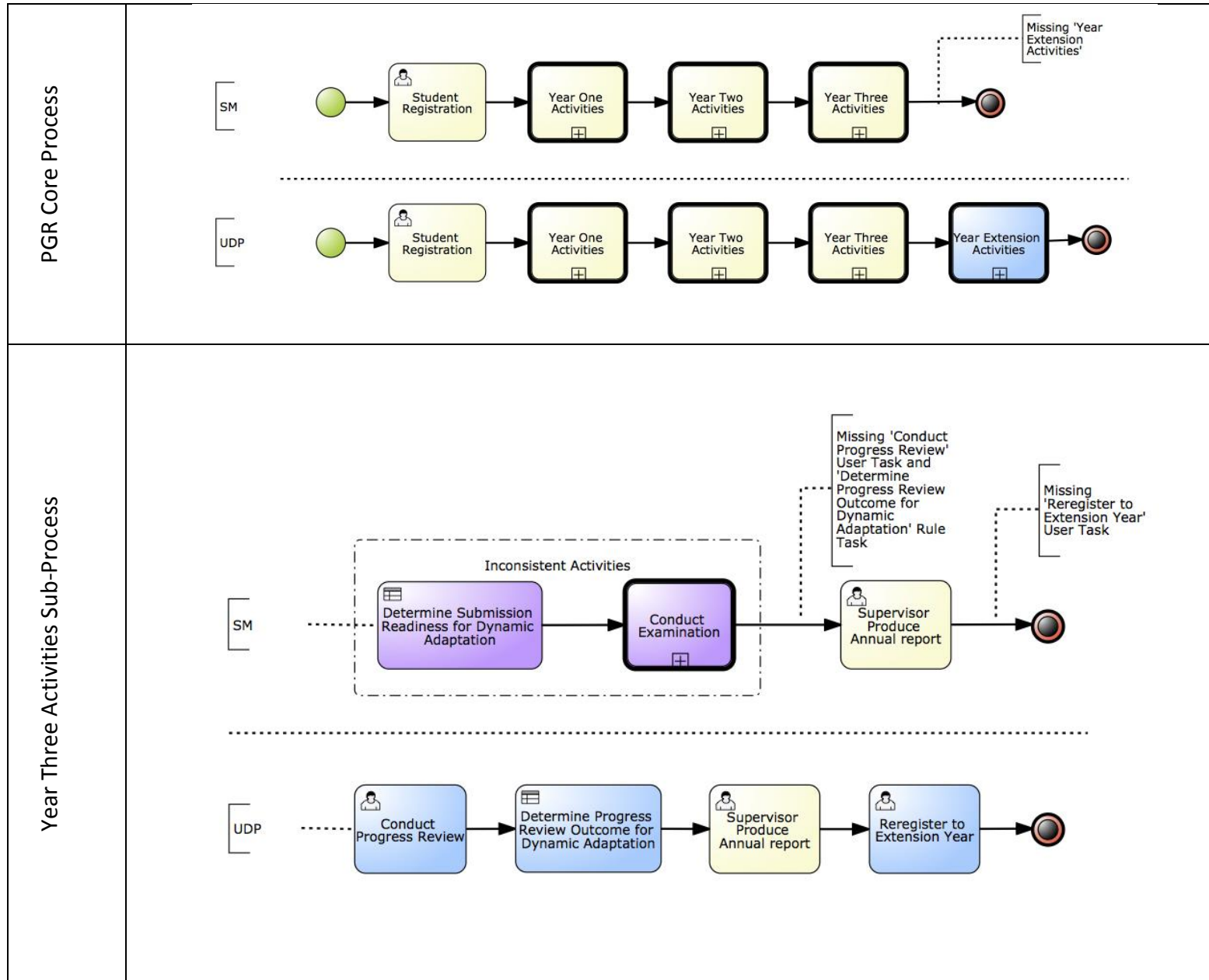


Figure 6-11 Inconsistent Two-Tier Model after Specific Instance Dynamic Adaptation

(Inserted tasks are indicated in blue colour)

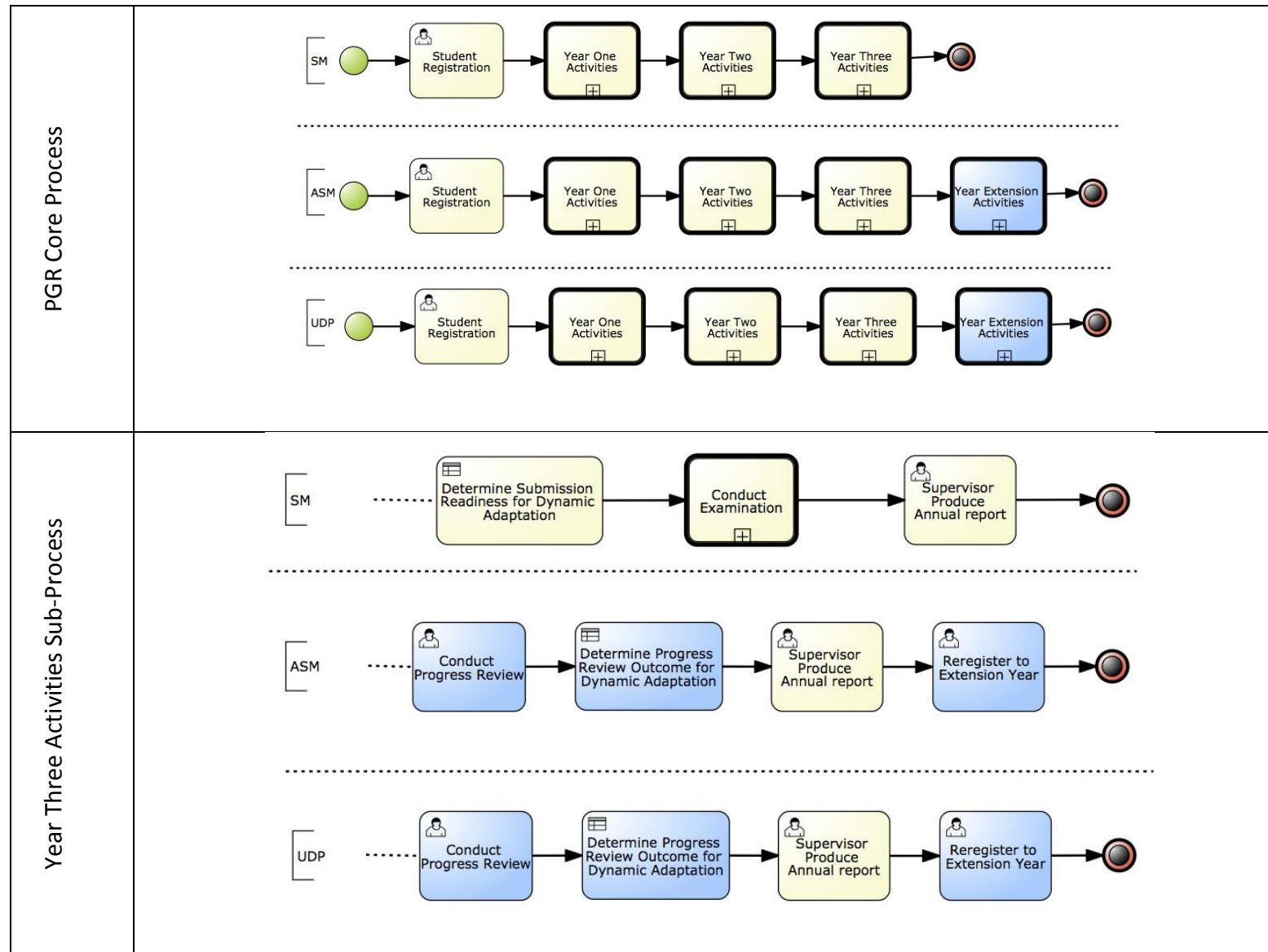


Figure 6-12
Consistent Three-
Tier Model after
Specific Instance
Dynamic Adaptation

(Inserted tasks are
indicated in blue
colour)

The proposed three-tier model introduces the ASM layer that is placed in between SM and UDP. ASM takes the role of SM in representing instance specific change of compliance requirements based on the dynamic events of the student. ACWfMS applied the three-tier model. Running the same test as above, changes on the specific instance are propagated back to UDP and ASM. SM is maintained to represent compliance requirements at the process structure level that is used as the initial template for all process models. The state of the three-tier model after the dynamic adaptation is shown Figure 6-12.

Both tests show that it is vital for the concept of the reference model in WfMS to maintain consistency after the dynamic adaptation process that is required by the compliance requirements. However, the test shows that the two-tier model is constrained to support adaptation for a collection of related process instances. The test confirmed on the targeted aspect of C5 that ACWfMS delivers the evolution of ASM with the concept of three-tier, non-static reference model in order to support a specific instance change, thus having a consistent reference model.

6.4.4 Scenario 4: Meeting and Training Frequency with Data Check

This scenario is set-up to evaluate C7 on checking compliance at the data level. This test scenario is based on frequency of supervisory meetings and number of training days per academic year that need to comply with the requirements of the University before students are allowed to re-register every year. SM for the First Year Sub-Process can be referred to in Section 6.3 (b), as an example of student yearly activities. Part of the process validates the frequency of meetings or number of training sessions on the third and sixth months of each registration year.

As stated in Sections 2.5 and 3.3.3, business processes such as BPMN are well established in representing logic and requirements at an abstract level. However, when it comes to low level specifications, the business process may become complicated and inefficient. On the other hand, the business rules technique has been used to define precisely the logic and requirements of a process in structured natural language. In Chapter 5, the ACWfMS architecture was proposed to adapt a hybrid technique that involves the integration of business processes and business

rules to support dynamic adaptation and compliance validation that requires run-time information.

This evaluation compares the technique of checking data during run-time by using classical BPMN process logic against the proposed ACWfMS hybrid technique. The test scenario with the classical BPMN process is shown in Figure 6-13. This process utilised the escalation event technique that identifies a situation for a process to react to. Escalation is thrown by either an 'End Event' or a 'Throwing Intermediate Event'. It is then caught either by the 'Event Boundary' or 'Event Sub-processes'.

In this test scenario the 'Validate Meeting' and 'Training Frequency' sub-processes calculate the meeting and training frequency. If either the meeting or training frequency is less than three or six on each respective sub-processes then 'Email Service' task is triggered to send a reminder to the respective student and supervisor(s).

The simple test scenario using classical BPMN process provides a well-defined model that specifies all the possible paths for each of its possible instances. However, this technique added more and more decisions and complex gateway logic, escalation paths and error handlings into the process model.

On the other hand, the ACWfMS hybrid technique simplifies the model by eliminating alternative process paths and describes constraints or deviations of different situations into the business rule statements. An example of the ACWfMS hybrid technique solution for this test scenario is shown in Figure 6-14. The supervisory meeting and training frequency is validated with a data check. The data check utilised business rules to verify if the number of meetings or training days does not comply. The process then needs to react to resolve these non-compliant activities.

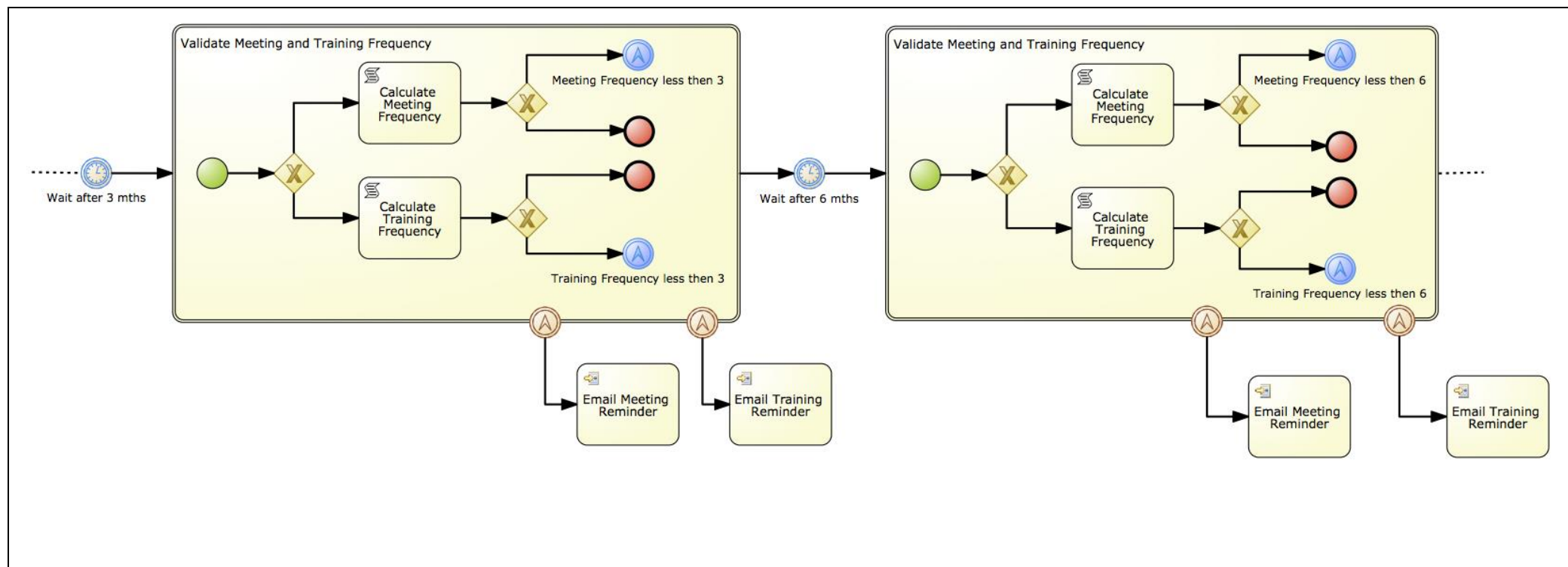


Figure 6-13 Validate Meeting and Training Frequency using BPMN process logic

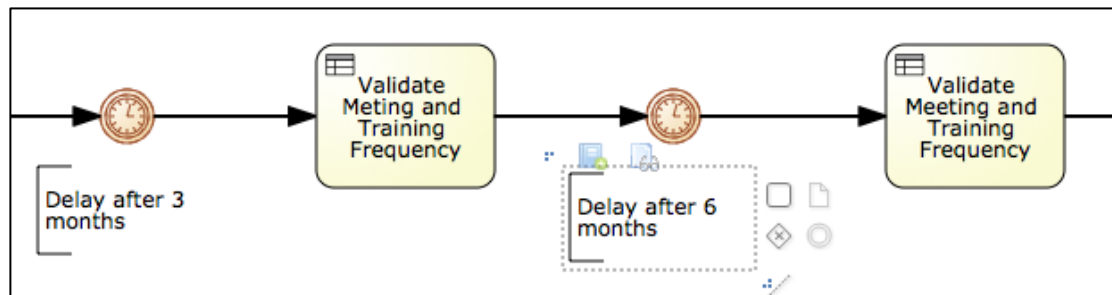


Figure 6-14 Validate Meeting and Training Frequency using ACWfMS hybrid techniques (Business Process and Business Rules)

In this test scenario, the number of training days after the third month of year one registration is less than two days. The process instance should be able to trigger 'Email Services' to send a training reminder to the student. ACWfMS was able to produce an expected result for data checking. An email for training reminder to student and supervisor is received as shown in Figure 6-15 and Figure 6-16.

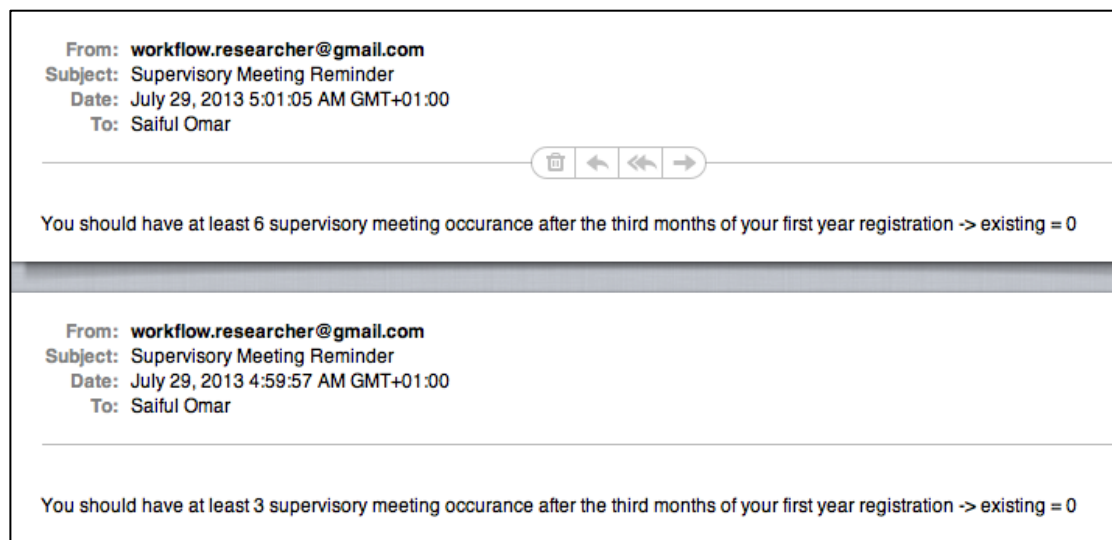


Figure 6-15 Supervisory Meeting Email Reminder

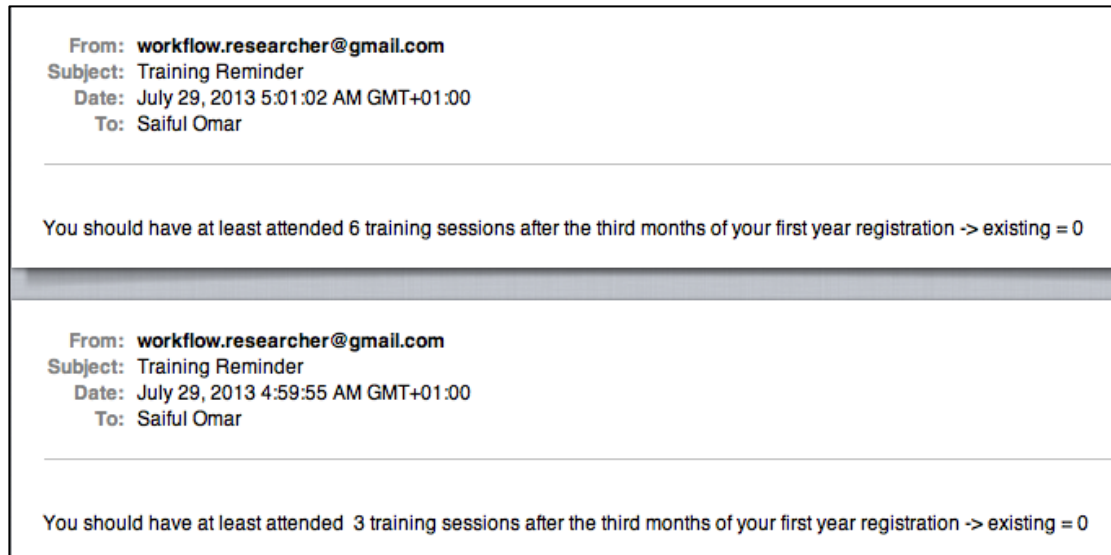


Figure 6-16 Attend Training Email Reminder

One of the important qualities of utilising the ACWfMS hybrid technique is to monitor and enforce run-time data to comply with the requirements. One of the advantages of applying the hybrid technique over classical BPMN pattern logic is on the ability to simplify a complex and hard-to-maintain process model and frequently changing rule set. Nevertheless, the balance of integration between process and rules in WfMS needs to be investigated further.

For this scenario, the test confirmed on the targeted aspect of C7 that ACWfMS checks compliance at data level by examining the instance against business rules with data check.

6.4.5 Scenario 5: Application for Leave of Absence with User Adaptation

This scenario is set-up to evaluate C8 for instance tracking and C9 for instance migration. This test scenario is based on an *ad-hoc* task where a student is applying for leave of absence with the user adaptation technique. The user adaptation technique has been evaluated in Scenario 3.

The proposed modules for instance tracker and instance migration were discussed in Section 5.2.4. User adaptation is performed at the process definition level. It is necessary to propagate the changes to the on-going process instances.

This test scenario makes a comparison in executing user adaptation with and without the ACWfMS instance tracker module. Without the instance tracker, the process editor will not be able to provide information of specific instance activity states. This may cause the user to change the process so that it will not comply with the current instance execution state. Figure 6-17 shows that the user added 'Apply Leave of Absence' activity in which the instance activity states has progressed too far.

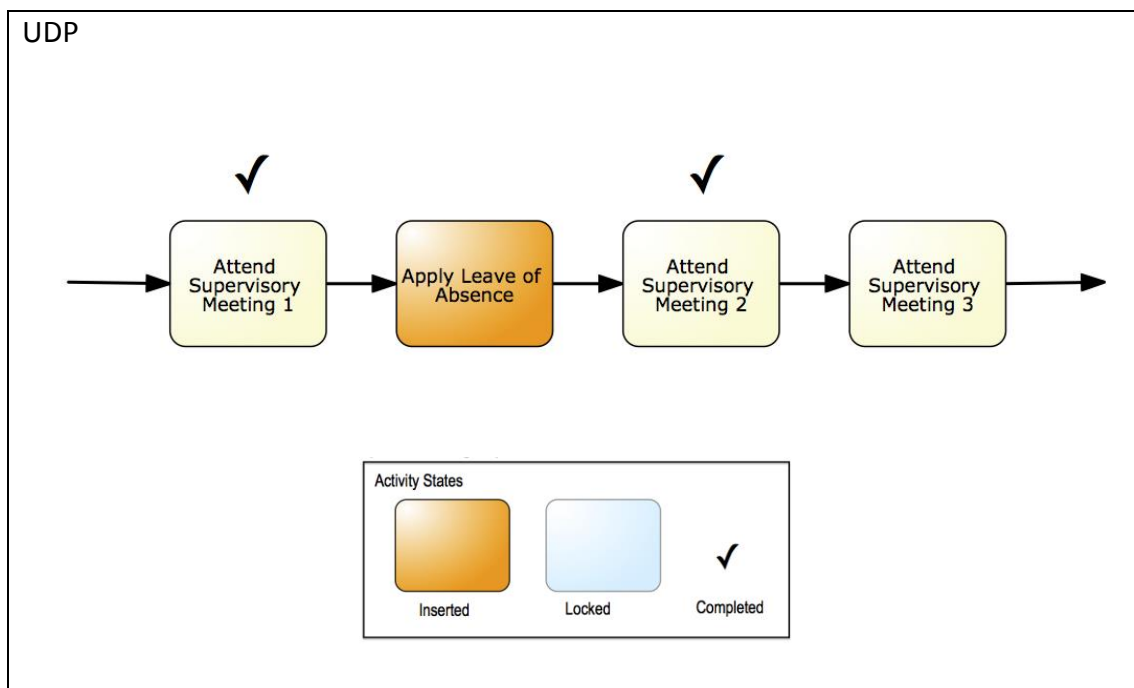


Figure 6-17 User Adaptation without ACWfMS Instance Tracker

With the ACWfMS instance tracker module, it tracks activities that have been executed in the process instance and signals the process editor to lock those executed activities from any further changes. As shown in Figure 6-18 the 'Attend Supervisory Meeting 1' and 'Attend Supervisory Meeting 2' activities are identified as completed activities and are locked by the editor. Executing the ACWfMS instance migration module migrates the affected instance with the updated process definition. For the instance migration module test, as shown in Figure 6-19, the inserted 'Apply Leave of Absence' activity in UDP is reflected in the workflow management of the current instance.

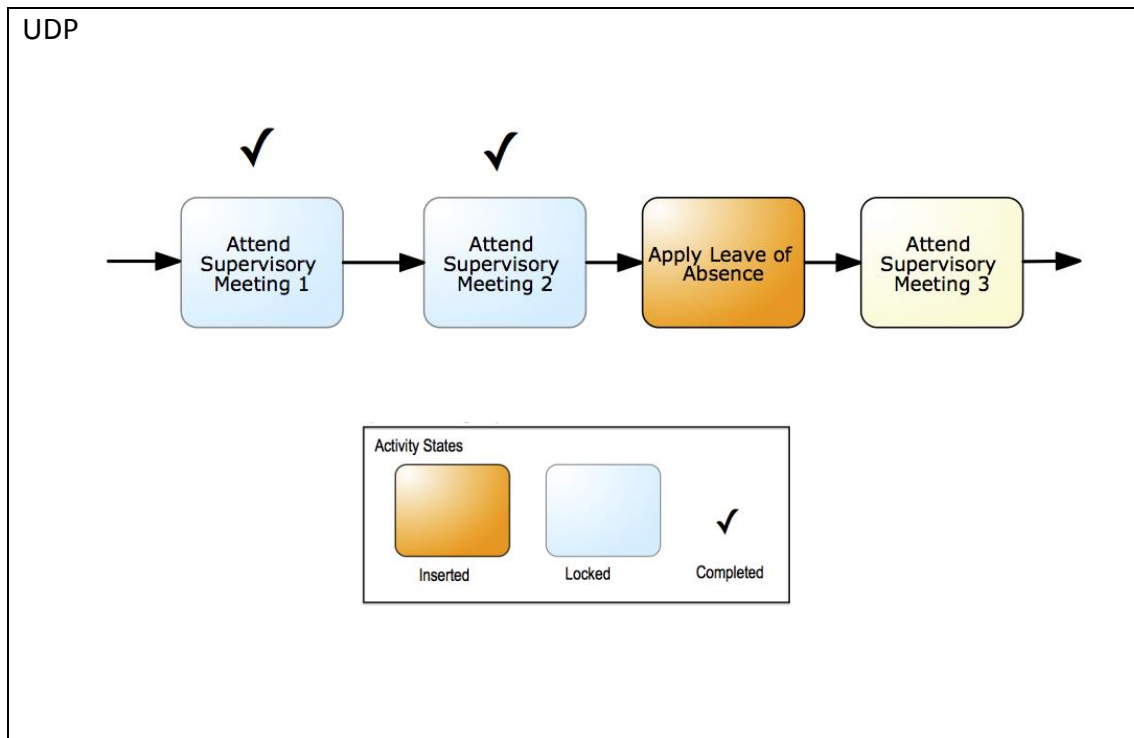


Figure 6-18 User Adaptation with ACWfMS Instance Tracker Module

Personal Tasks		
Refresh	View	Release Skip
Priority	Process	Task Name
0	PGR.YearOne	ApplyLeave

Figure 6-19 Specific Instance Migration with ACWfMS Instance Migration Module

The test confirmed on the targeted aspect of C8 and C9 on performing instance tracking and instance migration. ACWfMS is able to ensure the change process with user adaptation maintains affected instance states to be compliant with the new process definition.

6.5 Conclusion

In this chapter, an evaluation was established which contained nine criteria covering the integrated framework of adaptive and compliance workflow. A test plan was outlined based on the requirements identified in Section 3.4, that specifies which scenario will be used to evaluate which specific aspect of which criterion. The Postgraduate Research Process was used as the case study domain to evaluate

ACWfMS's ability to cope with specific instance adaptations effectively and at the same time, making sure each unique instance is compliant with the policy requirements.

The Postgraduate Research scenarios are used to evaluate each criterion that fits with the scope of this research on supporting instance specific changes. ACWfMS demonstrated how BPMN and business rules are used to capture compliance requirements that are represented as the standard model (SM).

It is convenient to define process models using the ACWfMS graphical process editor. However, in order to come to consensus among policy makers that are always distributed over many locations, the process editor could be enhanced with a collaborative modelling module.

Supporting flexible processes with selection and adaptation techniques were compared. The results showed that the selection technique is suitable to support static and repetitive business processes without any evolution of the process definition. The adaptation technique is suitable when required changes were not anticipated during build-time. ACWfMS applies both techniques, thus reducing the time in monitoring and making unnecessary changes in the process instance and UDP.

The scenarios test the effectiveness of ACWfMS to support the criterion of performing process change with user and dynamic adaptation and checking compliance at structural and data level. Having shown the ability to automate compliance check and support ad-hoc and dynamic evolution process change, it can be seen that the potential value of the ACWfMS lies in the adaptive reference model. All evaluation criteria that were tested and compared with ACWfMS's abilities against generic BPMN techniques met with satisfactory results.

Chapter 7

Conclusions and Future Work

7.1 Introduction

This chapter provides a critical discussion on the contribution of this Thesis and makes recommendations for future research. Section 7.2 reviews this Thesis and discusses the proposed system architecture and prototype. Section 7.3 summarises the main contribution of this Thesis. Section 7.4 outlines the limitations of ACWfMS; Section 7.5 discusses the applicability of the system; Section 7.6 discusses the directions for future research; and Section 7.7 provides some concluding remarks.

7.2 Thesis Review

The aim of this Thesis was defined as:

“To investigate current workflow technology in order to develop a novel solution of an integrated adaptive and compliance workflow management system architecture that handles dynamic and ad-hoc process modification, as well as automating compliance validation features throughout the process lifecycle”.

In order to achieve this aim, workflow technology was reviewed in Chapter 2 with a focus on the benefits, categories, standards and development tools used in adaptive and compliance workflow. Adaptive workflow supports ad-hoc processes through user adaptation and evolutionary changes with dynamic adaptation. Compliance in a workflow will enable an organisation to conform with required standards, regulations and rules. The compliance and adaptive features in workflow solutions need to be integrated, as having the adaptive capability without any mechanism to

ensure an updated process does not violate compliance requirements may lead to the execution of a non-compliant process. Therefore, a novel solution of an integrated adaptive and compliance development tool is necessary to support the workflow process lifecycles.

The third chapter of this Thesis consists of an investigation into the key aspects of integrating adaptive and compliance checking techniques in WfMS. These aspects were studied within the context of process change and compliance supports. The two techniques of process change were identified as adaption and selection. Applying both techniques enables workflow to be more dynamic, robust and time saving. Also, migration techniques were reviewed to deal with process instances that are currently running. Migration techniques are categorised as cancellation, with propagation and without propagation. The migration 'with propagation' technique delivers a real-time impact to the current instances.

Techniques to support compliance checks in workflow processes were also reviewed and classified as: build-time compliance checking; run-time compliance monitoring; and compliance auditing. The integration of build-time and run-time compliance supports is vital in order to identify compliance errors, assist in process specifications and prevent non-compliant tasks from being performed.

It was identified that a standard two-tier process model is insufficient to maintain and ensure that the concept of a reference model is valid throughout the adaptive process. Therefore, the three-tier process model was developed and introduced making sure the concept of a reference model would still be valid for compliance checking within the adaptive workflow environment. This is because the reference model will need to be changed for each instance depending on what library of sub-processes has been added.

Based on the review of both process change and compliance supports, nine system requirements were presented for integrating process adaptation and compliance techniques in a WfMS, as follows:

R1: *Representation for Business Process Modelling and Execution;*

R2: *Representation of Compliance Requirements into Standard Model;*

R3: *Representation of Compliance Requirements into Business Rules;*

R4: *Adaptation with Dynamic change and User Intervention;*

R5: *Evolution for Adaptive (non-Static) Standard Model;*

R6: *Compliance Validation at Structural level;*

R7: *Compliance Validation at Event and Data Level;*

R8: *Tracking Instance Support; and*

R9: *Migration Instance Support.*

The requirements were evaluated against existing adaptive and compliance integrated development tools and the outcomes show that there is no technique that covers all compliance validation scenarios and ensures compliance over a dynamic workflow lifecycle.

The fourth chapter presents the research methodology and evaluation methods of this Thesis. The chapter discusses the utilisation of a systems architecture engineering method as the main blueprint to design the ACWfMS prototype. The evaluation takes the form of a case study to investigate the application of ACWfMS to support the Postgraduate Research (PGR) process. The main challenge of implementing PGR with generic WfMS is the nature of the PGR process, that each PGR student has a unique process which cannot be pre-determined and is complex. Ad-hoc PGR activities require user adaptation to be added to the workflow process during run-time. Furthermore, the user adaptation needs to be validated to make sure that the updated process is still compliant with the PGR code of practice. The PGR process provides sufficient evaluation scenarios to validate the proposed ACWfMS architecture and evaluate the prototype as a proof of concept, specifically on the adaptation and compliance requirements.

Based on the requirements identified in chapter three, the fifth chapter of this Thesis presents the design of the system architecture to support an integrated workflow development tool that handles process adaptation and compliance validation. The major components of the architecture consist of a three-tier process model, business rules and plugin modules. The three-tier process model consists of the Standard Model, Adaptive Standard Model and User Defined Process. The three-tier process model used sub-processes to modify an instance during runtime, while at the same

time enabling automatic compliance checking. The architecture applies business rules to support dynamic adaptation and data-driven compliance validation. Furthermore, the WfMC architecture was extended to support new system requirements through plugin modules. Following the design of the system architecture, a prototype ACWfMS was implemented using the jBPM open source development platform.

The sixth chapter of this Thesis presents the evaluation case study based on the Postgraduate Graduate Research domain using ACWfMS as the development tool. A test plan was outlined with testing criteria based on the system requirements specified in Section 3.4. All criteria were evaluated. The ACWfMS abilities were compared against generic BPMN techniques. ACWfMS out performs BPMN in supporting process flexibility and compliance validation in all aspects.

7.3 Contributions

This section reviews the main contributions of this Thesis. Details of each contribution are aligned with the system requirements specified in Section 3.4.

Contribution 1: *Conceptual design of a novel adaptive and compliance workflow management system architecture for handling process adaptation and compliance features throughout the process lifecycle.*

This Thesis presented a set of system requirements for supporting an integrated adaptive and compliance workflow development tool. Based on the presented system requirements, this Thesis proposed the integrated system architecture to manage process adaptation and automate compliance validation. The major components of this architecture consist of process models, business rules and plugin modules. In more detail, the results concerning this architecture are:

- R1.** This Thesis demonstrated process modelling with BPMN standards that serve as graphical business process representations and workflow instance execution. The modelling is done through a process editor that translates the process requirements into a diagrammatical process model in turn providing clarity in understanding the abstract organisational policy concept.

- R2.** This Thesis applied standard model to capture process requirements that separate the compliance requirements from the actual workflow process. The standard model captures the static and frequently used paths and is used for compliance validation at the process structure level.
- R3.** This Thesis applied business rules to support dynamic adaptation and data-driven compliance validation. Business rules are used for compliance validation at task and data level.
- R4.** This Thesis presented a novel system architecture that automates compliance validation and assists dynamic workflow adaptation, specifically to deal with an evolving business process as it progresses according to situations that cannot always be prescribed. This architecture exhibits the technique of user adaptation with structural checks and dynamic adaptation with data-awareness checks. User adaptation is a technique of updating User-Defined Processes by means of user intervention and, in turn, making sure updated processes do not violate compliance requirements. Dynamic adaptation is the technique of updating specific instance at run-time by means of business rule logic to accommodate dynamic events of compliance requirements. Further, this Thesis presented a research prototype - ACWfMS is implemented based on the proposed system architecture.

Contribution 2: *Proposed three-tier reference models: Standard Model, Adaptive Standard Model and User-Defined Process.*

This Thesis identified that the two-tier reference model is insufficient to handle specific instance changes within the adaptive workflow environment. A three-tier reference model was proposed to ensure the concept of a reference model would still be valid before and after making specific instance changes. The Standard Model represents a knowledge base of organisational policy requirements and is used as an initial template for all process models. The Adaptive Standard Model represents the evolutionary standard model. The User-Defined Process represents the implementation process or case. A detailed discussion on the Adaptive Standard Model follows:

- R5.** This Thesis presented the Adaptive Standard Model as a dynamic reference model. The Adaptive Standard Model adapts to User-Defined Processes as it evolves by propagating the changes back to this model. At each moment, a User-Defined Process and workflow instance is attached to a single Adaptive Standard Model. The Adaptive Standard Model provides a consistent reference model and acts as a knowledge base to support compliance checks for a specific instance process execution.

Contribution 3: *Enhance process validation for non-compliant process through automatic tracking and managing conformance and process execution for specific instances.*

This Thesis presented the technique of visual explanation for structural compliance violations to the user via a process editor. This helps the user to understand and manage the compliance violation affecting the specific instance. Moreover, this Thesis identified that it is not sufficient to enforce all compliance checks at the structural level. Thus, this Thesis presented the technique of a data-driven compliance checker for run-time monitoring of business process execution. Automating compliance checks saves the effort of manually identifying such violations. The detailed adaptation techniques follows:

- R6.** This Thesis presented the *user adaptation* technique with *structural checks* that is applied either during build-time or run-time. During build-time the initial User-Defined Process can be improved to suit an individual process. During run-time the user adaptation may support ad-hoc or unanticipated activities. Moreover, this Thesis demonstrated the structural compliance module which validates the correctness and completeness of process flow for User-Defined Process against the Adaptive Standard Model. Through the process editor, visual compliance feedback is generated to identify non-compliance process behaviour.
- R7.** This Thesis presented the *dynamic adaptation* technique with a *data-driven check* that is applied during run-time. This technique applies the business rules to monitor relevant data during the process execution. This technique detects

potential violations as early as possible in order to allow strategies to resolve compliance violations.

Contribution 4: *Propose an instance tracker tool that assists process adaptation via process editor.*

R8. This Thesis presented an instance tracking module that tracks process instances through execution logs. This Thesis extended the jBPM process editor with the instance tracking module as a plugin. With the extended plugin, the process editor can be used to ensure compliance during user adaptation by locking executed nodes from further changes and to avoid control and data loss.

Contribution 5: *Propose an instance migration tool that allows an updated process instance to continue executing based on the updated logic and to cope with dynamic changes during run-time.*

R9. This Thesis presented an instance migration module that propagates updated processes to the on-going process instances. This Thesis extended the jBPM console with the instance migration module as a plugin. With the extended plugin, the console can be used to migrate affected process instances with updated process definitions after a successful user adaptation.

7.4 Applicability of the System

The results obtained in this study are applicable to a broader tool and domain:

- At a workshop on the Challenges of Scientific Workflows (Gil et al. 2007) issues were discussed by the domain scientists, computer scientists, and social scientists and the conclusions were that (i) workflows can provide a formal specification of scientific analysis processes from the data collection, through analysis to data publication; (ii) workflows could accelerate the pace of scientific progress by supporting scientists in creating, merging, executing and re-using processes; and (iii) workflows can act as key enablers for reproducibility of experiments involving large scope computations. The workshop stressed the vision of scientific workflow in supporting dynamic,

adaptive and user-steered workflows. The scientific workflows are typically data-centric as opposed to task-centric business workflows. Wang et al. (2009) highlighted that the correctness of data flows is crucial to the execution of scientific workflows. ACWfMS has proven its capabilities in supporting dynamic adaptation and user adaptation as demonstrated in Chapter 6. Further, the example shown in Section 5.2.5 shows that ACWfMS supports structural compliance checking for control flow and data flow. Therefore, ACWfMS has the potential to support the requirements of scientific workflow.

- Case Management refers to the coordination of the work that is unpredictable and requires human intervention (Burns 2011; Motahari-Nezhad & Swenson 2013). With the basic description of the case management, ACWfMS may be applied as a case management platform, specifically by using the ACWfMS adaptive reference model technique to handle case templates and ad-hoc tasks.
- The public sector is constantly faced with pressures to reform, comply and improve their services. With the increasing demands from the public and stakeholders, organisations are being closely monitored to ensure that their services are creative, innovative, efficient, and dynamic. ACWfMS supports compliance checking automation that adapts to a dynamic environment. Thus, the applicability of ACWfMS in the public sector can increase compliance, transparency, accountability, efficiency and public satisfaction. Further, ACWfMS supports the public sector with the rapid adaptation to changes in government act or legislation.

7.5 Limitations of the Research and System

Although a number of novel contributions on adaptive and compliance workflow were presented in this Thesis, there is always room for improvement. The research limitations are:

- Due to time limitations, the scope of this study is limited to the specific instance process adaptation rather than a collection of instances.

Nevertheless, the proposed architecture, specifically the three-tier process model, was designed to cope at all levels of process adaptation;

- Due to the scale of the project, limited evaluation of the case domain was used to test, demonstrate and evaluate key components of ACWfMS;

And, for the system, limitations are:

- The system architecture presented in this Thesis focussed on centralised WfMS. Montagut and Molva (2007) suggested that a distributed WfMS eliminates the performance bottleneck issues;
- The compliance check module uses the actual activity name or label to match the activities between the adaptive standard model and user-defined process. Thus, the compliance checks will not be able to resolve variations of names. This can be addressed by using ontology capabilities, where the term used to describe the activity “can be shared and re-used by others in the same domain to minimise ambiguity” (Chung et al. 2008).

7.6 Recommendation for Future Work

Based on the identified limitations in Sections 7.5, future work in the expansion of the research scope involves:

- Extend the scope of the study to cater for a collection of workflow instances of process adaptation. A possible approach is to identify modification policies for handling active workflow instances and reference models;
- Further evaluation on ACWfMS with more case domains that include real users from the industry;

Future work to improve the system implementation involves:

- Identify system requirements to support process adaptation and compliance for a distributed WfMS environment. Based on the identified requirements, a new system architecture is to be designed that supports the compliance checks automation technique to serve local and global reference models. The system architecture should also manage and track individual instances that are involved in the synchronisation within the collaborative workflow

processes for user and dynamic adaptation. Additional system requirements based on security, time and privacy are also being considered;

- Develop a process dictionary with the use of domain ontology. The process dictionary entries may come from the policy regulator or business requirement documents. During process modelling the tasks are derived from the process dictionary items. The use of explicit, reusable and sharable domain ontologies has broad applicability in workflow technology. ACWfMS can utilise domain ontology to enable a consistent match between the model and user-defined process entities for compliance checking. Further, the use of ontologies overcomes the challenges in a distributed WfMS to define and manage process vocabularies across organisational collaboration.

7.7 Concluding Remarks

This Thesis reviewed the concept of workflow technology with particular techniques to support the integration of adaptive and compliance workflow systems. The review identified a lack of automated tools to ensure the specification and execution of enterprise processes, that are dynamic in nature, complied with policy standards. A set of requirements to bridge the automation of integrated process adaptation and compliance supports was presented. The contribution of this Thesis is the development of a novel system architecture that provides assistance in detecting non-compliant errors while managing flexibility and adaptation for specific process instances according to situations that cannot always be prescribed. A research prototype, ACWfMS, was developed based on the proposed system architecture for testing and evaluation purposes. A study on a postgraduate research process was performed to evaluate the novel features. It is concluded that the design and implementation of the novel architecture that forms the major component of ACWfMS has realised the aim and objectives set out at the beginning of this Thesis based on the result of a case study evaluation. Future work to widen the scope of the research to include real users and to extend the functionalities of ACWfMS is proposed.

References

- Aalst, W.M.P. Van Der et al., 2009. Flexibility as a Service. In *Database Systems for Advanced Applications*. Springer-Verlag, pp. 319–333.
- Aalst, W.M.P. Van Der et al., 2003. Workflow Patterns. *Distributed and Parallel Databases*, 14(1), pp.5–51.
- Aalst, W.M.P.V. Der & Jablonski, S., 2000. Dealing with workflow change: identification of issues and solutions. *Computer systems science and engineering*, 5(5), pp.267–276.
- Agostini, A. & De Michelis, G., 2000. A light workflow management system using simple process models. *Computer Supported Cooperative Work (CSCW)*, 9(3-4), pp.335–363.
- Alavi, M., 1984. An assessment of the prototyping approach to information systems development. *Communications of the ACM*, 27(6), pp.556–563.
- Asuncion, C.H., Iacob, M. & van Sinderen, M.J., 2010. Towards a Flexible Service Integration through Separation of Business Rules. In *2010 14th IEEE International Enterprise Distributed Object Computing Conference*. IEEE, pp. 184–193.
- Babar, M.A. & Gorton, I., 2004. Comparison of scenario-based software architecture evaluation methods. In *Software Engineering Conference, 2004. 11th Asia-Pacific*. pp. 600–607.
- Baeyens, T., 2004. The State of Workflow. Available at: <http://www.theserverside.com/news/1365159/The-State-of-Workflow> [Accessed August 1, 2012].
- Becker, J. et al., 2011. Supporting Business Process Compliance in Financial Institutions—A Model-Driven Approach. In *10th International Conference on Wirtschaftsinformatik*. pp. 355–364.
- Boehm, B. et al., 1995. Software requirements negotiation and renegotiation aids: A theory-W based spiral approach. In *Software Engineering, 1995. ICSE 1995. 17th International Conference*. p. 243.
- Bonitasoft, 2013. Bonitasoft. Available at: <http://www.bonitasoft.com/> [Accessed June 9, 2013].
- Börger, E., 2012. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software & Systems Modeling*, pp.305–318.

- BPMN, 2011. Business Process Model and Notation. Available at: <http://www.omg.org/spec/BPMN/2.0/> [Accessed March 27, 2013].
- Brien, P.D.O. & Wiegand, M.E., 1998. Agent based process management : applying intelligent agents to workflow. *Engineering*, 13(Ker 13207), pp.161–174.
- Browne, P., 2009. *JBoss Drools Business Rules*, Birmingham: Packt Publishing.
- Burns, E. V., 2011. *Case Management 101: 10 Things You Must Know About Case Management* L. Fischer, ed., Future Strategies Inc.
- Carroll, J.M., 2000. Five reasons for scenario-based design. *Interacting with computers*, 13(1), pp.43–60.
- Charfi, A. & Mezini, M., 2004. Hybrid web service composition: business processes meet business rules. *Proceedings of the 2nd international conference on Service oriented computing*, pp.30–38.
- Chinosi, M. & Trombetta, A., 2012. BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1), pp.124–134.
- Chung, P.W.H., Cheung, L. & Machin, C., 2008. Compliance Flow – Managing the compliance of dynamic and complex processes. *Knowledge-Based Systems*, 21(4), pp.332–354.
- Dawson, C., 2005. *Projects in computing and information systems: a student's guide* Second Edi., Addison Wesley Publishing Company.
- Decker, G., Overdick, H. & Weske, M., 2008. *Oryx—an open modeling platform for the BPM community*, Springer Berlin Heidelberg.
- Dijkstra, E.W., 1983. The structure of “THE”-multiprogramming system. *Communications of the ACM*, 26(1), pp.49–52.
- Drools Community, 2013. Drools - JBoss. Available at: <http://www.jboss.org/> [Accessed July 12, 2013].
- Earl, M.J., 1982. Prototype systems for accounting, information and control. *ACM SIGMIS Database*, 13(2-3), pp.39–46.
- Forgy, C.L., 1982. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1), pp.17–37.
- Fötsch, D., Pulvermueller, E. & Rossak, W., 2006. Modeling and Verifying Workflow-based Regulations. In *Conference on Advanced Information Systems Engineering*. pp. 825–830.
- Garlan, D., 2000. Software architecture: a roadmap. In *ICSE '00 Proceedings of the Conference on The Future of Software Engineering*. New York: ACM Press, pp. 91–101.

- Gil, Y. et al., 2007. Examining the Challenges of Scientific Workflows. *Computer*, 40(12), pp.24–32.
- Goedertier, S. & Vanthienen, J., 2007. Declarative Process Modeling with Business Vocabulary and Business Rules. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*. pp. 603–612.
- Governatori, G. & Sadiq, S., 2009. The journey to business process compliance. *Handbook of Research on BPM*, pp.1–32.
- Harmon, P. & Wolf, C., 2011. *Business Process Modeling Survey*, Business Process Trends.
- Hay, D. & Healy, K., 2000. *Defining business rules what are they really?*, Business Rules Group.
- He, X. et al., 2007. A metamodel for the notation of graphical modeling languages. In *Computer Software and Applications Conference*. pp. 0–3.
- Hee, K. van, 2004. *Workflow management: models, methods, and systems*, The MIT press.
- Hill, J.B., Pezzini M. & Natis, Y.V., 2008. Findings: confusion remains regarding BPM terminologies. *Gartner Research, Stamford, CT.*, G00155817.
- Hockey, J., 1994. New territory: Problems of adjusting to the first year of a social science PhD. *Studies in Higher Education*, 19(2), pp.177–190.
- Hollingsworth, D., 2004. The workflow reference model: 10 years on (2004). In *Fujitsu Services, UK; Technical Committee Chair of WfMC*.
- JBoss, 2013. JBCRA. *2013 JBoss Community Recognition Awards*. Available at: <https://www.jboss.org/jbcra/jbcra2013>.
- JBPM, 2013. jBPM - JBoss Community. Available at: <http://www.jboss.org/jbpm/> [Accessed June 9, 2013].
- JBPM, 2012. jBPM User Guide. Available at: <http://docs.jboss.org/jbpm/v5.4/userguide/> [Accessed August 15, 2013].
- JBPM, 2013. Web-based BPMN2.0 Designer for jBPM. Available at: <https://github.com/droolsjbpm/jbpm-designer> [Accessed July 13, 2013].
- Jose, S., 2011. Global Business Process Management Market to Exceed US\$5.0 Billion by 2017. *Global Industry Analysts, Inc.* Available at: http://www.prweb.com/releases/business_process/management_BPM/prweb8285965.htm [Accessed August 9, 2012].
- Kazman, R. et al., 1996. Scenario-based analysis of software architecture. *Software, IEEE*, 13(6), pp.47–55.

- Kharbili, M. El et al., 2008. Business Process Compliance Checking: Current State and Future Challenges. In *MobIS*. pp. 107–114.
- Kluza, K., Kaczor, K. & Nalepa, G., 2012. Enriching business processes with rules using the Oryx BPMN editor. In *Proceedings of the 11th international conference on Artificial Intelligence and Soft Computing-Volume Part II*. Springer-Verlag, pp. 573–581.
- Knolmayer, G., Endl, R. & Pfahrer, M., 2000. Modeling processes and workflows by business rules. In *Business Process Management*. Springer Berlin Heidelberg, pp. 16–29.
- Knott, R. & Dawson, R., 1999. *Software project management*, Group D Publications Limited.
- Ko, R.K.L., 2009. A computer scientist's introductory guide to business process management (BPM). *Crossroads*, 15(4), pp.11–18.
- Ko, R.K.L., Lee, S.S.G. & Lee, E.W., 2009. Business process management (BPM) standards: a survey. *Business Process Management Journal*, 15(5), pp.744–791.
- Koehler, J. & Vanhatalo, J., 2007. Process anti-patterns: How to avoid the common traps of business process modeling. *IBM WebSphere Developer Technical Journal*, 10.2.
- Kradolfer, M. & Geppert, a., 1999. Dynamic workflow schema evolution based on workflow type versioning and workflow migration. *Proceedings Fourth IFCIS International Conference on Cooperative Information Systems. CoopIS 99 (Cat. No.PR00384)*, pp.104–114.
- Kumar, A. et al., 2010. Ensuring compliance with semantic constraints in process adaptation with rule-based event processing. In *RuleML'10 Proceedings of the 2010 international conference on Semantic web rules*. Berlin, Heidelberg: Springer-Verlag, pp. 50–65.
- Kumar, A., Yao, W. & Chu, C., 2012. Flexible Process Compliance with Semantic Constraints Using Mixed-Integer Programming. *INFORMS Journal on Computing*, 25(3), pp.543–559.
- De Leoni, M., Mecella, M. & De Giacomo, G., 2007. Highly Dynamic Adaptation in Process Management Systems Through Execution Monitoring. In *Proceedings of the 5th International Conference on Business Process Management*. Berlin, Heidelberg: Springer-Verlag, pp. 182–197.
- Lethbridge, T.C., Sim, S.E. & Singer, J., 2005. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(3), pp.311–341.





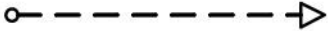
- Liu, X., Su, J. & Yang, J., 2011. Preservation of Integrity Constraints by Workflow. In *Proceedings of the 2011th Confederated International Conference on On the Move to Meaningful Internet Systems - Volume Part I*. Berlin, Heidelberg: Springer-Verlag, pp. 64–81.
- Loughborough University, 2012. Code of Practice on Research Degree Programmes. Available at: <http://www.lboro.ac.uk/students/welcome/research/codes-of-practice/> [Accessed July 15, 2013].
- Lu, R., Sadiq, S. & Governatori, G., 2008. Measurement of Compliance Distance in Business Processes. *Information Systems Management*, 25(4), pp.344–355.
- Ly, L., Knuplesch, D. & Rinderle-Ma, S., 2011. *SeaFlows Toolset—compliance verification made easy for process-aware information systems*, Springer Berlin Heidelberg.
- Mendling, J., 2009. Event-driven process chains (epc). In *Metrics for Process Models*. Springer Berlin Heidelberg, pp. 17–57.
- Mendling, J., Lassen, K.B. & Zdun, U., 2008. On the transformation of control flow between block-oriented and graph-oriented process modelling languages. *International Journal of Business Process Integration and Management*, 3(2), pp.96–108.
- Montagut, F. & Molva, R., 2007. Enforcing integrity of execution in distributed workflow management systems. In *IEEE International Conference on Services Computing*. Salt Lake City, UT, pp. 170–177.
- Motahari-Nezhad, H.R. & Swenson, K.D., 2013. Adaptive Case Management: Overview and Research Challenges. In *2013 IEEE 15th Conference on Business Informatics*. IEEE, pp. 264–269.
- Mutschler, B., 2008. Unleashing the effectiveness of process-oriented information systems: Problem analysis, critical success factors, and implications. *Systems, Man, and Cybernetics*, 38(3), pp.280–291.
- Naumann, J.D. & Jenkins, A.M., 1982. Prototyping: The New Paradigm for Systems Development. *MIS Quarterly*, 6(3), pp.29–44.
- Nunes, V.V.T., Werner, C.M.L. & Santoro, F.M., 2011. Dynamic process adaptation: A context-aware approach. *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp.97–104.
- Nurcan, S., 2008. A Survey on the Flexibility Requirements Related to Business Processes and Modeling Artifacts. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. Waikoloa, HI: IEEE, pp. 378–378.

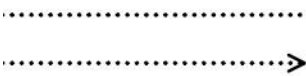





- OASIS, 2007. Web Services Business Process Execution Language Version 2.0. *wsbpel-v2.0-OS*. Available at: <https://www.oasis-open.org> [Accessed April 30, 2013].
- Omar, M.S.H., Chung, P.W.H. & Dawson, C.W., 2013. A Novel Workflow Management System for Handling Process Adaptation and Compliance. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, pp. 1174–1179.
- OMG, 2011. BPMN. *Object Management Group*. Available at: <http://www.omg.org/spec/BPMN/2.0> [Accessed August 6, 2012].
- Pesic, M., 2007. Declare: Full support for loosely-structured processes. In *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*. Annapolis, MD, pp. 287–287.
- Poppendieck, M. & Poppendieck, T., 2006. *Implementing Lean Software Development: From Concept to Cash*, Addison-Wesley Professional.
- Rademakers, T., 2013. *Activiti in Action: Executable business processes in BPMN 2.0*, Manning Publications Co.
- Redhat, 2013. jBPM - JBoss Community. Available at: <http://www.jboss.org/jbpm> [Accessed July 5, 2012].
- Reichert, M., Rinderle, S. & Dadam, P., 2003. ADEPT workflow management system: flexible support for enterprise-wide business processes. In *BPM'03 Proceedings of the 2003 international conference on Business process management*. pp. 370–379.
- Reijers, H.A., Rigter, J.H.M. & Alast, W.M.P., 2003. The case handling case. *International Journal of Cooperative Information Systems*, 12(3), pp.365–391.
- De Rezende, J. et al., 2006. Supporting Student-Supervisor Scientific Collaboration. In *2006 10th International Conference on Computer Supported Cooperative Work in Design*. Ieee, pp. 1–6.
- Del Rio, A. & Soluciones, D., 2012. The great unknown: BPM Open Source - BPM Leader. Available at: <http://www.bpmleader.com/2012/03/22/the-great-unknown-bpm-open-source/> [Accessed August 7, 2012].
- Robson, C., 2002. *Real world research: A resource for social scientists and practitioner-researchers*, Wiley-Blackwell.
- Rozinat, A. & Aalst, W.M.P. Van Der, 2008. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1), pp.64–95.

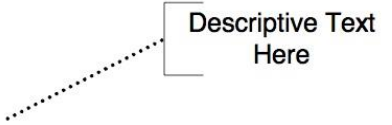
- Runeson, P. & Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), pp.131–164.
- Sackmann, S. & Kahmer, M., 2008. A classification model for automating compliance. In *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing*. Washington,DC, pp. 79–86.
- Schonenberg, H. & Mans, R., 2008. Towards a taxonomy of process flexibility. In *Proceedings of CAiSE'08 Forum*. Montpellier, France, pp. 81–84.
- Schrage, M., 2004. Never go to a client meeting without a prototype. *Software, IEEE*, 21(2), pp.42–45.
- Schumm, D. et al., 2010. Business Process Compliance Through Reusable Units of Compliant Processes. In *Proceedings of the 10th International Conference on Current Trends in Web Engineering*. Berlin, Heidelberg: Springer-Verlag, pp. 325–337.
- Sutcliffe, A., 2003. Scenario-based requirements engineering. In *Requirements engineering conference, 2003. Proceedings. 11th IEEE international*. pp. 320–329.
- Wang, L., Huang, Z. & Luo, M., 2009. Supporting Dynamic Workflow Adaptation in a Dataflow-Constrained Workflow Net. In *2009 International Conference on New Trends in Information and Service Science*. IEEE, pp. 1000–1005.
- Weber, B., Reichert, M. & Rinderle-Ma, S., 2008. Change patterns and change support features – Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, 66(3), pp.438–466.
- Weber, B., Sadiq, S. & Reichert, M., 2009. Beyond rigidity – dynamic process lifecycle support. *Computer Science - Research and Development*, 23(2), pp.47–65.
- Weske, M., 2001. Formal foundation and conceptual design of dynamic adaptations in a workflow management system. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference*.
- Weske, M., Aalst, W.M.P.V. Der & Verbeek, H.M.W., 2004. Advances in business process management. *Data & Knowledge Engineering*, 50(1), pp.1–8.
- WfMC, 1999. *Workflow Management Coalition Terminology & Glossary*, Workflow Management Coalition.
- WfMC, 1995. *Workflow Management Coalition The Workflow Reference Model*, Workflow Management Coalition.
- WfMC, 2012. XML Process Definition Language. *WfMC-TC-1025*. Available at: <http://www.xpdl.org/> [Accessed April 30, 2013].

- White, S., 2004. *Introduction to BPMN*, IBM Cooperation.
- White, S., 2005. *Using BPMN to model a BPEL process*, BPtrends.
- Workflow Patterns Initiative, 2011. Workflow Patterns Initiative. Available at: <http://www.workflowpatterns.com/> [Accessed April 29, 2013].
- Yin, R.K., 2003. *Case Study Research: Design and Methods*, Thousand Oaks, USA: Sage Publications.

Appendix 1 - Basic BPMN Modelling Elements and Description

Element	Description	Notation
Event	An Event is something that “happens” during the course of a Process or a Choreography. These Events affect the flow of the model and usually have a cause (<i>trigger</i>) or an impact (<i>result</i>). Events are circles with open centers to allow internal markers to differentiate different <i>triggers</i> or <i>results</i> . There are three types of Events, based on when they affect the	
Activity	An Activity is a generic term for work that company performs in a Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process Model are: Sub-Process and Task, which are rounded rectangles. Activities are used in both standard Processes and in Choreographies.	
Gateway	A Gateway is used to control the divergence and convergence of Sequence Flows in a Process and in a Choreography. Thus, it will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behaviour control.	
Sequence Flow	A Sequence Flow is used to show the order that	
Message Flow	A Message Flow is used to show the flow of Messages between two <i>Participants</i> that are prepared to send and receive them. In BPMN, two separate Pools in a Collaboration Diagram will represent the two <i>Participants</i> (e.g., PartnerEntities and/or PartnerRoles).	

Association	An Association is used to link information and Artifacts with BPMN graphical elements. Text Annotations and other Artifacts can be Associated with the graphical elements. An arrowhead on the Association indicates a direction of flow (e.g., data), when appropriate.	
Pool	A Pool is the graphical representation of a Participant in a Collaboration. It also acts as a “swimlane” and a graphical container for partitioning a set of Activities from other Pools, usually in the context of B2B situations. A Pool MAY have internal details, in the form of the Process that will be executed. Or a Pool MAY have no internal details, i.e., it can be a “black box.”	
Lane	A Lane is a sub-partition within a Process, sometimes within a Pool, and will extend the entire length of the Process, either vertically or horizontally. Lanes are used to organize and categorize Activities.	
Data Object	Data Objects provide information about what Activities require to be performed and/or what they produce, Data Objects can represent a singular object or a collection of objects. Data Input and Data Output provide the same information for Processes.	
Message	A Message is used to depict the contents of a communication between two Participants (as defined by a business PartnerRole or a business PartnerEntity).	
Group (a box around a group of objects within the same category)	A Group is a grouping of graphical elements that are within the same Category. This type of grouping does not affect the Sequence Flows within the Group. The Category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which Categories of objects can be visually displayed on the diagram.	

<p>Text Annotation (attached with an Association)</p>	<p>Text Annotations are a mechanism for a modeller to provide additional text information for the reader of a BPMN Diagram.</p>	 <p>Descriptive Text Here</p>
---	---	--

Appendix 2 – Publication and Recognition Resulted From This Thesis

- A conference paper was presented for the 27th IEEE International Conference on Advanced Information Networking and Applications Workshops, Barcelona, Spain, March 25-28, 2013. The title of the published paper, “A Novel Workflow Management System for Handling Process Adaptation and Compliance”, (Omar et al. 2013), Page 1174 – 1179, DOI 10.1109/WAINA.2013.258. This conference paper will be extended into a journal publication.
- Nominated for the ‘JBoss Community Recognition Awards 2013’ under ‘New Features’ (JBoss, 2013). The contributions were made during the prototyping stages. First contribution is on ‘Locking and Unlocking’ feature of the jBPM Designer (web-based business process editor) that allows users of jBPM Designer to lock certain parts of the business process model in order to foster collaboration during the modelling phase of business process model. Second contribution is on ‘Microsoft Academic Search Service Node’, the first community contributed community jBPM service node definition and implementation.

Appendix 3 – Glossary of Terms

API (Application Programming Interface)	An application programming interface (API) is a software program that facilitates interaction with other software programs.
BPMN (Business Process Model and Notation)	A standard set of graphical shapes and conventions with associated meanings that can be used in modelling a business process. It is currently maintained by the Object Management Group (OMG). BPMN2 standard comes with dual functionality: diagrams to communicate and modelling for execution.
Orchestration	Orchestration defines processes that are internal to a specific organization. Thus, they are contained within a single Pool.
Choreography	Choreography focus on the coordination of interactions of participants in the model. Choreography provide details of the exchange of messages between Pools.
Collaborations	Collaboration depicts the interactions between two or more business entities.
Build Time	The period of time when automated and/or manual workflow process descriptions are defined and/or modified electronically.
Business Process	A set of one or more linked activities which collectively realise a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.
BPEL (Business process execution language)	A standard executable language, based on XML, for describing a process that uses web service calls to communicate with the outside world.
OASIS	OASIS is a non-profit global consortium that drives the

(Organization for the Advancement of Structured Information Standards)	development, convergence, and adoption of e-business and web service standards.
OMG (Object Management Group)	OMG is an international, open membership, not-for-profit computer industry standards consortium. OMG Task Forces develop enterprise integration standards for a wide range of technologies and an even wider range of industries.
Process definition	The computerised representation of a process that includes the manual definition and workflow definition.
Process Instance	Process instance represents one specific instance of a process that is currently executing. Whenever a process is started, a process instance is created that represents that specific instance that was started. It contains all runtime information related to that instance.
Run Time	The period of time during the process is operational, with process instances being created and managed.
Workflow	A workflow is an automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.
WfMC (Workflow Management Coalition)	A consortium, formed to define standards for the interoperability of workflow management systems. It was founded in May 1993 as an offshoot of the Black Forest Group with original members including IBM, Hewlett-Packard, Fujitsu, ICL, Staffware and approximately 300 software and services firms in the business software sector.

Workflow Management System (WfMS)	A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.
WS-BPEL (Web Services Business Process Execution)	WS-BPEL is a standard executable language for specifying actions within business processes with web services.
XPDL (XML Process Definition Language)	XPDL is a format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products. XPDL is designed to exchange the process definition, both the graphics and the semantics of a workflow business process.