Generic Business Process Modelling Framework for Quantitative Evaluation

Jamal Mustafa Al-Tuwaijari

Newcastle University

Newcastle upon Tyne, UK.

A thesis submitted for the degree of

Doctor of Philosophy

April, 2013

To My Loving Parents and Family

Abstract

Business processes are the backbone of organisations used to automate and increase the efficiency and effectiveness of their services and products. The rapid growth of the Internet and other Web based technologies has sparked competition between organisations in attempting to provide a faster, cheaper and smarter environment for customers. In response to these requirements, organisations are examining how their business processes may be evaluated so as to improve business performance.

This thesis proposes a generic framework to expand the applicability of various quantitative evaluation to a large class of business processes. The framework introduces a novel engineering methodology that defines a modelling formalism to represent business processes that can be solved for a set of performance and optimisation algorithms. The methodology allows various types of algorithms used in model-based business process improvement and optimisation to be plugged in a single modelling formalism. As a part of the framework, a generic modelling formalism (MWF-wR) is developed to represent business processes so as to allow quantitative evaluation and to select the parameters for the associated performance evaluation and optimisation.

The generic framework is designed and implemented by developing software support tools using Java as object oriented programming language combining three main modules: (i) a business process specification module to define the components of the business process model, (ii) a stochastic Petri net module to map the business process model to a stochastic Petri net, and (iii) an algorithms module to solve the models for various performance optimisation objectives. Furthermore, a literature survey of different aspects of business processes including modelling and analy-

sis techniques provides an overview of the current state of research and highlights gaps in business process modelling and performance analysis. Finally, experiments are introduced to investigate the validity of the presented approach.

Acknowledgements

First and foremost I would like to express my gratitude to my supervisor Prof. Aad van Moorsel for giving me the opportunity and support to complete this research. For all his valuable comments, hints, suggestions and many more that inspired this work. I would have been lost without his support and I wish to express my appreciation.
I would also like to thank the members of my thesis committee, Prof. Pete Lee and Dr. Carlos Molina-Jimenez. My gratitude is also extended to the School of Computing Science for giving me the opportunity to undertake this memorable learning experience. I would also like to thank all people and staff of the School who assisted me at different stages of the research. I would also like to thank all my friends for their support and being a lively and entertaining part of the journey. I dedicate this research to my home country Iraq for supporting me financially.
I wish to acknowledge my greatest debt to my family, in particular my mother, father, uncle and my brothers and sisters. They never stopped encouraging me to finish this thesis. Finally, heartfelt thanks go to my wife and my children who have stood behind me and given me huge support during the thesis work. They suffered considerably because of my academic interests. I thank Allah for all that and all the things that have happened to me in my life.

Jamal Al-Tuwaijari

Declaration

I declare that this thesis is my own work and it has not been previously submitted, either by me or by anyone else, for a degree or other qualification at any educational institute, school or university.

Jamal M. Al-Tuwaijari

Contents

1 Introduction **1**

1.1 Context and Motivations 1

1.2 The Aims 4

1.3 The Challenges 5

1.4 Contributions of the Thesis 5

1.4.1 Main Research Contribution 5

1.4.2 Publications Arising From This Research 7

1.5 Thesis Outline 7

2 Business Process and Workflow Systems: An Overview **9**

2.1 Business Process 10

2.1.1 Process Definition 10

2.1.2 Business Process Definition 11

2.1.3 Business process elements 14

2.2 Business Process Modelling 17

2.2.1 Resources modelling 22

2.3 Business Process Analysis and Optimisation 23

2.3.1 Business process optimisation 29

2.3.1.1 Resources scheduling 30

2.4 Quantitative aspects of business processes 32

2.4.1 Techniques of quantitative analysis 33

2.5 Stochastic Petri net 34

2.6 Summary 39

3 Generic Framework for Business Process Modelling and Analysis **41**

3.1 Introduction 41

3.2 Fundamentals of the Generic Framework 42

3.2.1 Preliminary Information 42
3.2.2 Framework requirements 43
3.2.3 Characteristics of the framework 44
3.2.4 Basic assumptions of the framework 45
3.3 Design of the Framework 45
3.3.1 Business process specification module 46
3.3.2 Stochastic Petri net module 48
3.3.2.1 Software tools for building SPN models 48
3.3.2.2 Mapping business process into SPN 49
3.3.3 Algorithm module 50
3.4 The framework from the user's perspective 51
3.5 The framework from the perspective of the support tools designer . . 52
3.6 Summary 54

4 Business Process Workflow with Resources (MWF-wR) Model 55

4.1 Introduction 55
4.2 The Fundamentals of Business Process Workflow Modelling 56
4.2.1 Workflow model and the *MWF-wR* formalism 56
4.2.2 Workflow characteristics and *MWF-wR* 57
4.2.3 Workflow structure 58
4.3 Requirements of the *MWF-wR* Formalism 60
4.4 Formal Definition of Modelling Workflow with Resources (*MWF-wR*) 61
4.5 Mapping the *MWF-wR* Formalism into Stochastic Petri Nets 64
4.5.1 The *MWF2SPN* algorithm for mapping *MWF-wR* into *SPN* . 65
4.6 Mapping *MWF-wR* components into SPN 68
4.7 Summary 72

5 Software Tool Architecture 74

5.1 Introduction 74
5.2 Requirements of Support Tool for Framework Design 75
5.2.1 Functional requirements 75
5.2.2 Non-functional requirements 76
5.3 Assumption 77
5.4 Design Components 77

5.5 Implementation 82
5.5.1 Implementation requirements 83
5.5.2 The implementation of the components 83
5.5.2.1 Mapping *MWF-wR* model into *SPN* 83
5.5.3 Mapping the business process model into Matlab 89
5.6 Summary 94

6 Experimental Results 95
6.1 Introduction 95
6.2 Enhancement of Availability of Resources 96
6.2.1 Availability weak point analysis over an SOA deployment scenario 96
6.3 The Framework of Weak Point Analysis Methodology 99
6.3.1 The *MWF-wR* business process specification module 100
6.3.2 Stochastic Petri nets module 101
6.3.3 Availability weak point analysis algorithm module 103
6.4 Business process workflow scheduling 113
6.4.1 Grid workflow scheduling in WOSE Scenario 114
6.5 The framework for Business Process Workflow Scheduling 115
6.5.1 The *MWF-wR* business process specification for workflow scheduling module 116
6.5.2 Stochastic Petri Nets Module 117
6.5.3 Business process workflow scheduling module 119
6.6 Summary 131

7 Conclusions and Future Work 132
7.1 Summary of the Research 132
7.2 Summary of Contributions 133
7.3 Limitations of the Research 135
7.4 Suggestions for Future Work 135

References 137

List of Figures

1.1 Three main modules of the generic framework 6

2.1 Business process features 13

2.2 A three dimensional view of a workflow modelling from [50] 25

2.3 Types of business processes analysis from [63] 28

2.4 A graphical Petri Nets Example 36

3.1 The basic modules of the generic framework 46

3.2 Business process specification module 46

3.3 Stochastic Petri net module 48

3.4 Algorithm module 50

3.5 The generic framework from a user perspective 51

3.6 The framework from tool designer's perspective 53

4.1 Example of workflows graph 57

4.2 Basic workflow routings 59

4.3 The $MWF - wR$ task in Petri net 68

4.4 Task and resource of $MWF - wR$ in Petri net 69

4.5 Multiple-resources of $MWF - wR$ in Petri net 70

4.6 SPN model of example 4.1 $MWF - wR$ model 71

5.1 Framework component from engineering perspective 78

5.2 Framework components from support tool designer perspective 78

5.3 Software support components form modeler perspective 82

5.4 The framework implementation components 84

6.1 Framework for weak point analysis methodology 100

6.2 Framework for business process workflow scheduling 115

List of Tables

2.1 Business Process Definitions . . . 11
2.2 Performance metrics of business processes . . . 32

3.1 Example of SPN file elements . . . 50

4.1 MWF2SPN algorithm for mapping *MWF-wR* to *SPN* . . . 67

5.1 MWF-wR file . . . 86
5.2 *CSPL* file of mapping *MWF-wR* task . . . 87
5.3 Format for mapping *MWF-wR* resource into *CSPL* file . . . 88
5.4 SPN2MATLAB algorithm for mapping *CSPL* to Matlab . . . 90
5.5 CSPL file of Example 4.1 . . . 91

6.1 Workflow-resource relationship matrix . . . 97
6.2 Workflow-resource relationship matrix . . . 98
6.3 Component failure and costs. . . . 98
6.4 MWF-wR file for weak point analysis . . . 101
6.5 High availability weak point analysis algorithm . . . 104
6.6 Resource to workflow mapping . . . 105
6.7 Component failure and costs. . . . 106
6.8 Resource components . . . 106
6.9 Components of resources availability . . . 106
6.10 Resources availability . . . 106
6.11 Resources cost . . . 107
6.12 Workflow availability . . . 107
6.13 The file of the example of high availability . . . 114
6.14 The parameters used by the workflow scheduling algorithm . . . 116
6.15 Scheduling workflow algorithm . . . 120

6.16 The parameters of the example for workflow scheduling 122
6.17 The selected parameters for workflow scheduling example 123
6.18 The file of workflow scheduling example 131

Abbreviations

BAM	Business Activity Monitoring
BEA	Bureau of Economic Analysis
BPEL	Business Process Execution Language
BPI	Business Process Improvement
BPM	Business Process Modelling
BPMG	Business Process Management
BPMI	Business Process Management Initiative
BPML	Business Process Modelling Language
BPMN	Business Process Modelling Notation
CPM	Critical Path Method
CSPL	C-based Stochastic Petri Net Language
CTMC	Continuous Time Markov Chains
DEDS	Discrete Event Dynamic System
GERT	Graphical Evaluation and Review Technique
GUI	Graphical User Interface
ILP	Integer Linear Programming
IT	Information Technology
MILP	Mixed Integer Linear Programming
MRM	Markov Reward Model
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
MWF-wR	Modelling Workflow with Resource
PDF	Probability Distributed Function
PERT	Program Evaluation and Review Technique
PN	Petri Net
QoS	Quality of Services
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SPN	Stochastic Petri Net
SPNP	Stochastic Petri Net Package
OOPL	Object Oriented Programming Language
WFMS	Workflow Management Systems
XML	Extensible Mark-up Language
YAWL	Yet Another Workflow Language

Chapter 1

Introduction

This chapter presents the context of and motivation for this work. Then, the aims of the research and challenges faced are stated. Furthermore, the main contributions of the research are described and finally an overview of the structure of this thesis is presented.

1.1 Context and Motivations

Today it is generally accepted that business processes are the backbone of organisations and that their design is central in automating and increasing the efficiency and effectiveness of the production and delivery of services and products. Business processes take place at all levels of the organisation's activities and include events that can be seen and those which are invisible to the customer [1]. The rapid growth of the Internet and other Web based technologies has created a ”faster, cheaper, and smart” environment, and presents organisation with both opportunities and challenges [2]. A dynamic economy, altered customer requirements, and the emergence of electronic commerce have sparked competition between organisations. With the aim to be the winners in this competition, the majority of companies now pay more attention to quality and customer satisfaction in order to gain more profit [3]. To achieve this aim, companies have to make quick decisions, transfer information rapidly with shorter cycles times [4]. These goals can be accomplished by adopting a precise assessment of the performance and optimisation of their business processes.

Business processes can be seen as a structured flow of business activities. The workflows of the critical administrative and business functions in an organisations can be described and managed and the use of available resources can be optimised. Business processes have been defined by Smith and Fingar [5] as "the complete and dynamically coordinated set of collaborative and transactional activities that deliver value to customers". It is crucial for the organisation to manage and analyse their business processes and their structure in order to identify their critical success factors and to improve such factors while eliminating their weaknesses.

Business process redesign, business process improvement and business activity monitoring have gained a lot of attention in business process management,[6, 7]. Adopting analysis techniques to provide precise evaluation of business processes is an important activity for organizations in order to allow IT managers to balance the operational and economic costs of business processes and the resources used. Business processes comprise several components that combine and interact with each other to achieve the required objectives. In the context of this thesis, identifying these elements and their relationship is the basis for providing successful performance analysis and optimisation of business processes. This is because of the interactive relationship between the different components where the failure of one component in achieving the required level of service affects the overall performance of the system.

As organisations become larger and conduct more complicated tasks, the organisation of its work becomes more and more complicated. This calls for computerised information systems to support the management and analysis of business processes, and the use of information technology to achieve business objectives [8]. Workflow technology has emerged as a way to facilitate meeting these requirements. The workflow is the automation of the messaging processes among several business processes, and the business process is automated in such a way that data and tasks are passed between human and machines to achieve the overall business goals according to predefined sets of rules [9]. Organisations should therefore evaluate their business processes precisely in a sufficiently automated and rapid way. The analysis and optimisation of business process performance is the purpose of techniques used in this thesis. For instance, analysing the availability and scheduling of available resources and throughput calculations are examples of techniques used for improving and optimising business processes. However, it is important for organisations to perform these techniques in an automated and effective manner if they are to lead to efficient performance and greater profit.

The analysis of business process performance has a variety of uses. The following reasons prompted organisations to pay more attention with regard to business process modelling and analysis [10]:

- To monitor and control
- To maximise the effectiveness of improvement efforts
- To achieve alignment with organisational goals and objectives
- To reward and to discipline

Quantitative methodologies for business processes evaluation are needed in order to understand, model, design, analyse, and operate such large infrastructures. Quantitative measures are used to estimate the quality of service. The cost of business processes typically depends on their behaviour over time, due to time-cost dependency factors. In addition, factors such as customer satisfaction, which cannot be simply quantified, also depend on quantitative measures [11]. Modelling business processes for quantitative evaluation and to produce quantitative results from models in an automated way allows closer scrutiny than purely qualitative analysis. It helps decision makers conduct evaluations and make decisions [12]. In addition, it helps organisations to improve and optimise the performance of their business process by identifying weaknesses and determining the existence of bottlenecks in the system.

In the Internet age, quantitative performance analysis has started to receive special attention in business process management systems, because the delivery of the right level of quality of services (QoS) needs quantitative evaluation which is crucial for organisations and customers [13]. Moreover, the cornerstone of successful business process management is a continual quantitative evaluation through monitoring the quality of service and other business processes performance metrics [14]. Most of the studies in the literature aim at modeling verification by proposing some kind of mapping from business process modelling techniques into formal languages. However, very few published studies try to use models for the analysis of the quantitative behaviour and optimisation of business processes [11]. This has encouraged us to seek a new and effective methodology to support quantitative evaluation.

The analysis and optimisation of business process performance may be based on different approaches depending on application, context and the objective of analysis. Each approach uses different types of solution or optimisation algorithms and parameters. The limitation in the area of business process quantitative evaluation is lack of generic methodology that can be used for all solution techniques and in such a that each algorithm can be applied to as broad as possible a class of business processes. Therefore, the design of a generic modelling framework for quantitative evaluation requires a good understanding of the different types of algorithms for the analysis of business process performance.

Many specification languages and techniques have been introduced for modelling business processes, such as business process modelling language [15], business process modelling notation [16], and business process execution language for web services (BPEL4WS or BPEL, in short) [17]. These languages aim to facilitate the construction of business process workflows, define the associated behaviour and specify the order in which activities are performed. However, although these modelling languages are able to support business process users in different phases of business process management, they are not well-suited models for quantitative evaluation. Therefore, this study presents a different generic business process modelling framework that is able to represent any business process and support performance analysis approaches. The framework has the ability to apply different types of algorithms used for performance analysis and optimisation to as wide a class of business processes as possible.

1.2 The Aims

The main aim of this research is to expand the applicability of performance evaluation and optimisation algorithms to as wide a class of business processes as possible.

1.3 The Challenges

Providing a generic framework for a wide variety of quantitative evaluation and optimisation algorithms is not an easy task since it needs to adopt modelling methodologies, solution algorithms and software tools to support this evaluation. To fulfill these requirements, the research faces numerous challenges, the most significant of which are as follows:

- The lack of unified formal definitions of business process components. Instead many definitions exist, which makes it difficult to identify a generic model that covers all types evaluation and optimisation algorithms. Moreover, there are many modelling techniques applied to business process, but so far none of them focusses business processes model- based quantitative evaluation.

- Adding quantitative information to a business process model and representing it in a generic format is very complicated because numerous algorithms are used for business process performance analysis and optimisation. So, it is hard to define a generic and flexible framework suitable for a large set of algorithms.

- Developing software support tools for the automated implementation of a framework is also not easy since the framework consists of three main modules and incompatibilities between the formats of files in each module must be overcome. This is because each module may use different file formats and combining these module requires specific mapping relations between them.

1.4 Contributions of the Thesis

1.4.1 Main Research Contribution

While recent active research efforts into the quantitative evaluation of business process workflows have developed various methods, including those focusing on performance analysis and optimisation, many issues remain in this domain. Of particular importance are the generic modelling approach used for conducting the performance analysis and optimisation of business processes in an effective methodology, and the software support tools used for this.

The main contribution of this thesis is to introduce a methodology that systematically expands the application of a variety of quantitative evaluation and optimisation algorithms to as large a class of business processes as possible. This result in the following concrete contributions:

- The development of a generic modelling methodology that allows a large number of quantitative evaluation and optimisation algorithms to be applied to as large a class of business processes as possible. The framework developed consists of three main modules: a business process specification module, a stochastic Petri net module and an algorithms module, as shown in Figure 1.1. Each module defines a phase in the framework. The methodology proposed in this work is different from existing methods in the literature, in that software support tools are developed to map business process models into a stochastic Petri net (SPN) model so that automated performance analysis and optimisation can be conducted.

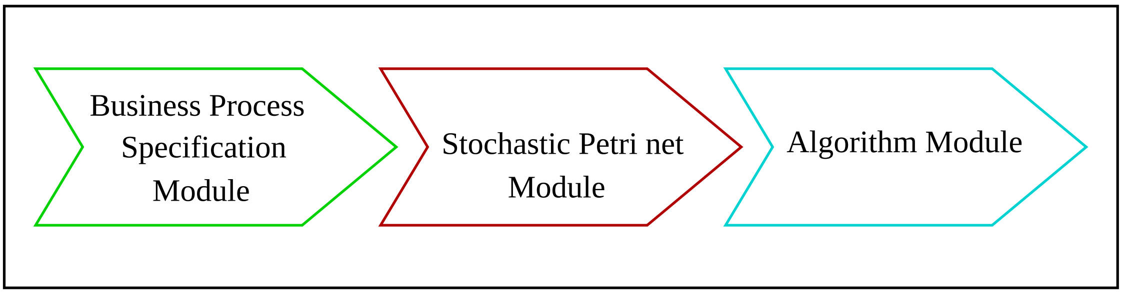


Figure 1.1: Three main modules of the generic framework

- The development of a novel modelling formalism for business processes to support the above methodology. The work develops the Modelling Workflow with Resource (*MWF-wR*) formalism, which can model any business process and introduce solutions for a set of improvement and optimisation metrics.

- The design and implementation of software support tools using Java as object oriented programming language for the implementation of the framework. The software support tools developed are used to automate the processing of the three main modules. This includes the design of user friendly interfaces, mapping relations between different formats in the three modules, and computing algorithms used for quantitative evaluation. These software tools help in and facilitate the automation of the mapping relation between modules and the conducting of the quantitative evaluation.

- The application of the generic automated quantitative evaluation of business process workflows in detailed case studies. The experimental results for real applications were also tested using the proposed quantitative framework to demonstrate its capability to plug in two analysis and optimisation algorithms. To demonstrate the applicability of the approach and the support tools, two case studies are presented. These two cases studies conduct high availability of business processes and resource scheduling of business process workflow.

1.4.2 Publications Arising From This Research

This work has been documented in part in the following publications:

1. J. Al-Tuwaijari and Aad van Moorsel, "Automated Generic Weak Point Analysis of Business Processes". In *10th* UK Performance Engineering Workshop,2010. [18]
   This work introduces the idea of our approach for the automated evaluation of business processes and presents a study of automated high availability enhancement.

2. J. Al-Tuwaijari and Aad van Moorsel, "Generic Business Process Model for Quantitative Evaluation". In Poster Session at The Newcastle Connection 2012, UK, Augest, 2012. [19]
   This work extended the approach to a more generic version and a mini paper and a poster titled "Generic Business Process Modelling for Quantitative Evaluation" was presented.

1.5 Thesis Outline

The thesis is organised as follows:

- Chapter 2 draws upon background information on concepts that are utilised throughout the research. These include the fundamental concepts and definitions of business processes and workflow, formal languages for business process modelling, analysis and optimization and the tools used for these purposes.

- Chapter 3 presents the generic framework for modelling business process for quantitative evaluation. Explanations of the architecture and the components of the framework are introduced. The three main modules of the framework are then discussed, and the components and functions of each module are illustrated. The chapter also explains the mapping relations between these module to produce the functionality of the framework.

- Chapter 4 describes the proposed generic formalism for modelling business processes. A novel *MWF-wR* formalism is presented for modelling any business process. The requirements, descriptions of components of the formalism and formal definition are introduced. In addition, an example of business processes defined in the formalism is presented.

- Chapter 5 proposes the guidelines for developing software support tools using Java as object oriented programming language which able to automate the functions of the framework. This include the formats and formalism used to map relations among modules and the tools used to facilitate the implementation of the framework. The chapter also describes the *CSPL* file of *SPNP*, which is used as a modelling tool to represent business processes for quantitative evaluation.

- Chapter 6 introduces the experimental results to demonstrate the applicability of the proposed approach. Two examples are presented, the first of which is an experimental study is used to automatically solve the problem of weak point analysis [20]. The second experimental study is used the framework to automatically solve the problem of scheduling workflows [21].

- Chapter 7 concludes the thesis, and summarises the achievements, contributions and problems of this research. Additionally, possible extensions of the research are suggested to help in the efficient performance analysis and optimisation of business processes.

Chapter 2

Business Process and Workflow Systems: An Overview

The design, analysis and management of business processes are crucial activities for organisations aiming to effectively compete in today's volatile business environment [22]. Performance analysis and optimisation and the continuous improvement of business processes are key for organisations to be successful. In addition, they can establish competitive advantage by improving quality and efficiency and reducing costs. This chapter introduces the general concepts and definitions of business processes and workflows and discusses the modelling, analysis and optimisation techniques which are particularly relevant to this research. Gaps in the literature concerning the quantitative evaluation of business processes are then highlighted. This will show the importance of and the motivation for the present research.

The chapter is organized as follows: Section 2.1 defines the basic terms used in the analysis of business processes and their elements. In Section 2.2, modelling and classification issues in the context of business process analysis are presented. Section 2.3 introduces analysis and optimisation to different perspectives on the analysis and optimisation of business process, and the methods used for these purposes. Section 2.4 then illustrates the quantitative aspects of business processes that relevant to the research. Section 2.5 describes stochastic Petri net where it is used as a tool for mapping business process model into SPN model. Finally, Section 2.6 summarises this chapter.

2.1 Business Process

This section introduces the concept of the business process, which is the central focus of this research. The execution of a business process involves real people, materials, clients, machines and computers, and delivers one or more actual products. In this sense, its execution is the actual manifestation of a business process. The following sections discuss the various definitions of business processes used in the literature and presents the main concepts involved. The various elements of business processes are then detailed, and current issues in the field are discussed.

2.1.1 Process Definition

The term business process has received wide acceptance. Business operations are considered as a process, and the main features of the process are reflected in particular business functions. Many definitions of such a process have been presented in the literature. A process has been defined in general term as a set of partially ordered steps intended to reach a goal [23]. Another important definition of a process was formulated by Grover [24]: where a process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs. According to M. Havey [25] a process involves movement, time and work and actions are performed over intervals of time to achieve, or make improvements to, some objective.

J. Li et al. [26] presented a detailed definition of a process and its elements. This includes the participating activities, for instance, the main participants, information about the individual activities, and associated IT applications and data. Platt [27] defined the nature of the process as "the transformation of something from one state to another state through partially coordinated agents, with the purpose of achieving certain goals that are derived from the responsibility of the process owner".

Finally, Medina-Mora et al. [28] classified processes in an organisation in three categories of material processes, information processes, and business processes. They specified that business processes are market-centred descriptions of an organisation's activities, implemented as information processes and/or material processes. These definitions of process imply the main attributes possessed by most processes.

2.1.2 Business Process Definition

Table 2.1: Business Process Definitions

Business Process Definitions	Author(s) & Year	Ref.
The logical organisation of people,materials,energy,equipment and procedures into work activities designed to produce a specified end result (work product)	Pall (1987)	[31]
A collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer. A business process has a goal and is affected by the event occurring in the external word or in other processes	Hammer & Champy (1993)	[32]
A set of partially order process steps,with set of related artifacts,human and computerised resources,organisational structures and constraints,intended to produce and maintain the requested software deliverables	Lonchamp (1993)	[37]
Group of related tasks that together create value for a customer	Hewitt (1995)	[33]
A related group of steps or activities that use people,information and other resources to create value for internal or external customers.The steps are related in time and place,have a beginning and end, and have inputs and outputs	Alter (1996)	[42]
A business process is a set of inter related work activities characterised by specific inputs and values added tasks that produce specific outputs	Saxena (1996)	[40]
An approach for converting inputs to outputs. It is the way in which all the resources of an organisation are used in a reliable, repeatable and consistent way to achieve its goals	Zairi (1997)	[41]
A business process consists of activities orderd in a structured way with the purpose of providing valuable results to the customer	Agerf.et.al (1999)	[38]
A business process is a set of one or more linked procedures or activities that collectively realise a business objective or policy goal,normaly within context of an organisational structure defining functional roles and relationships	Fan (2001)	[39]
A specific ordering of work activities across time and place,with a beginning, an end, and clearly identified inputs and outputs	Irani et al (2002)	[35]
A process is a completely closed, timely and logical sequence of activities which are required to work on a process−oriented business object	Becker & Kugeler (2003)	[36]
The term "business process" refers to a conceptual way of organising work and resources	Reijers (2003)	[34]
Step−by−step rules specific to the resolution of a business problem	Havey (2005)	[25]

The first explicit study of business processes was conducted by Fredrick Taylor in 1911, concerning the design of work procedures to improve efficiency and effectiveness [29]. However, many researchers and authors have proposed comprehensive definitions of business processes. Different definitions and approaches originate from different areas of study. Business processes can be defined from different viewpoints, each dealing with a particular aspect of business process. For example, the functional view deals with process activities; whereas the organisational view considers who will perform the activities and where. The informational view on the other hand focuses on informational entities; and finally behavioural views specify when and how the activities involved are performed [30]. This section discusses the various existing definitions of business processes. The most important are summarised and then these are distilled into a definition that is compatible with the purpose of this research, and compatible with its objectives. The aim is to use a general definition that takes into account the quantitative aspects of business processes which are of interest in this work.

Table 2.1 demonstrates the most common definitions of business processes that have appeared in the literature. The differences between them are then specified and considered in the light of the present work. In 1987 Pall [31] defined a business process as “the logical organisation of people, materials, energy, equipment and procedures into work activities designed to produce a specified end result (work product)”. Another definition of the business process was presented by Havey [25], who described it as the ”step-by-step rules specific to the resolution of a business problem”. According to Hammer and Champy [32], a business process is “a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes”. Meanwhile Hewitt [33] defines a business process as a group of related tasks that together create value for a customer.

According to Reijers [34], “the term ’business process’ refers to a conceptual way of organising work and resources. In this sense, a business process is not tangible. However, product instances are produced by executing or instantiating the business process”. On the other hand, Irani et al. [35] provided a definition which depicts the structure of the business process as “a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs”. These and other prominent definitions [36, 37, 38, 39, 40, 41, 42] of a business process

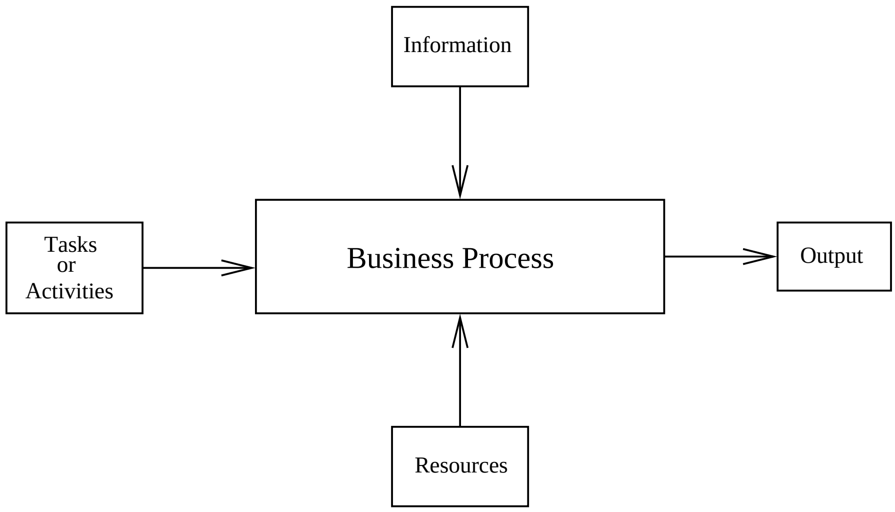


Figure 2.1: Business process features

are presented in Table 2.1, considering business processes from different perspectives.

It is apparent from an analysis of the above definitions that very similar concepts are used to describe business processes. Common elements can be identified in the majority of definitions. The business process is said to consist of elements related to process input, the process itself, the transformation of inputs or the workflow, human and computerised resources, and outputs. The output is designed to achieve the organisation's goal and objectives or to produce value for a customer. Despite the existence of these different definitions, four key features of any business process can be identified, which gives a clear picture of what business processes are. These are depicted in Figure 2.1 and include:

- specific definable inputs
- a set of clearly definable tasks or activities, performed in some order
- computerised and human resources
- specific outcomes or results which create value of some kind for the customer

However, despite the existence of these common elements in most definitions of business processes, there is no consensus. Some definitions have been criticised for not focusing on business components and not adequately distinguishing between manufacturing or production processes. It has been claimed [43] that the majority of definitions of the business process are limited in depth and therefore any models arising from them will also be limited, having been formulated in a purely mechanical way. Moreover, Peer and Brigitte [44] reported that no generally accepted definition of the term business process exists because of the fact that business processes have been considered in a number of very different disciplines. It can be assumed that the activities of the flow are executed repeatedly, so they considered this flow as the basis of a business process. The range of different definitions and criticisms of them as well as the lack of consensus on a general definition of business processes indicates the importance of providing a generic framework for the modelling of business processes, and at the same time indicates the difficulty of designing such a framework.

For the purpose of this research, the definition of a business process should include all essential aspects and elements. Therefore, the following definition of business process is adopted throughout this research. A business process is a set of related tasks that are carried out using a set of resources such as machines, software applications, humans or others to perform a business operation in order to produce value for the organisation.

2.1.3 Business process elements

To model and analyse the performance of business processes, it is essential to understand and identify their various elements and parts [45]. Despite a wide variety of definition a common ground between them can be used to determine the structural elements of business processes and the relationships between them. Such common elements can be considered to be the basic elements involved in nearly all definition of a business process. For instance, van der Aalst [46] reported that two necessary elements need to be specified in order to fully define a business process: these are the activities which describe the partially ordered sets of tasks, and the resources used to process tasks. The term tasks tends to be synonym for activities, which are the smallest elements used in the analysis of business processes [47]. Accordingly, activities and resources are considered as the two basic elements of business processes [46]. Activities or tasks are considered to be the basic elements used in achieving the goals of business processes which exploit inputs to deliver desired outcomes. On the other hand, resources are the most important elements of the activities or tasks to be executed. The two basic elements of business processes, activities/tasks and resources, are discussed in more detail in what follows, and the various issues related to these two elements as described in the literature are then identified.

1. **Activities or Tasks:**

   There are two main trends in the literature regarding the understanding of activities. The first tends to use the term activity without specifying any internal structure, so that activities cannot be decomposed. The second tends to define activities in terms of their internal structure. Many authors use the terms activity and task as equivalent in the context of business processes. In any case, activities or tasks are seen as the executable parts of the business process which utilise resources as basic elements in their execution. In this study the terms tasks and activities are used as equivalents in the context of business processes. The following discussion considers the opinions in the literature concerning tasks.

   Wherever a definition of business processes is found, activities or tasks will be considered as essential elements. There are many perspectives on the nature of activities and tasks in the context of business processes, and numerous definitions have been presented. These depend on the author's perspective concerning business processes and the details required. Basten and Van der Aalst [48] reported that "activities are assumed to be atomic entities without internal structure". This definition contrasts with the perception of Stohr and Zhao [49] that an "activity is a discrete process step performed either by a machine or human agent. An activity may consist of one or more tasks". Meanwhile the definition presented by Van der Aalst is a more simple definition of an activity as a transaction: "One can think of a task as a logical unit of work and an activity as a transaction" [50].

   Van der Aalst [46] has also provided a comprehensive definition which identifies tasks as representing the internal structure of activities. He proposed that "business processes are centered around activities. Each activity specifies the set and the order of tasks to be executed in order to achieve the business process goal". In the same context, Stohr and Zhao [49] proposed that an activity may consist of one or more tasks. Orman [47] stated that an activity can be further decomposed into tasks. In a similar vein, Hajo [34] argued that the smallest distinguishable part of a process is often referred to as a task, and he proposed that a "task is a specification of a part of work within a business process". In this study a task is consider as the essential element of the work required to achieve the goal of the business process and must be included in any model or analysis of business processes.

2. **Resources:**

Resources are the second fundamental element of business processes which are used by tasks and transformed so as to create the output of the process. Business processes are usually defined in the context specific organisations which utilise resources to complete their tasks. Resource management has been recognised as an important aspect of workflow management systems. Generally, resources in a business process are agents such as persons, machines, software or applications, that execute tasks. There are many definitions of resources in the literature. Hajo [34] stated that the product of a business process is delivered by the commitment of resources, and "a resource is a generic term for all means that are required to produce a product within the settings of a business process". Biazzo [51] presented a generic definition of resources as including everything that is either used or modified by tasks. In other contexts Van der Aalst and van Hee [52] and Li et al. [26] considered a resource as any human and/or machine which contributes to the execution of activities. Similar opinions were expressed by Castellanos et al. [53], who claimed that resources execute the activity, and they are usualy humans or machines.

There are several views concerning the classification of resources. Hajo [34] characterised them as consumable and reusable. The former being consumed when they are applied, and the later can be committed for a long period of time. Moreover, he specified two main dimensions used to define resource classes as functional and organisational. A resource class based on functional characteristics is known as a role, function or qualification. An organisationally oriented resource class is based on criteria used to recognise different parts of an organisation; for instance, departmental, geographic, or product divisions.

In accordance with the above review of definitions of tasks and resources, it can be argued that they are two basic elements of any business process. Tasks are transformational steps which use resources as inputs and produce new ones as outputs. Resources are physical or information objects. Both concepts constitute essential elements in the modelling and analysis of business processes and the relationships between them determine performance. The work in this thesis aims to propose a modelling framework to facilitate the quantitative evaluation of business processes including these two elements. The way in which resources are allocated to tasks, and how and when the resources are used to complete tasks is the main objective of the performance evaluation and optimisation of business processes. Therefore, specifying the relationship between these two elements in an effective and easy way is one of the major aims of this work.

2.2 Business Process Modelling

This section discusses the modelling of business process and existing techniques used for such modelling in the literature. Business process modelling can be considered as the first and the most important step in business process management [54, 55]. For the purposes of this research, business process models are required which provide the means for quantitative analysis in order to select the appropriate performance measures and obtain realistic knowledge about organisational efficiency.

A model, generally speaking, as defined by Wilson [56] is “the explicit interpretation of a situation, or one idea about that situation. It can be expressed in mathematics, symbols or words, but is essentially a description of entities, processes or attributes and the relationships between them”. Another definition of models proposed by Beer et al. [57] is that “all models, whatever their purpose, have one common feature, which is the mapping of elements in the system modelled”.

Business process modelling enables a general understanding and analysis of business operations, and therefore business process models can contribute to the analysis of and integration of activities in an organisation [58]. Successful business process modelling relies on an adequate view of the nature of business processes and their elements. So, it is clear that the objective of business process modelling is to acquire explicit knowledge about the business processes including quantitative aspects.

Business process modeling has increased the ability to understand business operations and to make rational decisions about organising activities in a measurable and understandable way [59]. It can be seen as a collection of techniques, methodologies and tools used to support the analysis and improvement of business processes. There are many methods, techniques and tools used in modelling business processes. Kettinger et al [60] reviewed 25 such methodologies, 72 techniques and 102 tools in the context of business process modelling. The various techniques vary extensively in the degree to which they provide the ability to model different business process perspectives. Some techniques focus mainly on functions, others on data, and yet others on roles.

However, Kueng et al. [61] recognised several important categories of approaches to the modelling of business processes, as follows:

- Activity-oriented approaches: these define a business process as a specific ordering of tasks which offer support for process models.
- Object-oriented approaches: these use the principle of object orientation to business process modelling, such as specialization, encapsulation and inheritance.
- Role-oriented approaches: where a role is involvement in a set of activities conducted to perform a specific job.
- Speech act-oriented approaches: these observe the communication process as a four-phased loop: proposal, agreement, performance, and satisfaction.

In this study it is necessary to define a framework for modelling business processes in term of processes, tasks, resources and the interactions between them in order to facilitate quantitative evaluation. So, the activities involved in the business process and the resources used need to be modeled and the relationships between them determined in a new way. The activity-oriented, Object-oriented and role-oriented perspectives are therefore the most important approaches to be consider.

Business process modelling includes the modelling of activities or tasks and their temporal and causal relationships, in addition to the resources used to complete these tasks according to specific business rules. According to Aguilar-Saven [58], "business process models are used to learn about the process, to make decisions on the process, or to develop business process software. For each of these purposes particular business process models are better suited depending on their particular constructs." Many different classifications of business process modelling can be found in the literature, each of which provides classification according to a specific perspective. Aguilar-Saven [58] classified business processes according to two perspectives. The first focuses on the purposes of the business process model used and whether it is used for the description of learning, to support decision making in process design or development, process execution, or to allow information technology (IT) enactment support. The second perspective concerns whether the model is active in allowing the user to interact with it, or passive in not allowing this facility.

Giaglis [62] proposed a different classification of business process and information system modelling techniques which depends on three variables of evaluation. The first variable is breadth, focusing on the goals of modelling, such as in process management, improvement, or development. The second variable is depth, concerned with functional, informational, behavioural and organisational issues. The third variable is fit, where lies in the suitability of projects with the technique.

An influential review of business process modelling by Melao and Pidd [45] differentiated between three perspectives towards understanding the nature of business processes. They then identified specific types of modelling techniques appropriate to each perspective. Firstly, deterministic machines perspective identifies a fixed sequence of activities or tasks which change inputs to outputs in order to achieve the business process objectives. The second perspective deals with the complex, dynamic and interactive features of business processes, whereas the final perspective is social construct which concerned with human issues in business processes where the people who enact the business processes have various roles and values. The final perspective on feedback.

It is clear from the above that there has been a lack of focus on modelling the quantitative aspects of business processes and theses has not been recognised as an important task. This shows how difficult it is to develop a generic framework for modelling quantitative aspects of business processes. However. there are many techniques and approaches used for modelling business processes. Each technique or approach captures different aspects of business processes and has its own benefits and drawbacks. The following are the three main types of business process modelling techniques which can be recognised in the literature:

- **Business process languages approach**:
  To easily model, describe and specify the structure of business process workflow, different formal specification languages have been introduced. The most popular are Business Process Modelling Language [15], Business Process Modelling Notation [16], and Business Process Execution Language for Web Services (BPEL4WS or BPEL, in short) [17]. According to Vergidis et. al [63], only simulation technique have been proposed explicitly in literature for business process models. The languages aim to facilitate the construction of business process workflows, define the behaviour and specify the order of activities or tasks. Although each business process language has its own semantics and syntax, executable languages are usually XML-based, and in this context BPEL and BPML are the most distinctive. According to Havey [25], BPEL is the most popular because it has gained the support of the International Business Machines (IBM) corporation, Microsoft and the Bureau of Economic Analysis (BEA). BPEL is a language for composing Web services and hence enabling business processes to be fully automated. The main characteristics of the language are as follows [17]:

  – It is an XML-based language.
  – It provides support for both executable and abstract business processes.
  – It is built on web services specifications
  – It is a language for orchestration composing web services
  – Its process is a web service in itself

  Business Process Modelling Language is an orchestration language similar to BPEL and developed by Business Process Management Initiative (BPMI). It is also an XML-based language. The BPML specifications provide an abstract model for expressing business processes and supporting entities. BPML defines a formal model for the expression of abstract and executable processes that address all aspects of business processes, including activities of varying complexity, transactions and their compensation, data management, concurrency, exception handling and operational semantics [15]. Another business process modelling approach is JBoss Business Process Management execution language named jPDL. This language facilitates business process development using visual tools [64]. Finally, Van der Aalst and ter Hofstede [65] developed Yet Another Workflow Language (YAWL) as a graphical process language based on Petri nets, which is built to support business process patterns. Despite the efficiency and effectiveness of these languages in modelling the structure, graphical and behaviour aspects of business processes, they can not directly used to model the quantitative aspects.

- **Formal/mathematical models:**
  In formal models process concepts are defined precisely, and mathematics is used in order to collect and analyse information about them. The most important properties of such models are that they can be verified mathematically, and checked for consistency [66]. Van der Aalst [67] proposed that business process models should have a formal foundation since, with precise definitions there is no room for confusion and raise the possibility for analysis. On the other hand, the process design has a lack of formal methods [68] as most of business process elements are of qualitative nature and it is difficult to represent them in a formal manner amenable to analytical methods [69].

  Despite the general lack of mathematical models in the literature, several approaches have been proposed. For example, Powell et al. [70] presented a mathematical model that can be subject to quantitative analysis. This model depicts the most important elements of a generic business process and suggests mathematical methods which could be used to measure and control business processes. Another mathematical model adopted by Hofacker et al. [68] defines a business process in terms of mathematical constraints and objective functions. Limitations of this model are that it cannot model complex constructs and can only model sequential processes. Formal mathematical approaches to modelling have received some criticisms. For instance, it is difficult to represent real-world business processes, especially when they include complex features such as, parallel or hierarchical flows and decision points. In addition, the use of mathematical notations is not easy for the business analyst because a lot of work is required to "create, and maintain a formal business process and retain its consistency" [66]. Formal modelling removes ambiguity from the specification of business processes and once they are described accurately, they can be verified mathematically, and quantitative information can be extracted by the use of suitable analytic tools.

- **Diagrammatic techniques:**
  Diagrammatic techniques such as flowcharts are the earlier technique used for modelling business processes and to depict them in a simple way, but the majority of such techniques do not use standard notation [25]. These techniques are widely used to visually depict the flow of a business process in a clear way and no technical expertise is needed. They involve informal representations of processes and do not have the structure necessary to support more complex and standardised constructs [71]. Moreover, these techniques lack quantitative information and their qualitative notation prevents further analysis and the development of analytic methods and tools [52].

2.2.1 Resources modelling

In the context of business processes, resources include who will do the job or be responsible for it [16]. Resource is any entity which is required to perform a task. It can be classified as a human or non-human entity [72]. Examples of non-human resources are plant, equipment, processors, memory or printer machines, or they can be virtual entities such as web services, software libraries or databases. The runtime of a task may be related to its resource requirements. An organisation can be defined as a formal grouping of human and non-human resources that conduct tasks pertaining to a common set of business objectives [72]. Human resources have a specic position within the organisation, and can be members of one or more units of the organisation. A resource may also have one or more associated roles. Roles are mechanisms for grouping resources that have similar job roles or responsibility levels. The resources may have features that describe their specific characteristics which can be useful when allocating tasks. These features can be stated as belonging to a group, or a specific individual, role, or position in an organization.

Non-human resources are classified in durable or consumable [72] in nature. Resources may be associated with a schedule, which is a list of tasks that a resource is committed to undertaking at specified future times, and with a history which is a list of tasks that a resource has completed at some time in the past. The following are the most important aspects of the modelling of resources:

1. A task in a business process may carried out by one or more resources.
2. Multiple tasks in the same business process may compete for access to resources at the same time.
3. A resource can perform more than one type of task.
4. The resources can cooperate together to perform a task.
5. A roles can be assigned to the resources used to execute the tasks.
6. Resources have a variety of values of cost in performing the tasks.

2.3 Business Process Analysis and Optimisation

This section discusses different views with respect to business process analysis and optimisation, and provides a classification of the current techniques used in this area. The better design of business processes results in improvements and cost savings, while badly designed business processes result in errors and resource ineffectiveness [73, 74]. The term business process analysis can be used with a wide variety of meanings, with the use of different types of techniques such as simulation, verification and performance analysis. The term performance also has many definition in the literature. One of these definition is "the way the organization carries its objectives into effect" [75]. The goal of business process performance analysis is to gain insight into operational processes, with the aim of improving and optimising them. In addition, it depicts the characteristics of business processes, identifies possible bottlenecks and resolves them, where parts of the process which might be improved can be determined.

Van der Aalst and Ter Hofstede [65] reported that business process analysis aims to investigate the properties of business processes that are neither obvious nor trivial. Irani et al. [35] pointed out a number of specific reasons for analysing business processes:

1. Identifying processes that facilitates communication among participants.

2. Specifying of the tasks required to achieve the transition from a current to a new process.

3. Recognising the sources of problems in existing processes, so as to help ensure that they are not repeated in the new process.

4. Understanding the current processes to provide a comparative measure of value for proposed changes.

There are many approaches to business process analysis, most of which are based on subjective rather than objective methods. According to Jonkers and Franken [11], hardly any studies can be found in the literature that explain the use of models to quantitative analysis and optimization of business processes.

Business process workflows can be described using the three dimensions [50] of case, resource and process, as shown in Figure 2.2. The case dimension denotes that a workflow consists of different cases processed individually. The resource dimension denotes that tasks in the workflow are carried out by organisational resources, and the process dimension denotes that workflow consist of a number of tasks in a particular routing order. Many approaches have been used to measure the performance of business processes, and each involves different metrics related to the three business process workflow dimensions cited above. Jansen-Vullersan et al. [76] recognised four main types of metrics for business process workflow performance concerning time, cost, quality and flexibility. Each dimension has its own specific metrics, which can be explained as follows.

1. **Time dimension:**

   Time is the fundamental measure of performance [77] and many performance measures have been derived for time dimension. For instance, lead time is related to case dimension and is defined as the time it takes to handle a case.

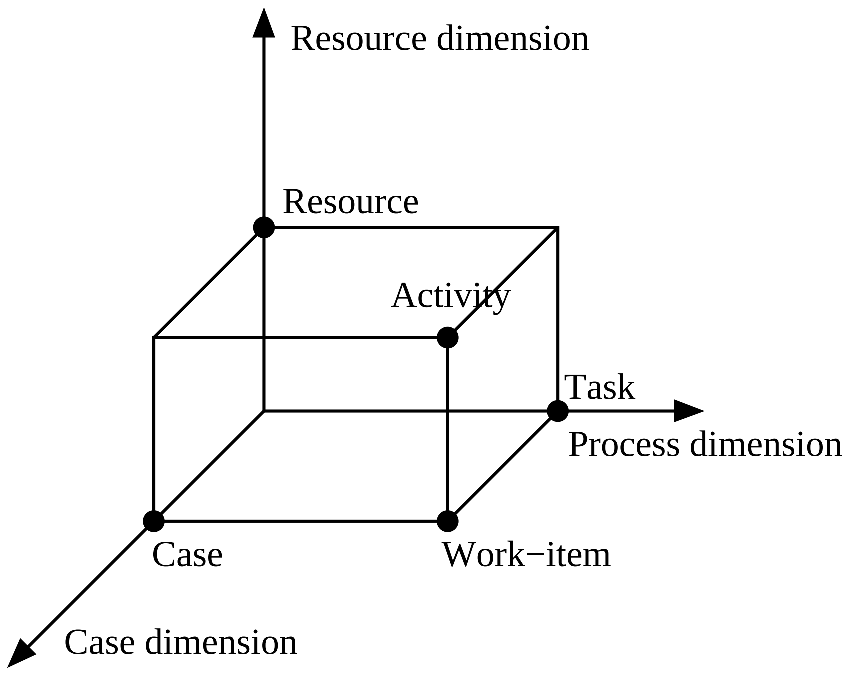


Figure 2.2: A three dimensional view of a workflow modelling from [50]

Throughput time is an other performance measure which denotes the time between the moment a task is completed and the moment the next task is completed [76]. Throughput time is itself composed of several sub-metrics:

(a) Queue time: the period of time that a task spends waiting for a resource to become available.

(b) Service time: the time resources spend handling the task.

(c) Move time: the time it takes to move a case between tasks.

(d) Wait time: all other delays related to a case; for example, the time waiting in a parallel branch to be able synchronise.

2. **Cost dimension:**
Cost is related to the three dimensions. For example, long lead times can result in a more costly process, and low quality can lead to expensive reworking, whereas low flexibility can also result in a more costly execution of the process [76]. However, there are many metrics related to cost, including resource usage, running costs, administration costs, and transport costs. However, costs can be classified as either variable or constant. Variable costs depend on the level of output, and constant costs are independent of production volume [77].

3. **Flexibility dimension**

Flexibility is defined as the degree of freedom that users have to make decisions about the execution of business processes [78]. Many flexibility metrics have been identified for tasks, resources and business process workflows as a whole. The following main metrics of flexibility performance were proposed by Jansen-Vullersan et al. [76]:

(a) Mixed flexibility refers to the ability to process cases, and is measured in terms of the number of cases that a task can handle. For resources it is defined as the number of cases that each resource can handle, and for workflows the number of cases that the workflow can handled.

(b) Labor flexibility is the ability to perform tasks. It is defined for resources as the number of executable tasks, and for workflows as the available resources per task per case.

(c) Routing flexibility is defined as the ability to process a case using multiple routes.

(d) Volume flexibility refers to the ability to handle changing volumes of input.

(e) Process modification flexibility concerns the ability to modify the process, including the number of sub-flows in the workflow, the complexity of the process, and the number of outsourced tasks.

Time, cost and range are different aspects of flexibility that can be considered for each of the metrics.

4. **Quality dimension:**

Quality covers subjective performance evaluations of business process in terms of either internal or external quality [79]. Internal quality is based on worker and external quality based on organisation or customer that initiates business process and receive the output. Internal quality can be define as the quality of a business process from an operator's perspective and is determined in terms of worker satisfaction. Many metrics have been used to assess internal quality. For instance, skill variety considers the number of different tasks and case types, whereas task significance denotes the degree that the job has an impact on the worker. Task identity is defined as the ratio of the number of executed tasks and the total number of tasks per workflow, while autonomy concerns the ratio of the number of authorized decisions and the total number of decisions. Co-worker relations refer to the quality of relationships between employees and their co-workers [79]. The internal quality of business processes cannot be measured directly because some metrics are determined and influenced by different factors depend on the type of process.

External quality concerns the measurement of client satisfaction, with two performance measures: the quality of output and the quality of the process. The quality of output includes performance, conformance and serviceability, and the quality of the process involves information about application, language simplification and information availability [79]. As with internal quality, the external quality of business processes cannot be measured as a whole only by performance metrics. The type of the process determines the metrics of the external quality.

The importance of performance metrics along the various dimensions depends on the techniques used or organisation which may have their own priority of performance metrics. For example, some organisations focus on quality of service to be more important than costs, while other focus more on costs rather than quality of service. All of the dimension mentioned above are considered to be important in the present study with more focus on the time dimension and the cost dimension as both are considered the core of techniques used for performance analysis and optimisation of business processes.

Business process analysis can be classified according to the modelling techniques used. Three formal types of business process analysis based on Petri nets and related to mathematical models technique proposed by Van der Aalst [80]. These three types of analysis concern validation, verification, and performance. Validation analysis tests the business process behaves, verification analysis set up the correctness of a business process, and performance analysis focuses on the quantitative evaluation of service levels, throughput times, resource utilisation and other quantitatively measurable factors.

Along the same lines as Van der Aalst [80], Li et al. [81] presented another classification of business process analysis. They propose that "workflow model analysis is conducted mainly at three levels, i.e., logical, temporal, and performance levels". Logical level concerns on the correctness verification of event dependency relations, resource allocation, and information utilisation. Temporal level focuses on interval dependency relations of workflow model with imposed timing constraints. Finally, the performance level focuses on evaluating the ability of workflow to meet requirements with respect to some key performance indicators such as, maximal parallelism, throughput, service levels, and sensitivity. The analysis of resource availability and utilisation and average turnaround times, for example, is performed at this level. However, despite, performance analysis of business process is the base of quantitative analysis of business processes, it has not captured the attention of many researchers until now [82].

Figure 2.3 shows another classification of types of business process analysis presented by Vergidis et. al. [63]. They propose three types of analysis each related to specific modelling techniques. The first is observational analysis which is applied in diagrammatic models. The second approach consists of performance evaluation, validation, and verification and all of these methods are applied in mathematical models, and finally the business process language modelling approach which includes algorithmic performance evaluation.

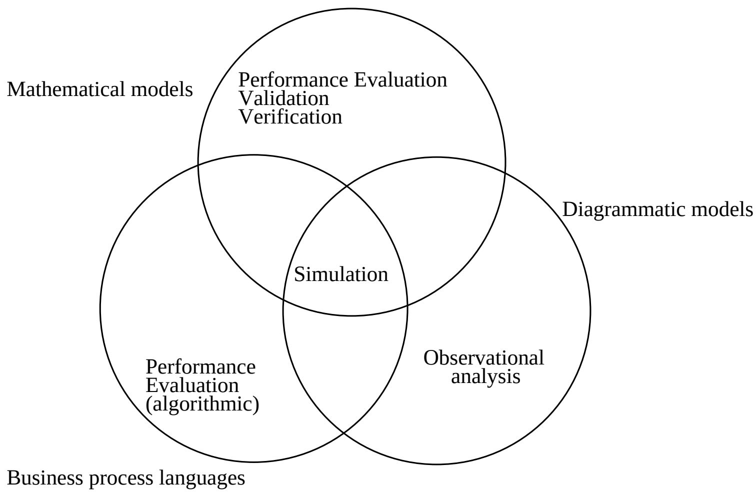


Figure 2.3: Types of business processes analysis from [63]

2.3.1 Business process optimisation

Optimisation is considered as a class of business process performance analysis. Zhou and Chen [83] reported that business process optimization aims at reducing lead-time and cost, improving the quality of products, and enhancing the degree of satisfaction among customer and personnel so that the competitive advantages of organisations can be maintained. However, despite the importance of optimisation, there is no systematic optimisation methodology for business processes [68]. Several approaches exist for analysis of business processes. Vergidis et al. [84] stated that business process optimisation is the automated improvement of business processes using prespecified quantitative measures of performance. They identified three main problems that obstacle business processes optimisation:

1. There is a lack of business process representation suitable for quantitative modelling. Most business process modelling techniques use visual diagrammatic approaches which do not allow quantitative analysis.

2. Methods of business process optimisation have been mostly manual and based on simplistic or isolated cases without generalisation capabilities.

3. There is no attempts in business process optimisation under multiple criteria.

Chen and Zhang [85] proposed an optimisation approach based on the used of algorithms to schedule large-scale business process workflows with various quality of service (QoS) parameters. This algorithm enables users to specify their own QoS preferences, and its objective is to find a solution that meets all QoS constraints and optimises the user-preferred QoS parameters. Van Hee et al. [86] proposed an algorithm to determine the optimal allocation of resources in order to maximise the time performance.

Kamrani et al. [87] presented an approach which aimed to optimise overall business processes by optimally assigning tasks to agents. Two main categories of processes, assignment-independent and assignment-dependent, are distinguished, and then each of these categories is divided into three types: deterministic, Markovian and non-Markovian processes. Two algorithms were introduced for these types of processes, one of which finds the optimal solution and the other is applicable to a large number of critical tasks provides a near-optimal solution.

According to Van der Aalst et al. [65], formal business process languages are associated with analytic techniques that can be used for investigating the properties of processes, but an optimisation approach based on executable process languages was not observed in literature. One of the most important technique related to business process optimisation is workflow scheduling. It considered as the successful optimisation approach in business process workflow area [88]. In the next section, the details of workflow scheduling is presented.

2.3.1.1 Resources scheduling

Resource scheduling is considered as the main approach used in business process optimisation and is important in this study where the framework is used for the automation of real workflow scheduling problem. Generally speaking, scheduling is a decision making process to determine when, where and how to produce a set of products, given specific requirements concerning a time horizon, a set of limited resources, and processing recipes [89]. In the context of business processes, Van der Aalst [88] defined the scheduling problem as the optimal allocation of scarce resources to tasks over time. Another definition presented by Senkul el al. [90] is that Workflow scheduling is the problem of finding a correct execution sequence for tasks over time while satisfying constraints within the model. Business process performance may vary in terms of time and cost depending on the workflow scheduling used. An effective methodology for workflow scheduling means that business process take less time, costs less and is of higher quality. This indicates the importance of workflow scheduling for organisations.

Floudas and Lin [91] reported that mathematical programming, and especially Mixed Integer Linear Programming, has become one of the most widely explored methods for scheduling problems, because of its rigour, flexibility and extensive modeling capability. Josef Kallrath [92] claimed that mixed integer optimization can provide a quantitative basis for decisions and allow complex problems to be coped with successfully as a useful technique to reduce costs and support other objectives.

There are many researches have been presented in literature concern workflow scheduling approach. Floudas and Lin [91] presented an overview of advances in MILP-based approaches to process scheduling problems. They proposed a classification of types of mathematical models and identified the strengths and limitations of each. Pinto and Grossmann [93] proposed a classification of scheduling problems and an overview of assignment and sequencing models used in scheduling techniques with mathematical programming . Yash et al. [21] presented a static optimisation workflow model using a stochastic programming technique to establish a Service Level Agreement (SLA) and satisfying the end-user workflow QoS requirements. Buyya and Tham [94] presented a cost-based workflow scheduling algorithm for time-critical applications. Their proposed scheduling algorithm develops a workflow schedule which minimises execution costs and meets the time constraints imposed by the user. Senkul et al. [95] presented a framework for workflows whose correctness is determined according to set of resource allocation constraints and develop techniques for scheduling such systems. Combi and Pozzi [96] focused on the temporal aspects of business process scheduling and presented conceptual organisational model for task assignment policies including time and skills of agents, deadlines for task completion, and other temporal aspect which improve the overall performances of scheduling activity of business processes. Brataas et al. (1997) consider the evaluating of the performance of workflows they proposed a framework involving both manual and automated activities. The lack of this framework is have not been adopted automation.

Several important observations can be made from the above survey of business process performance analysis and optimisation. Firstly, quantitative analysis has not received as much attention as qualitative analysis. Most existing modelling techniques use visual diagrammatic methods which cannot be used to conduct quantitative analysis. Business process performance analysis and optimisation techniques adopted different types of algorithms, which use different types of parameters in their execution. Specifying these parameters by modeller and analysts takes time and effort, and is error prone. This highlights the importance of a generic methodology that can overcome these shortages and support the modeller and analysts in order to conduct performance analysis and optimisation in easy and flexible way.

2.4 Quantitative aspects of business processes

This section discuss the quantitative aspects of business processes which are related to this research. Quantitative analysis is one of two formal analytic techniques used in business process analysis. Qualitative analysis focuses on the question of whether or not a process design meets a specific criteria. Quantitative techniques, on the other hand, are used to calculate or approximate the size or level of specific properties. However, despite the importance of the quantitative analysis of business processes, few studies in the literature show the use of appropriate models for studying of the quantitative behaviour and optimisation of business processes [11].

In the business process context, quantitative analysis consists of calculating indices of performance, responsiveness, resource utlisation, productivity, quality of service and reliability. A business process can be viewed from different perspectives, however, resulting in different performance measures [11] which may or may not be related. Table 2.2 summarises the perspectives and performance metrics:

Table 2.2: Performance metrics of business processes

Perspective	Performance Measure	Description
Process	*completion time*	The time required to complet one instance of a process
Resource	*utilisation*	The percentage of the operational time that a resource is busy
Product	*processing time*	The amount of time that actual work is perform on relisation of a certain result
System	*throughput*	The number of transactions that a system completes per time unit
User	*responce time*	The time interval between issuing a requist and receiving the result

1. In the process perspective the main performance measure is completion time which is defined as the time required to complete one task of a process which involves, orders, products, multiple customers.

2. In the resource perspective utilisation is a measure of the percentage of the operational time during which a resource is busy.

3. In the product perspective processing time is the time required to perform the result of actual work with out waiting times.

4. From a system perspective throughput is related to the number of requests that a system completes per time unit. In the context of communication networks, throughput may be called bandwidth or processing capacity, and all of these depend on capacity and the number of available resources.

5. From the user or customer perspective response time is defined as the time between sending a request and receiving the result, thus equalling the sum of processing and waiting times. In information technology applications, response time refers to the time between a database query and the production of the results of the search.

2.4.1 Techniques of quantitative analysis

Quantitative techniques can be categorized into simulation and analytical techniques. Simulation is a kind of approximation, while analytical techniques deliver exact numerical results [97]. In analytic techniques, an algorithm provides an exact result on the basis of both the formal model and well-defined relationships between the specified components of the business process. The quantitative evaluation of business processes including performance analysis aims to determine service levels, throughput times, resource utilisation and other performance metrics. The most common formalisms and mathematical theories used to model and analyse business processes analytically are, Markov chains, queuing theory, critical path method, program evaluation review technique and graphical evaluation and review technique [97].

There are three types of information about business processes which need to be collected and analysed. These concern time constraints, the efficiency of the utilisation of resources and time performance [98]. The most important quantitative measures of business processes are thus temporal measures, reliability measures, and cost measures [11]. Also, a combination of temporal and reliability measures to introduce performability measures is possible, such as the performance-to-cost ratio [99]. Temporal performance measures include any time that is taken to complete a specific process.

Three classes of methods can be applied to derive estimates of performance measures: quantitative, simulation and analytical methods [11]. Simulation is a common technique used for analysing business processes, and it provides quantitative estimates of the impact that the design of a process is likely to have on its performance so that a quantitatively supported choice of the best design can be made [100]. Simulation techniques provide a structured environment within which to understand, analyse, and improve business processes [101]. This type of technique is flexible and allows for both the visualisation and performance analysis of business processes, and it has the following advantages [102]:

- Flexibility: Any complex situation can be investigated using simulation.
- Simulation can be used to assess waiting times, occupation rates and fault percentages within same model.
- Simplicity: Only a little specialist knowledge is necessary to understand the model.

However, simulation also has disadvantages, as reported by Van der Aalst [102]:

- A simulation study can be very time consuming.
- The accurate interpretation of simulation results is required.
- Simulation does not offer proofs, but only supports 'what if' analysis.

2.5 Stochastic Petri net

As a part of the methodology presented in this thesis, the business process model is mapped into SPN in order to create a file with a standard format established so as to map the components of business processes to the elements of the SPN model. Petri nets are a graphical and mathematical modelling tool that have already been used as a modelling language for business process workflow systems [50]. Petri nets have a strong mathematical foundation for formal techniques used in both structural modelling and a wide range of quantitative and qualitative methods of analysis. In addition, Petri nets are supported by many tools and applications [50, 103]. In the context of business processes, Petri nets are used in two main ways, as graphical and mathematical tools. For the former, Petri nets are used as a method of visual communication similar to flow charts, block diagrams, and networks, and as a mathematical tool are used to set up mathematical models of how the behaviour of systems is governed. There are at least three good reasons for using Petri Nets for business process workflow modelling and analysis [104]:

1. A Petri net is state-based rather than event-based, so the state of any case can be modelled explicitly. Moreover, it has a strong mathematical foundation which makes it possible to set up formal mathematical models describing the behaviour of systems.

2. A Petri net is a graphical language which uses formal semantics despite its graphical nature. This allows graphical representations of models to be produced which helps in ease understanding the modelled system.

3. Petri nets are supported by many analytic techniques and applications, which makes them suitable for many areas, such as workflow management systems [82, 105], business process management [106, 107], and web service technology [108, 109].

The Petri net is a type of directed graph with an initial state called the initial marking. The basic graph of a Petri net is a directed, bipartite graph consisting of two kinds of nodes. The first are called places, and the second transitions. Arcs (or arrows) connect places and transitions. The arcs can only connect a place with a transition or a transition with a place. Connections between two nodes that are of the same kind are not allowed.

In graphical Petri net representations, places are drawn as circles and transitions as bars or boxes. The initial distribution of tokens among places is called the initial marking of the Petri net. The marking of tokens over places determines the state of a Petri net. A transition is enabled if each place connected to the transition contains at least one token. The firing of transitions changes the distribution of tokens and produces new states. Transitions are generally assumed to be the active components of Petri nets. Transitions can stand for tasks, events, operations, transformations, transportation, and so on.

On the other hand, places are usually passive and they could represent a medium, buffer, geographical location, phase, and condition. Tokens often indicate physical objects or represent information. The flow of these tokens and the firing of transitions are then used to modeled the dynamic behaviour of the system. Figure 2.4 shows a simple example of graphical Petri net, where the circles represents places, the bars correspond to transitions and tokens are represented by small dots. The

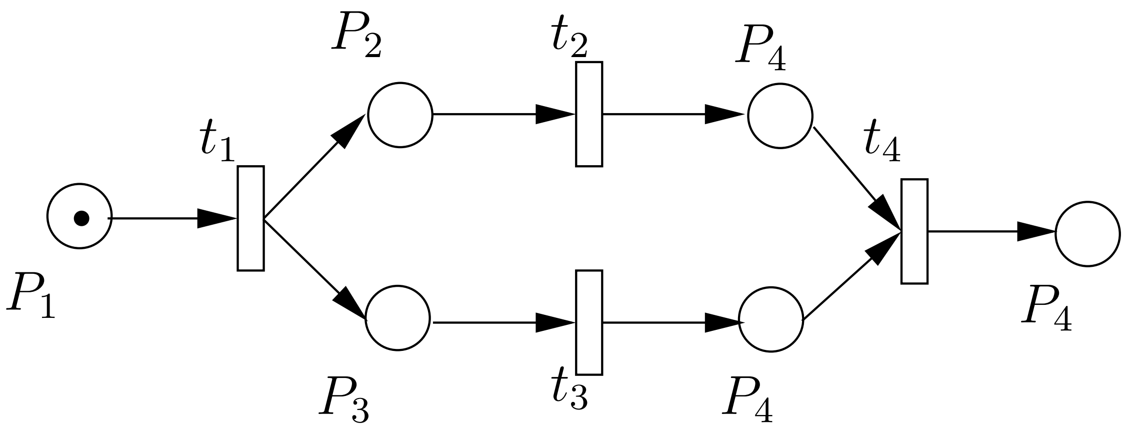


Figure 2.4: A graphical Petri Nets Example

roles of places and transitions depend on application domain and semantics. For example, transitions can represent the processing steps and places the input required and output produced. However, in the context of business process workflow applications, a Petri net model can use transitions to represent tasks and places stand for the pre- and post-conditions of the tasks or resources involved in the system. A transition is enabled if each input place contains at least a number of tokens which equal the weight of the flow relationship from the places to the transition; when a transition fires it consumes a number of tokens from each input place and produces a number of tokens equal to the weight of the flow relation from thr place to the transition.

Places with no incoming arc are called source places and place with no outgoing arc sink places. The preceding and the following places with respect to the transition are the input and the output places respectively for that transition. Places can contain tokens, which represent the marking of the place and reside in it. The numbers of tokens contained in the place are used to define the situation or condition of a Petri net state. The removal of tokens from input places and adding them to output places when firing transitions identifies a change in state and describe the dynamic behaviour of a Petri net. The initial marking denotes the distribution of tokens over places.

The arcs connect places with transitions or vice versa. Arcs are grouped in three types with respect to transitions. These types are as follows:

1. Input arcs: from places to transitions shown by arrows.

2. Output arcs: from transitions to places shown by arrows.

3. Inhibitor arcs: from places to transitions but with a circled head. An inhibitor arc is a special kind of arc used to reverse the logic of an input place. With an inhibitor arc, the absence of a token in the input place rather than its presence enables a transition to fire.

Stochastic Petri nets (SPNs) are built by assigning exponentially distributed random functions to the delays of Petri net transitions. This allows the extraction of quantitative and time-related performance results. The stochastic Petri net is a technique used to model and analyse the dynamic behaviour of parallel and distributed systems such as business processes. It is a mathematical modelling technique used to describe a probabilistic nature as a function of a parameter that usually has the meaning of time. It has been used as a helpful modelling formalism and analytic tool in many applications. It is used for the performance evaluation of distributed and parallel computer systems and has been proposed for modelling the qualitative and quantitative analysis of current systems [50]. SPNs are useful in the evaluation in terms of probabilities of the extent to which a number of characteristics, for instance availability, reliability, safety and maintainability, are satisfied in a system [110].

SPNs use timed transitions, and the delay's of a transition firing is a random variable with an exponential distribution. This means that the transition is associated with a random firing delay whose probability density functions are negative exponentials with specific rates. In this case, the distribution of a random variable X_i of the firing time of a transition is given by the equation: $FX_i(X) = 1 - e^{\lambda.x}$, and the average time of firing the transition T_i is equal to $1/\lambda_i$. This allows the mapping of a stochastic Petri net (SPN) system onto continuous time markov chains (CTMC) [111] in order to analyse and compute interesting performance measures such as the probability of transition firing, the probability of being in a subset of markings, or the mean number of tokens. Stochastic Petri nets (SPNs) are essentially high level models that generate a stochastic process and allow the stochastic nature of business processes to be captured. This includes the dynamic routing of business processes, resource allocation, the forking and joining of process control, and many other characteristics. This stochastic nature permits insights to be gained into the dynamic behavior of business processes and to measure performance indicators.

Using SPNs for performance evaluation simplifies the modelling of business process systems by generating the stochastic processes that govern the system's behaviour. However, the rules of SPNs can also be used to analyse and evaluate the performance of the business processes model. Analytic techniques are used to calculate, for example, the average throughput time of tasks, the estimated throughput of a process and the estimated occupation rate. This can be achieved by using one of the standard Petri net-based analysis tools. This thesis adopts a common formalism for stochastic Petri nets similar to that introduced elsewhere [112]. In this formalism, the firing of transitions in Petri nets is augmented with time and atomic operations. The two types of transitions in SPN are immediate and timed transitions. Immediate transitions fire without delay and have priority over the firing of timed transitions. Timed transitions fire after a random firing delay which is specified by the probability distribution function. The weight associated with immediate transitions is used to determine the firing probability in cases where the transition conflicts with other immediate transitions.

SPNs have the following formal definition.

Definition 2: The SPNs are defined [112] as a 6-tuple, $PN = (P, T, F, W, M_0, \lambda)$, where:

1. $P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of places
2. $T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of transitions.

3. $F \subseteq (P \times T) \cup (T \times P)$ is a set of flow relations.

4. $W : P \times T \cup T \times P \rightarrow \mathbb{N} \wedge [W(x, y) = 0 \Leftrightarrow (x, y) \notin F]$ is a weight function.

5. $M_0 : P \rightarrow \mathbb{N}_0$ is the initial marking, $\mathbb{N}$ is the set of natural numbers and $\mathbb{N}_0$ denotes $\mathbb{N} \cup \{0\}$.

6. $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ are firing rates associated with transitions.

The relationships between places and transition are $P \cap T = \phi$ and $P \cup T \neq \phi$. A place p is called an input place of transition t if there exists a direct arc from p to t. In contrast, place p is called an *output place* of transition t if there exists a direct arc from t to p. A place p contains zero or more tokens at any time, which are drawn as black dots. The initial distribution of tokens among places is called the initial marking of the Petri net. This marking represents a state in the Petri net.

The firing of transitions in SPNs is an atomic operation and changes the distribution of tokens and produces a new state. Each transition has a specific firing delay, which specifies the amount of time that must elapse before the transition can fire. The firing delay is a random variable with a negative exponential probability distributed function (PDF). The firing rate λ_i associated with transition t_i represents the parameter of the PDF associated with t_i, and the average firing delay of transition t_i is ${\lambda_i}^{-1}$. However, the stochastic Petri net package (SPNP) [113] is used as a modelling tool for building the SPN model in this research. The description of this tool has been presented in Chapter 3.

2.6 Summary

A literature survey of business processes in this chapter presents background information concerning important aspects of their quantitative evaluation, which are related to the contributions made by this research. These include definition, modelling, analysis and optimisation of business process as well their types and classifications. In addition, descriptions of the quantitative aspects of and techniques for analysing business processes are presented, and stochastic Petri nets are described. Furthermore, relevant work in the literature concerning the modelling, analysis and optimisation of business processes are is discussed. A number of interesting observations are highlighted concerning the lack of suitable approaches for the analysis and optimisation of business process performance, which has not received as much attention as techniques for modelling and analysing business processes themselves. Business process modelling has attracted the attention of many researchers and has resulted in a variety of modelling approaches, but there is still a need for methods and techniques to support both the modelling and the analysis of business processes and their performance and optimisation. The conclusions of this chapter form the foundation for the methodology proposed in the next chapter.

Chapter 3

Generic Framework for Business Process Modelling and Analysis

3.1 Introduction

Business process management supports business processes using methods, techniques and software to design, enact, control, as will as analyse operational processes. These processes involve humans, organisations, applications, and documents and other sources of information [54]. Modelling and analysis are critical in order to understand business processes and identify their strengths and weaknesses so as to improve performance and enhance the quality of the results obtained. Business process performance analysis is the most significant type of quantitative analysis in this area and is increasingly acknowledged to be a crucial component of business process management. The analysis of business process performance is used to evaluate the ability to meet requirements with respect to throughput and waiting times, service levels, and resource utilization [50].

This chapter describes the generic framework for business process modelling and analysis. A novel engineering methodology is proposed in this thesis to expand the application of quantitative evaluation algorithms to as large a class of business processes as possible. The value of this methodology is to allow modellers and analysts to quantitatively evaluate business processes, thus allowing them to identify the bottlenecks and to make better decisions in order to improve and enhance business processes. The main contribution of this chapter is to clarify the design of the generic framework using Java as object oriented programming language (OOPL) so as to allow a clear understanding of its constitutive modules that are used to achieve the aim of the methodology. The design of the framework is achieved by using Java and introduced in this chapter in its generic form from two perspective, those of users and tool designers. The latter perspective helps in setting the guidelines for the design and development of a software tool using Java that automatically addresses the view of the user perspective. The theoretical basis of the generic design of the framework is represented in the first perspective and the practical aspect is represented in the second perspective. The chapter forms the basis for the detailed description of the framework implementation and use which is presented in Chapters 5 and 6.

The remainder of this chapter is organised as follows. Section 3.2 provides a preliminary outline of the proposed framework including its requirements, characteristics and assumptions. Section 3.3 describes the design of the framework in its generic form, including its three main modules, then presents this design from the two perspectives of a user utilising this framework in Section 3.4, and a tool designer employing it in a software tool in order to automate it, which is presented in Section 3.5. Finally, Section 3.6 concludes the chapter

3.2 Fundamentals of the Generic Framework

This section describes the information that the users of the framework require, and the requirements and the characteristics that should be considered for inclusion in the proposed framework. Finally, the assumptions of the framework are presented.

3.2.1 Preliminary Information

Quantitative evaluation is a technique used to calculate or approximate the size or level of a specific property, and it yields results in numerical form. In the context of business processes, quantitative evaluation includes approaches such as performance analysis and optimisation. As discussed before in Section 2.4, these approaches use different types of metrics such as service levels, throughput times, and resource utilisation. They provide support in determining the presence of weak points and bottlenecks in the system and the influence of the adoption of an alternative resources or different design models. Any specialist who responsible for making a decision regarding any of the above-mentioned issues is a potential user of the proposed framework. This might be a business process engineer, a modeller, or even inexpert users.

Different algorithms are applied to models of business processes for performance analysis and optimisation. Applying the algorithm to a specific business process typically requires bespoke, manual effort by the analyst.

3.2.2 Framework requirements

The requirements for a framework used to perform the quantitative evaluation of business processes can be classified from two perspectives: theoretical feasibility, to show its possible suitability in principle, and practical feasibility, to show its automation and usefulness in the real world. These perspective are described in more detail in what follows.

1. Theoretical feasibility: This concerns the ability of the proposed framework to be used in the quantitative evaluation of business processes. The theoretical description helps in demonstrating the feasibility of the new framework, in addition to helping identify the aspects that should be automated. The theoretical description of the framework is presented in terms of its design, and from the perspective of a user of this design. This is help the reader to understand the details that are either shown to or hidden from users of the framework.

2. Practical feasibility: The framework has to be used in a way that is most helpful to its users. To address the practical feasibility related to automation of the framework in particular, a design for a software support tools using Java as object oriented programming language (OOPL) that employs the theoretical aspects required to be created. Furthermore, present the framework from the perspective of the designer of the tool who uses this framework will help in implementing a tool that automates the process of quantitative evaluation of business process. Using these tools will minimise the level of interaction with a user which is needed and allows the process to be automated. To achieve

this automation, the tools have to automatically map information from the business process model into the stochastic Petri net model. Then, parameters relevant for performance analysis and optimisation techniques are extracted and then mapped into Matlab so that these techniques can be applied.

Giving the above requirements, the proposed framework first needs to be presented theoretically. The theoretical description of the framework is represented by its design from the user perspective and the practical description is represented by consider it from the perspective of a tool designer.

3.2.3 Characteristics of the framework

A new methodology adopted for the design of a generic modelling framework for the quantitative evaluation of business processes faces many challenges, including definition of appropriate modelling techniques and choosing the metrics to be evaluated. In identifying these metrics, there is then the challenge of presenting their semantics and expressing these in a way that is appropriate to the elements of a stochastic Petri net model. Given these challenges, the following issues have to be considered when designing such a framework:

1. Different types of business process modelling for quantitative evaluation used to achieved by the implementation of the framework.

2. Different types of metrics are used to apply these modelling techniques.

3. A stochastic Petri net model is used to achieve a higher level of modularity.

4. Automation is used to fulfil the function of the framework.

The motivation for considering these characteristics is that the framework must be capable of being used for any type of business process, or in other words it must be generic and it must allow the use of a wide range of metrics for performance analysis and optimisation. Automation is one of the most important characteristics of the framework, and allows the mapping of information about business processes into stochastic Petri nets and extraction of the relevant parameters which is necessary to perform the quantitative evaluation process without the need for the intervention of the user. Hence, the tool takes the business process information as input and maps it into a stochastic Petri net, and then the output of the stochastic Petri net is mapped to Matlab.

3.2.4 Basic assumptions of the framework

The design of the framework is based on assumptions related to the business process model and the stochastic Petri net model. The business process model is supposed to be syntactically valid and pre-defined in machine-readable language. The stochastic Petri net model is assumed to be ready to use and that there a tool exists to represent the model with a standard file format which is readable after the mapping process has been completed. The file is created from the automatic mapping of business process information into the stochastic Petri net. Also, a high-level mathematical language is assumed to be available for use for the solution of the relevant algorithms.

3.3 Design of the Framework

This section describes the design of the components of the generic framework for quantitative evaluation of business processes using Java as object oriented programming language (OOPL). The generic framework can be looked at from the perspective of users and tool designers. This section is concerned with the framework design, while its implementation and use is discussed in Chapters 5 and 6. The key features of the generic framework are its three main modules which describe the structure of the framework as well as their essential components. Figure 3.1 shows the modules of the framework, which must be employed consecutively to achieve the framework's purpose. The main modules are defined as follows:

- Business process specification module
- Stochastic Petri net module
- Algorithm module

The first rectangle represents the business process specification module. In this module, the specifications of business processes includes definitions of the elements of the business process model and the relationships between them. The second rectangle represents the stochastic Petri net module, where business processes are mapped into a formal stochastic Petri net model [111]. The third rectangle represents the algorithm module. This module conducts the analytic solutions for performance analysis and optimisation by applying the algorithms relevant to the techniques used. The output produced by one module need to be transformed to the input

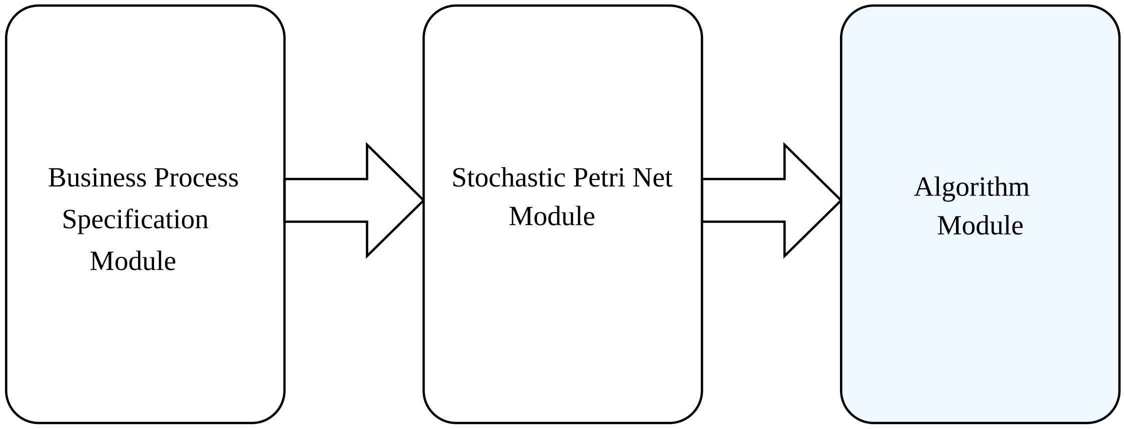


Figure 3.1: The basic modules of the generic framework

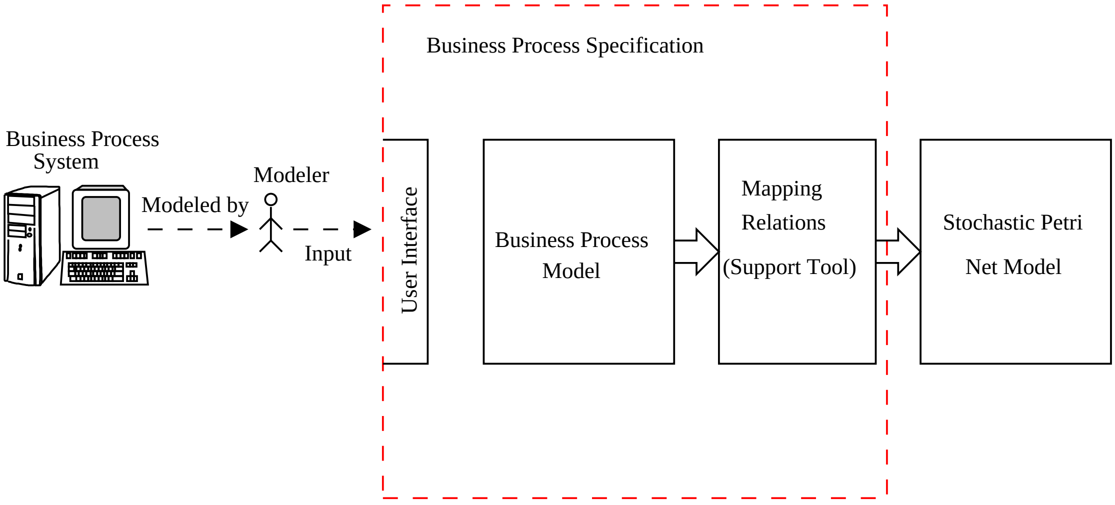


Figure 3.2: Business process specification module

parameters for the next module. A detailed description of each module is given below.

3.3.1 Business process specification module

In this module, the specification of business processes are defined clearly and their elements which are used as input to this module are identified. The module is shown in Figure 3.2. There are different ways to characterise various types of business process models used for performance analysis and optimisation. Business processes involve various tasks and relationships between them, and it is important to investigate the feasibility of using formal language to facilitate modelling and analysis. The modelling technique used should be parsed to obtain the information that is needed in conducting the analysis. This information is related to the technique used and the metrics required. There are many types algorithms for business process performance evaluation and optimisation exist in the literature. They can be quantitatively measured through a measurable metrics which commonly known as performance metrics. These metrics can be related to one of the business process performance dimension presented in Section 2.3. The most important performance metrics are either time-related, such as the throughput time of a process and service time of task, or cost-related such as process, material, and quality-related costs such as visiting frequencies and error rates.

- **Requirements**

  There are many requirements related to the business process specification module which can be specified as follows:

  - The formal representation of the elements of the business processes.
  - A precise representation of the resources used for tasks execution.
  - A valid business process model is available in machine-readable format.

  These requirements specify the attributes of the elements of business process and the dependencies between them so that they can be appropriately mapped into a stochastic Petri net model in the next module.

  The modeler will be asked to provide the main information of a business process model. Many techniques used for the analysis of performance and optimisation of business process can be defined in various languages, as shown in chapter 2. However, the framework proposed in this study is generic and used for represent any business process. Therefore, it should be able to model any business process model to which a specific performance analysis or optimisation algorithm applies. This depends on the modelling technique involved and the tools used to map the model into a formal based format.

  The challenges faced in this module include many approaches to the modelling of business processes which adopt different types of algorithms and metrics for their implementation. This makes the identification of the components of business process models and specifying the relationships between them a complicated task. Moreover, it is difficult to cater for all of these approaches in a single format to be used to map business process models into stochastic Petri net models. To overcome this problem, software support tools are designed using Java which can parse business processes defined using different modelling techniques and then map them into stochastic Petri net models. This will be explained latter in Chapter 5.

3.3.2 Stochastic Petri net module

This module generate stochastic Petri net models where business processes are mapped into stochastic Petri nets. The module is shown in Figure 3.3. The objective of generating a stochastic Petri net model is to create a file format that can be used to establish a standard format for mapping relations from the components of business process to the elements of the SPN.

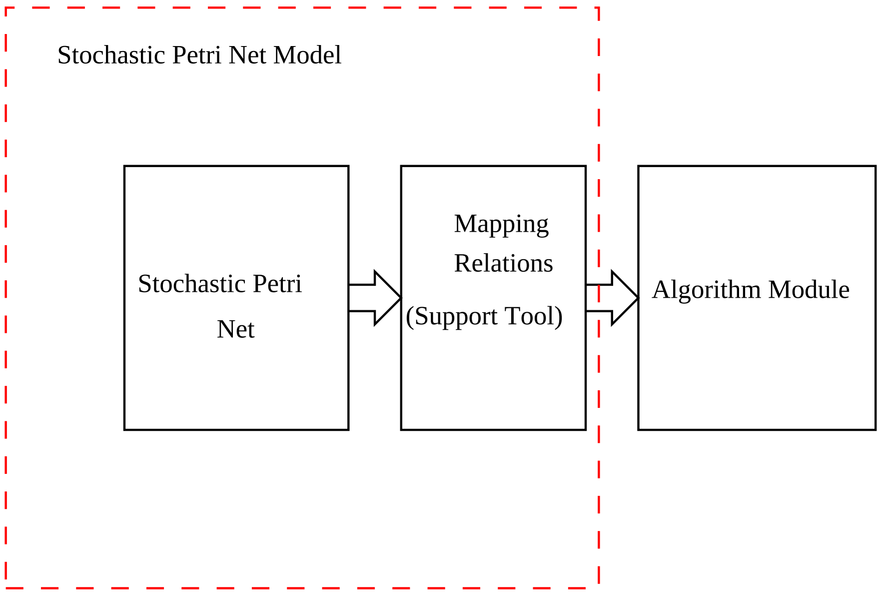


Figure 3.3: Stochastic Petri net module

3.3.2.1 Software tools for building SPN models

Several software tools have been introduced to carry out the steps necessary in building and solving SPN models. Since in this study business process models are mapped into SPN model, the review that follows relates to the tools used to build and solve such models. In this research the Stochastic Petri Net Package (SPNP) [113] is used as a software tool for the building and analysis of SPNs. SPNP is a software modelling tool used for analysing the performance, dependability and performability of complex system behaviour, and to support the specification and solving of SRN models. SPNP is used to build and solve the Stochastic Petri net (SPN) with the underlying Markov Reward Models (MRM). It is used to obtain steady-state, transient, cumulative, and time-averaged measures using an analytic model, as well as to define non-Markovian SPN models. SPNP has a graphical input which uses the iSPN interface and a textual input which uses a C-based SPN language (SCPL). CSPL is a subset of the C programming language used for extra constructs in defining the model primitives [114]. The iSPN interface has a set of GUIs to create the SPN model as follows:

1. A Petri net editor is used to build the graphical SRN model.

2. An environment GUI is used to choose the type of solver and analysis options, in addition to the parameters required by the solver.

3. A GUI for function definition is used to create the reward, probability, guard, distribution, and arc cardinality functions.

4. A GUI for the analysis frame is defines the time used to solve the reward variables.

In this module, SPNP is used as a tool to create the SPN model by mapping the business process information into this model.

3.3.2.2 Mapping business process into SPN

The components of the business process model are represented in different ways depending on the modelling technique used. One of the main challenges for modelers and developers concerns the exchange of data between the different types and formats of applications. The core of this module is the mapping relations to convert business process information into the SPN, which creates a correlation between them in such a way that the model generated contains information which represents the parameters used in performance analysis and optimisation techniques. This is done by taking the business process components as inputs and mapping them to the corresponding elements of the SPN model, an example of which is shown in Table 3.1. An algorithm to map business process components into stochastic Petri nets is proposed to achieve this mapping. The proposed algorithm automatically maps the business process into a stochastic Petri net model by means of a set of defined mapping operations. The algorithm was developed as part of the software support

Table 3.1: Example of SPN file elements

SCPL file elements
Function name
$place()$ $init()$ $rateval()$ $iarc()$ $oarc()$ $Cfunction()$

tools. The engineer needs to provide information about business process through a dedicated user interface. Details of the mapping relation and the support tools used to achieve these operations are presented in chapter 5. The output of this module is a SPN model that represents the business process. The SPN model is then used as the input to the algorithm module.

3.3.3 Algorithm module

The algorithm module is shown in Figure 3.4, where the stochastic Petri nets model from the previous module is used as input. The SPN model is mapped into the algorithms used for business process performance analysis and optimisation techniques. The mapping relations are carried out by using the developed algorithm that can map the elements of the stochastic Petri net model into the parameters of the algorithms of the technique used. The steps that have to be conducted in this module can be explained as follows:

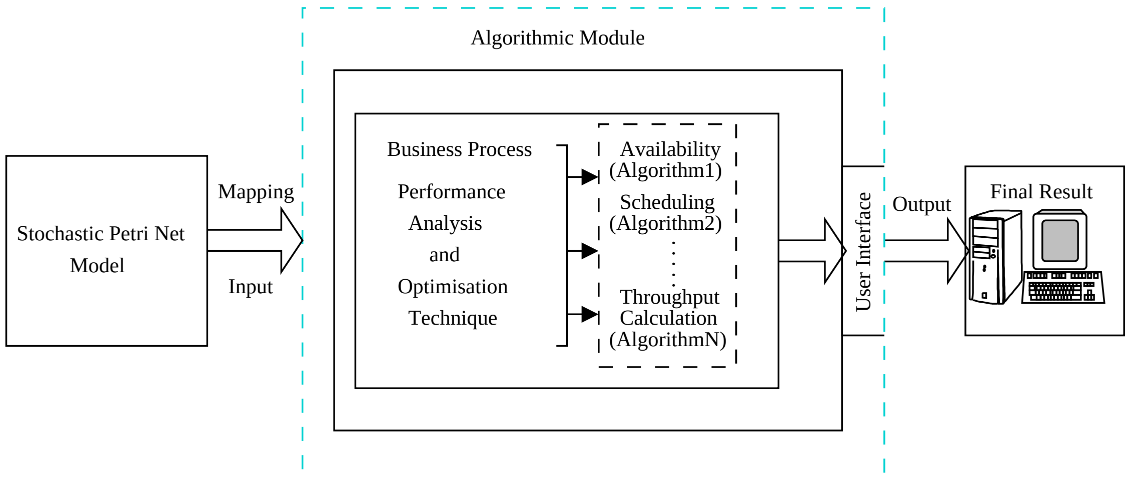


Figure 3.4: Algorithm module

- **Step 1**: The stochastic Petri net model generated from the previous module is used to generate input to this module.

- **Step 2**: The stochastic Petri net model is mapped into the algorithm module. This includes the extraction of the parameters used by the performance analysis and optimisation techniques from the elements of the stochastic Petri nets model.

- **Step 3**: The desired solution can then be conducted based on the algorithm selected and the values of performance indices. The output of this module is the results of performance analysis and optimisation, which are conducted according to the proposed algorithms used by these techniques.

3.4 The framework from the user's perspective

The user's view concerning the design of the framework depends on its ability to automate the aforementioned three modules. From the user's point of view, automating the modules allows the framework to be seen as a black box. Based on

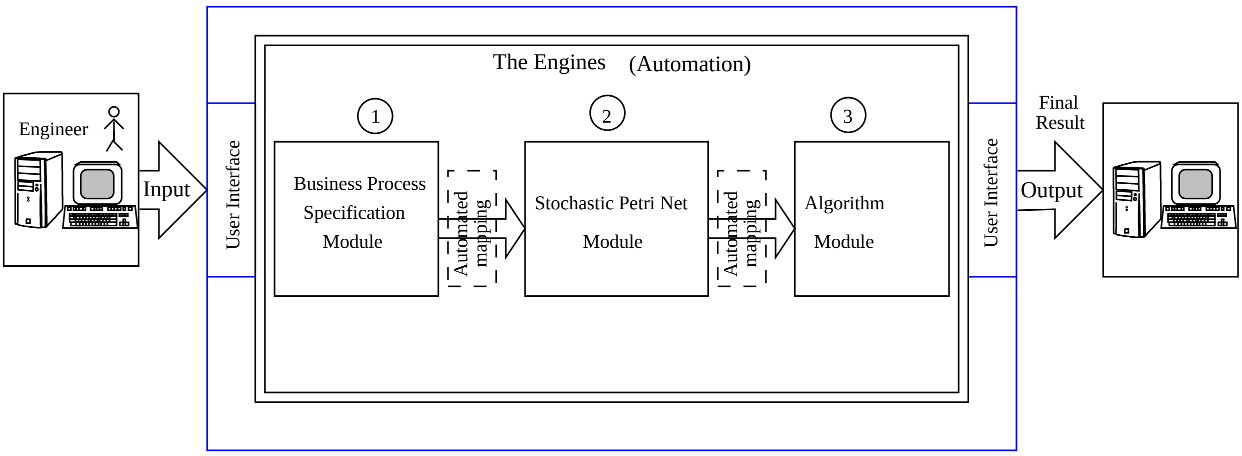


Figure 3.5: The generic framework from a user perspective

the design of the framework as described earlier, Figure 3.5 depicts the modules that have to be mapped automatically to the subsequent module. The framework is shown as three engines which represent the three main modules which are mapped to each other. The automated mapping of the modules is illustrated in this figure as dashed rectangles between engines 1 and 2, and 2 and 3. According to the figure, the framework works as follows:

1. The user has to supply the business process information through the user interface and in order to complete the generation of the model. This is part of the business process specification module.

2. The result of performance analysis and optimisation is shown to the user through the other user interface.

The implementation of the framework involves the automation of process of the function performed by the modules using software support tools which built in Java. The software support tools developed to automate the process are shown in Figure 3.5. The modules are the three internal boxes marked 1, 2, and 3 are fully automated and the user will not interact directly with the execution of their operations, which are implemented within the engines. The user interface gives the flexibility to feed the required information about the business process into the framework. After all of the processes are completed, the user receives the final results of performance analysis through the other user interface. However, in order to avoid repetition, descriptions of the automated parts used in the framework are presented in Chapter 5 where the design of the tool architecture is described.

3.5 The framework from the perspective of the support tools designer

The theoretical design of the framework proposed in Section 3.3 should make it possible to map the business process model into a stochastic Petri net model, and then to map the stochastic Petri net model into the algorithm model. The framework from the perspective of the designer of the software support tools is shown in Figure 3.6. The tool designer uses existing and new tools to create the support tools that implement the framework design. The existing support tools used are the Stochastic Petri Net Package (SPNP) [113], which is used to represent the stochastic Petri net model, Java as object oriented programming language (OOPL) and Matlab [115] which is a high-level mathematical language used to represent the algorithm model. The design of support tools for the new solution includes the creation of two algorithms. The first of these is used to map the business process specifications into the stochastic Petri net model which is represented by SPNP, and the second algorithm is used to map the stochastic Petri net model into the algorithm module which is

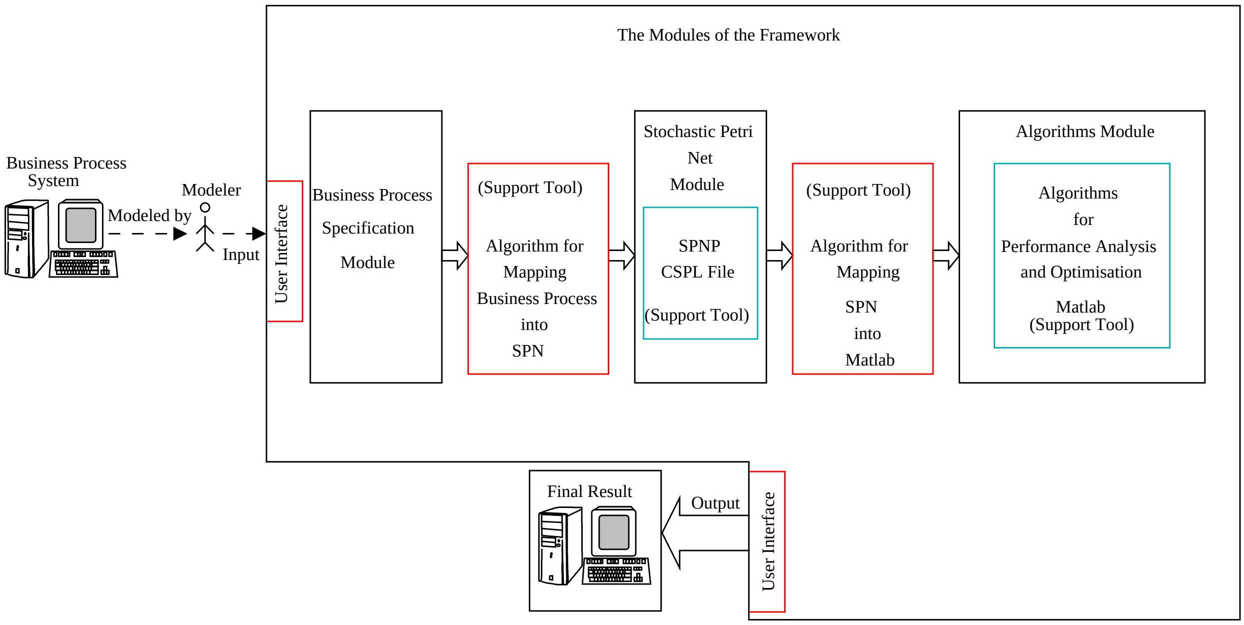


Figure 3.6: The framework from tool designer's perspective

represented in Matlab.

It is appealing to a user if this framework can be supported by a software tool which will automate the implementation of the modules. This software tool has to consider the theoretical perspective of the framework design so that a user's perspective can be adopted. The tool designer has to represent each module so as to fully automate its functionality. Furthermore, the designer of the tools has to use a set of novel and existing techniques to increase the modularity and automation of the tool using this framework.

The software tool support can be seen from two practical ways according to their uses, the support tools designed and the existing tools. The support tool designed is used in the framework depending on their functionality. These tools include, the user interfaces and the mapping relations between the modules of the framework as well as for solving the algorithm models. The GUI are used to insert the business process information, and to display the final results. The GUIs are built using the Java [116] as object oriented programming language (OOPL) on the Eclipse platform. The mapping relations support tool is designed using Java to map from one module of the framework to the next, as explained earlier. Matlab is a high-level mathematical language used as another type of software tool support for the algorithms implementation. The limitations of the framework when it implemented can be determined by the following two perspectives:

- Tools: limited by practical limitations of solution methods (CPU and memory of computing equipment).
- Translations: effort to implement one or more translators for every new algorithm an additional model formalism.

The description of the perspective of the designer of the support tools which represent the support tools architectural design is presented in chapter 5 along with the implementation of these tools.

3.6 Summary

This chapter describes the generic framework for business process modeling and analysis, and sets down the requirements, assumptions and characteristics that need to be included in its design. The design of the framework consists of three main modules of business process specification, stochastic Petri nets and algorithms. These should allow users to obtain the results of the quantitative evaluation of business processes in an automated way. The framework design is looked at from two practical perspectives. It is presented from a user's perspective, highlighting the automated steps where the user is responsible for entering business process information and there is no directed interaction between him and the functions of the modules. Also, a brief description of the framework is given from the perspective designer of the software tools which are used to implement the framework automatically. Full descriptions of the framework modules and implementation and the design of support tools are presented in Chapter 5 and 6.

Chapter 4

Business Process Workflow with Resources (MWF-wR) Model

4.1 Introduction

This chapter presents the generic formalism for modelling business process workflow with resources *MWF-wR*, (Modelling Workflow with Resource). The formalism is used to define various performance evaluation and optimisation algorithms of business processes. In the context of the framework presented in chapter 3, it fits with the business process specification module which is described in Section 3.3.1.

Many formalisms have been suggested for modelling workflows, but this chapter presents a generic formalism for modelling business process workflows that fits with the aim of this thesis. More specifically, the proposed *MWF-wR* formalism is defined to expand the applicability of various performance evaluation and optimisation algorithms to as wide a class of business processes as possible. In the *MWF-wR* formalism, tasks are defined and associated with resources in an extensible way. Petri nets have been widely used to model and analysis workflow processes and verify the behavioural correctness of workflows. In this work, the *MWF-wR* formalism is mapped into a Stochastic Petri net (SPN) [111, 117] model to carry out performance analysis and the quantitative evaluation of business processes in an automated a way as possible. The aim of the *MWF-wR* formalism in this chapter is to be as widely usable as possible (i.e generic) for modelling workflow components and to introduce correct input for algorithms used for business process analysis and improvement.

The chapter is organised as follows: in Section 4.2, the fundamentals of the workflow model are depicted, while the requirements for the *MWF-wR* formalism are outlined in Section 4.3. Section 4.4 demonstrates the definition of the modelling workflow with resources *MWF-wR* formalism. The relations for mapping the *MWF-wR* formalism into *SPN* are presented in Section 4.5. Section 4.6 discusses the representation of mapping elements of the *MWF-wR* formalism into elements of the *SPN* model. Finally, Section 4.9 summarise the chapter.

4.2 The Fundamentals of Business Process Workflow Modelling

Before presenting the *MWF-wR* formalism, the fundamentals of workflow modelling is discussed and the concepts relevant to the present study are evaluated. So, Section 4.2.1 explains the general concept of modelling workflow related to the the *MWF-wR* formalism. In Sections 4.2.2 and 4.2.3, the main characteristics and structure of workflows which are considered in building the *MWF-wR* formalism are presented.

4.2.1 Workflow model and the *MWF-wR* formalism

The concept of modelling workflow entails the task of creating workflow specifications. Up till to now there is no generally accepted workflow model, but there have been many efforts to define such a model [118]. In the *MWF-wR* formalism, a business process workflow model is developed which incorporates traditional workflow modelling features as well as having the ability to 'annotate' the model to introduce input for the algorithms used.

Typically, workflows are represented by workflow graphs or process maps, where the nodes represent the constituent tasks and the edges correspond to dependencies between them. In general, a workflow consists of a set of tasks, each of which may communicate with another task in the workflow. Figure 4.1 shows an example of a workflows graph that is based on the aforementioned notions where the boxes stand for tasks in the workflow and arrows represent the relations between them.

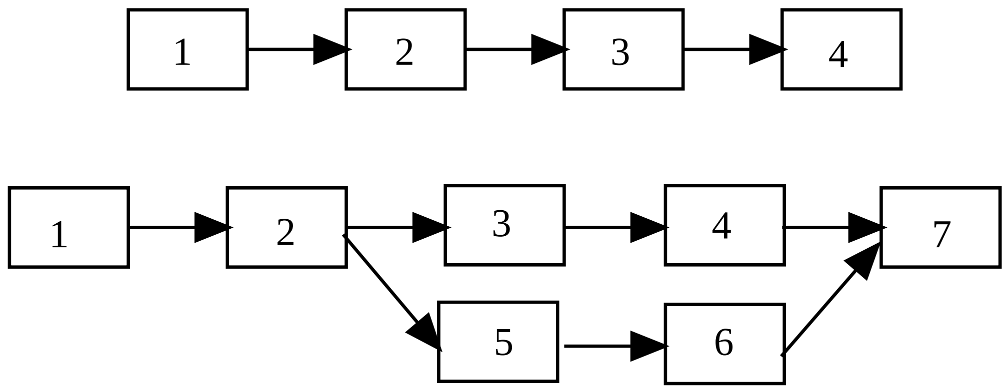


Figure 4.1: Example of workflows graph

The general modeling method can be summarised in the following steps [119]:

- Identification of events and resources: given a system description, identify the major events, processes, resources, conditions, routing information, etc.
- Identification of relations: the relationships among the above identified events, and/or processes have to be determined. The resources should be identified and major allocation policies for shared resources and routing information concerning on products should be determined. Then the initial approximate net structure can be decided.

The sequence of processing tasks and the data flows between them in the *MWF-wR* formalism are determined by dependencies. The tasks can be executed sequentially, or in parallel.

4.2.2 Workflow characteristics and *MWF-wR*

The aim here is to use the *MWF-wR* formalism as a generic formalism and therefore this section presents the main characteristics of workflows which are related to the evaluation algorithm. In business process workflow as well as the *MWF-wR* formalism, tasks are performed by a number of resources. These resources can be any entities required to complete a task. They can be either a physical entity (processors, memory, printer machines, people, organizations, etc) and a virtual entity (software libraries, web services, databases, etc). The tasks in a workflow can be executed by one resource or a set of resources.

The ordering of the tasks that need to be executed is defined as partial ordering. The tasks in a workflow can be of different types, such as those using computer programs or these requiring manual jobs. However, the execution of tasks is achieved by the resources available involves costs, which depend on resource allocation or scheduling. This prompts organisations to seek optimal resource utilisation so as to achieve cost effectiveness.

4.2.3 Workflow structure

Workflow specifications consist of several types of routing [120]. In this section, the basic routings that are related to workflows are presented. The block structures of routing specify dependencies among tasks. The basic routings defined by Aalst et al. [121] capture elementary aspects of process control as shown in Figure 4.2. These basic routings are considered in the *MWF-wR* formalism and affect workflow evaluation and scheduling problems. The basic routings are defined as follows:

- **Sequential routing**: This is a fundamental building block used to model consecutive steps in a workflow process. It is used to deal with causal relationships between tasks. In a workflow process, a series of consecutive activities are executed one after another. One activity is enabled after the completion of another activity in the same process. In Figure 4.2.a, if task $Task_2$ is executed after the completion of task $Task_1$, then tasks $Task_1$ and $Task_2$ are executed sequentially.

- **Parallel routing**: This is used for modelling situations where the order of execution is less strict. In parallel routing more than one task can be carried out at the same time, or even in any order. Two building blocks are used to model parallel routing (Figure 4.2.b).

  - **AND-split**: In the workflow process, this is a point where a single thread of control splits into multiple threads of control which can be executed in parallel, thus allowing more than one task to be executed simultaneously or in any order.

  - **AND-join**: This is a point in the workflow process where multiple parallel activities converge into one single thread of control and the control

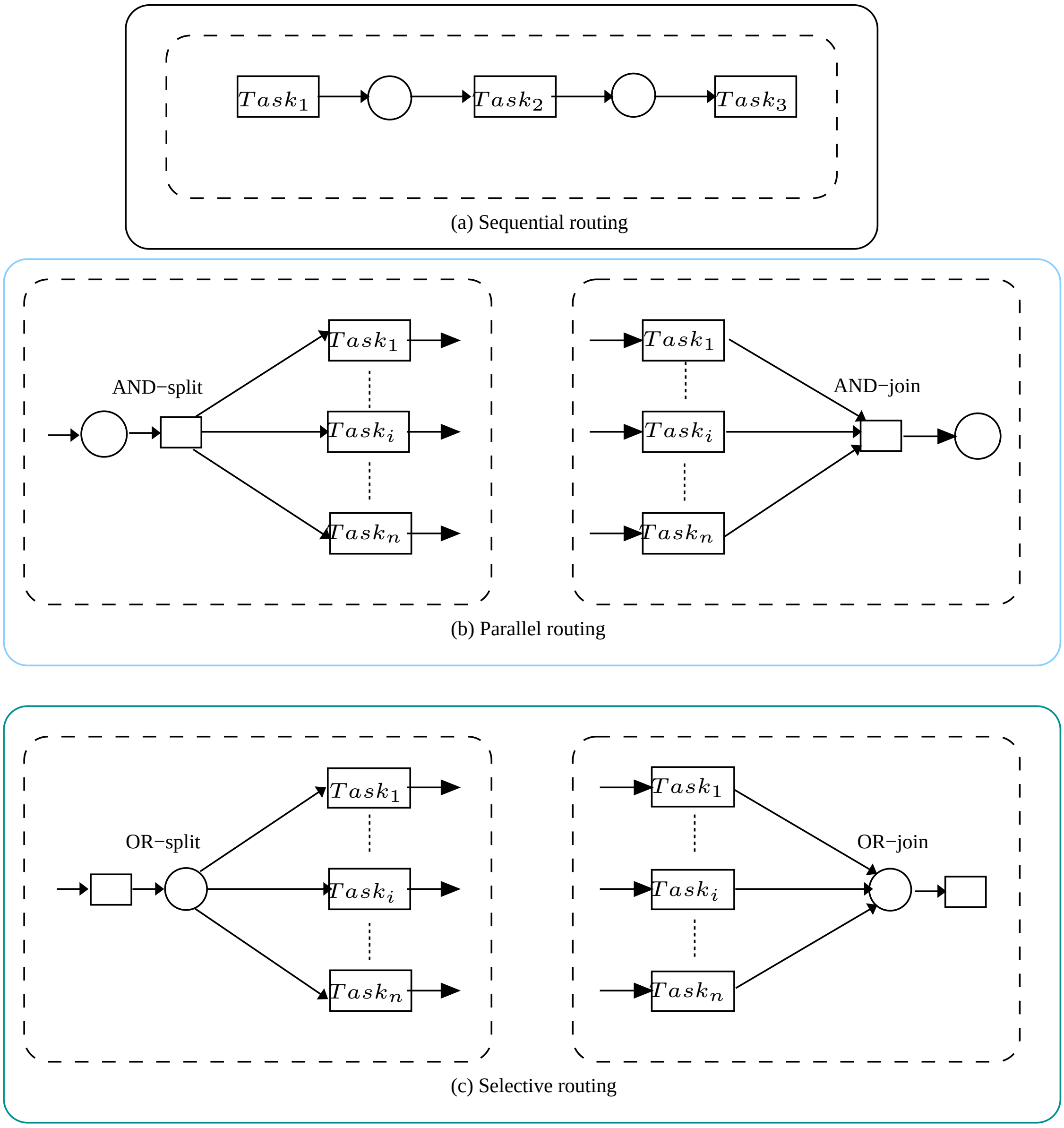


Figure 4.2: Basic workflow routings

is passed to the subsequent branch when all input branches have been enabled.

- **Selective routing**: This type of routing is used where a single thread of control makes a decision about multiple alternative subsequent workflow branches. Two building blocks are used to model this routing (Figure 4.2.c).
  - **OR-split**: A point in the workflow process where a branch divergence into two or more branches, based on a decision or workflow control data one of these branches is chosen.
  - **OR-join**: A point in the workflow process where two or more alternative branches come together without synchronization. In other words,

the merges will be triggered once any of the incoming transitions are triggered.

4.3 Requirements of the *MWF-wR* Formalism

Due to the complexity of real business process workflow systems and the various types of performance analysis and optimisation algorithms used, it is very difficult to involve all the information necessary in a single workflow model [122]. By asking the model to support quantitative criteria, formal syntax and semantics are required. Otherwise the model would not be executable and suitable for analytic techniques. Therefore, the proposed *MWF-wR* formalism should have formal semantics. It was decided that the *MWF-wR* formalism should be mapped into *SPN*, because *SPN* has formal semantics and has many support tool analysis. There are general requirements for modelling workflow [123]. As the *MWF-wR* is intended to be used as a generic formalism for modelling business process workflow, so, the requirements in defining the formalism so as to satisfy this aim should be identified. These requirements are as follows:

- The formalism should cover the general specifications of a business process workflow which include different types of algorithms for performance evaluation and optimisation.
- The formalism should present an approach which can be used to manage resources in the business process so as to satisfy the requirements of different optimisation and performance evaluation algorithms. It should be capable of define new variables which is used as input to these algorithms.
- The design of this formalism should provide a structure for the explicit representation of the elements of business processes.
- A workflow model supported by a formal modelling technique should have an underlying formal semantics, since this will support requirements of quantitative evaluation models.
- The formalism should be easily understandable for software developers, business process analysts and managers in order to be used for performance analysis and the optimisation of models.

Given the requirements for modelling business process workflows described above, the next section presents the definition of the *MWF-wR* formalism.

4.4 Formal Definition of Modelling Workflow with Resources (*MWF-wR*)

This section introduces the formal definition of the *MWF-wR*. The *MWF-wR* formalism is created to allow the representation of business processes and to be able to support the modelling of different phases of the business process. It has a formal semantics and is well suited to quantitative analyses. The *MWF-wR* formalism is formulated to model any business process and to facilitate the performance analysis of the models created. Many techniques have been developed for a variety of types of problem. The *MWF-wR* formalism defines the general workflow model and many types of performance analysis and optimisation techniques fit into this definition.

The previous section described the requirements for modelling workflow and then the requirements for the *MWF-wR* formalism to be able model the quantitative evaluation techniques. With respect to the above requirements, the definition of the *MWF-wR* formalism used in this work for modelling business processes explained in the following section.

Definition 4.1:
MWF-wR is a 5-tuple ($MWF - wR = TS, R, PRO, TM, TR$), where:

1. $TS = \{ts_1, ts_2, \cdots, ts_m\}$; finite set of tasks in the workflow.

2. $R = \{r_1, r_2, \cdots, r_n\}$; finite set of resources of the workflow, and each resource r_i is a set of variables such that:

   - $r_i = \{(c_i, s_{ij})\}$; for $1 \leq i \leq n$, and $1 \leq j \leq m$;
     (i.e. $r_i = \{(c_i, s_{ij}), (c_i, s_{ij+1}), \cdots, (c_i, s_{im})\}$;
   - c_i = cost of r_i
   - s_{ij} = processing time required to process task ts_j by resource r_i, $s_{ij} \in TM$.

3. $PRO \subseteq (TS \times TS)$; a partial order relationship among the tasks in the workflow, called the precedence relation.

4. $TM = \{s_{ij}\}$; a set of processing times for tasks; for $1 \leq i \leq n$, and $1 \leq j \leq m$; (i. e. s_{ij} is the processing time required to process task ts_j by resource r_i); $s_{ij} \in \mathbb{R}$.

5. For all i, $1 \leq i \leq n$, $TR_i \in (TS \times p_i(R)) \nrightarrow TM$; a set of partial functions from $(TS \times p_i(R))$ to TM defined for each task ts_i:

   - the resource set $p_i((R) \subseteq R$ that can be used to complete task ts_i, $1 \leq i \leq n$
   - the processing time s_{ij} required to process ts_j by specific resource r_i

This definition specifies the main elements of the formalism required to model a business process workflow. Tasks are denoted by TS and each task by ts_i. Resources are denoted by R and each resource by r_i. The resources are used to process the tasks and it is possible that one resource r_i can be used to process more than one task. This mean that a task ts_j can be processed by a set of resources $p_j(R)$. But one resource r_i can be used to process only one ts_j task at one time (i. e. no more than one task can be processed by one resource at the same time). Each resource r_i has a fixed cost c_i and depends on the algorithm used. We note that costs are constant. Time-dependent costs are not considered in this work. Inclusion of time-dependent cost would require resolutions similar to that of non-homogeneous Markov chains [124], with time dependence of costs instead of transition rates. It is assumed that resources are always available. The flow relation PRO is used to specify the ordering relations among tasks. If task ts_1 has to be processed before task ts_2, then $(ts_1, ts_2) \in PRO$ (i.e. the execution of task ts_1 has to be completed before the execution of task ts_2 may start). The execution time for each task is denoted by the set time TM and $\mathbb{R}^+ \cup 0$ is a typical choice for TM. The time s_{ij} denotes the processing time required to process task ts_j by the specific resource r_i. The following are defined for each task ts_j:

- a set of resources $p_j(R)$ that can be process the task.
- a set of times s_{ij} which denote the time required to process task ts_j by resource r_i.

- a set of costs c_j of each resource r_j.

The following example clarifies this definition and makes it more understandable.

Example 4.1:

Let us consider an example of the *MWF-wR* workflow with two tasks $\{ts_1, ts_2\}$ which have to be processed by three resources $\{r_1, r_2, r_3\}$. Let the *MWF-wR* business process be to buy a book, which consists of two tasks:

- ts_1 = order a book
- ts_2 = pay for the book

There are three resources available for processing these tasks. The resources are:

- r_1 =Alice
- r_2 = Bob
- r_3 = Web Service

The task *order book* can be carried out by *Alice* and *Bob*, and this is followed by the task *pay book* which can be carried out by *Bob* and the *Web Service*. The *process time* required to complete the task *order book* by *Alice* is equal to *(8 time units)* and that by *Bob* is equal to *(6 time units)*. The *costs* of these two resources are equal to 7 price units / per time unit and 9 price units / per time unit respectively. The *process time* required to complete the task *pay for book* by *Bob* equals *7 time units* and that by the *Web Service* equals *2 time units*. The *cost* of the *Web Service* then equals 4 price unit / per time unit.

The corresponding business process of the $MWF - wR = (TS, R, PRO, TM, TR)$ is specified as follows:

1. $TS = \{ ts_1, ts_2\}$;
    - ts_1 = order book
    - ts_2 = pay for book
2. $R = \{ r_1, r_2, r_3\}$ =

    { Alice, Bob, Web Service}

    - $r_1 = \{(c_1, s_{11})\}$

      Alice = {(7 price units, 8 time units)}

- $r_2 = \{(c_2, s_{21}), (c_2, s_{22})\}$
  Bob = {(9 price units, 6 time units), (9 price units, 7 time units)}

- $r_3 = \{(c_3, s_{32})\}$
  Web service = {(4 price units, 2 time units)}

3. $PRO = \{\ (ts_1 \times ts_2)\ \}$

   - Partial order = { order book, pay for book }

4. $TM = \{(s_{11}), (s_{21}), (s_{22}), (s_{32})\ \}$

   - $TM = \{(8), (6), (7), (2)\}$

5. $TR = \{((ts_1, (r_1, r_2), (s_{11}, s_{21})), ((ts_2, (r_2, r_3), (s_{22}, s_{32}))\}$

   - TR = {((order book, (Alice, Bob),(8, 6)), ((pay for book, (Bob, Web Service),(7, 2))}

This definition specifies the data required to formulate the*MWF-wR* model. Each task needs specific resources to complete its process time. The resources have their costs and process times associated with the task. The domain of the function *TR* signifies the resource sets able to process a specific task. The processing times are also specified by *TR*. So, a connection can be made between tasks and resources throughout this function. For the objectives of quantitatively evaluating business processes in this research, the *MWF-wR* formalism should be mapped into *SPN* model. The next section explain this mapping process.

4.5 Mapping the *MWF-wR* Formalism into Stochastic Petri Nets

Petri nets are considered to be one of the most popular formal techniques for the modelling and analysis of workflows and workflow systems, because of the workflow parts can be described as well-defined pieces of reality [104]. Despite the existence of many techniques for modelling workflow, Petri nets are the only formal techniques able to be used for both the structural modelling and quantitative analysis of business process workflow systems [82]. However, although Petri nets are widely used for the modelling and analysis of workflow system, they cannot be mapped onto workflow models in a straightforward way. This is because, for instance, there are no concepts of resources in Petri nets, so that an additional concepts with respect to the requirements of workflow models must be added in order to allow quantitative evaluation.

In general, the workflow can be mapped into Petri nets, where each task in the workflow is represented by a corresponding transition or place in the Petri nets. Places may represent the resources needed for a task to be performed. The logical relationships between tasks and the flow of work are then represented by arcs. The initial marking of places represents the start of possible states.

A more suitable form to be used for mapping business processes to Petri nets is an extensions of the Petri nets with time. *SPN* is an important technique that has been used amongst the other types of Petri nets for the modelling and performance analysis of business processes. Time is a natural concept in *SPN* models associated with timed transitions. Timed transitions have exponentially distributed firing delays, each of which is defined by exponential polynomials. To conduct the mapping of the *MWF-wR* into *SPN*, Stochastic Petri Net Package *(SPNP)* modelling tool was used to build the stochastic model. A customer structure file is produced using a *CSPL* (C-based SPN language) specific to *SPNP*. The *CSPL* file contains functions and variables used to describe *SPN* model. The mapping of *MWF-wR* into a *CSPL* file is demonstrated in the next Chapter. The next section defines the *MWF2SPN* algorithm that maps the *MWF-wR* model by means of a set of defined mapping relations and operations.

4.5.1 The ***MWF2SPN*** algorithm for mapping ***MWF-wR*** into ***SPN***

This subsection presents the *MWF2SPN* algorithm used for mapping from the *MWF-wR* formalism into the stochastic Petri nets model. The mapping relation allows all business processes within the *MWF-wR* class to be mapped into the SPN model with the objective of allowing the performance analysis of business processes to be conducted in an automated manner. SPN is used instead of classical Petri nets because it has no intrinsic notion of time, and so the processing time can be associated to tasks in a more flexible way. Moreover, *SPN* has many analytic tools that have a standard formats which can be used for mapping relations such as *SPNP*. The *MWF-wR* formalism for business processes described in definition 4.1 consist of tasks, resources, ordering relations between tasks, times set for the execution of tasks and the costs of resources. Hence, these concepts have to be mapped onto concepts of places, transitions and arcs in *SPN*. Before performing the mapping between *MWF-wR* and *SPN*, a formal definition of *SPN* is first presented. The common formalism for stochastic Petri nets [125] is adopted.

Definition 4.2:

SPNs have been defined as a 4-tuple, $SPN = (P, T, F, \lambda)$ [125], where:

1. $P = \{p_1, p_2, \cdots, p_m\}$ is a finite set of places

2. $T = \{t_1, t_2, \cdots, t_n\}$ is a finite set of transitions.

3. $F \subseteq (P \times T) \cup (T \times P)$ is a set of flow relations.

4. $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ is a firing delay associated with each transition.

The relationships between places and transitions are $P \cap T = \phi$ and $P \cup T \neq \phi$. A place p is called an *input place* of transition t if and only if there exists a direct arc from p to t. In contrast, place p is called an *output place* of transition t if and only if there exists a direct arc from t to p. A place p contains zero or more tokens at any time, which are drawn as black dots. The initial distribution of tokens among places is called the initial marking of the Petri net. This marking represents a state in *Petri net*. Each transition has a specific firing delay, which specifies the amount of time that must elapse before the transition can fire. The firing rate λ_i is associated with the timed transition t_i of the Petri net.

The *MWF-wR* formalism and *SPN* have now been defined in definitions 4.1 and 4.2 respectively. The *MWF2SPN* algorithm presented in Table 4.1 describes the mapping relation from *MWF-wR* to *SPN*. The *MWF2SPN* algorithm determines

Table 4.1: MWF2SPN algorithm for mapping *MWF-wR* to *SPN*

For all $p_i \in SPN$ and $ts_i \in$ MWF−wR do:
$p_i = ts_i$
(Mapping each task ts_i of MWF−wR
to place $p_i$of SPN)
For all $t_i \in SPN$ and $ts_i \in$ MWF−wR do:
$t_i = ts_i$
(Mapping each task $ts_i$to transition t_i)
For all $p_j \in SPN$ and $r_j \in$ MWF−wR do:
$p_j = r_j$
(Mapping each resource r_j to place p_j)
For all $p_j \in SPN$ and $r_j \in SPN$ do:
marking of p_j = c_j
(Mapping marking of place p_j to cost c_j of resource r_j)
For all $\lambda \in SPN$ and $s \in$ MWF−wR do:
$\lambda_{ij} = s_{ij}$
(Mapping firing rate of transition t_i to
processing time of task ts_i by resource r_j)
For all $p_i \in SPN$ and $p_i(R) \in$ MWF−wR do:
marking of p_i = number of resources in $p_i(R)$
(Mapping marking of place p_i to resource set $p_i(R)$
of task t_i)
End

the mapping relationship of the *MWF-wR* model into SPN. Each task ts_i in *MWF-wR* is mapped into a place p_i and transition t_i of SPN. The resource r_j is mapped to place p_j. For each task ts_i a resource set $p_i(R)$ is defined which signifies the resources that can be used to complete the task associated with the same transition t_i which represents task ts_i. The numbers of each resource set for each task are mapped to a number of tokens (marking) in the place representing this task. The firing rate of transition t_i is determined by the resource set $p_i(R)$ used to process the task ts_i. When task ts_i is processed by resource r_j, the processing time will be s_{ij}. The cost of resource r_j is mapped to the marking of place p_j. To clarify the mapping relation introduced in *MWF2SPN*, the next section illustrates the mapping relations of each element in more detail.

4.6 Mapping *MWF-wR* components into SPN

To clarify the *MWF2SPN* algorithm, this section presents details of the mapping of each component of *MWF-wR* into SPN. The mapping relations are given as follows:

1. **Task mapping**

   Figure 4.3 shows how a task ts_i modeled from the *MWF-wR* formalism in

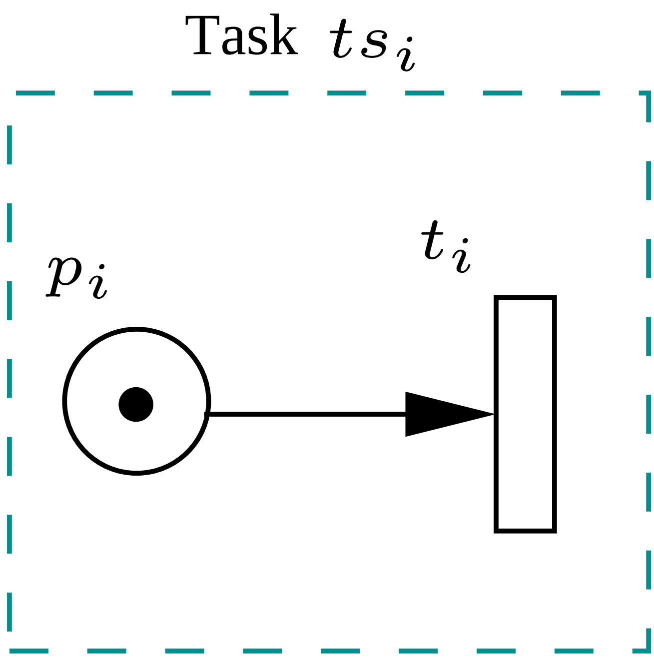


Figure 4.3: The $MWF - wR$ task in Petri net

   terms of an SPN. Each task is modeled according to place and transition. The token is used in the input place p_i of transition t_i to represent the resource set, or number of resources used to process this task. A transition t_i has at least two input places, which represents the task and the second which represents the resource used for task processing in cases only one resource is used (Figure 4.4). The firing rate of transition t_i is λ_i and is exponentially distributed. The processing time of a task ts_i is mapped to the firing rate λ_i of the corresponding transition t_i.

2. **Resource mapping**

   In business processes, the resources used for task execution are associated with costs. However, a task may be executed by one resource, or it can be executed by more than one resource. So, in the second case, the execution of the task is carried out by a set of resources. Figure 4.4 shows the modelling

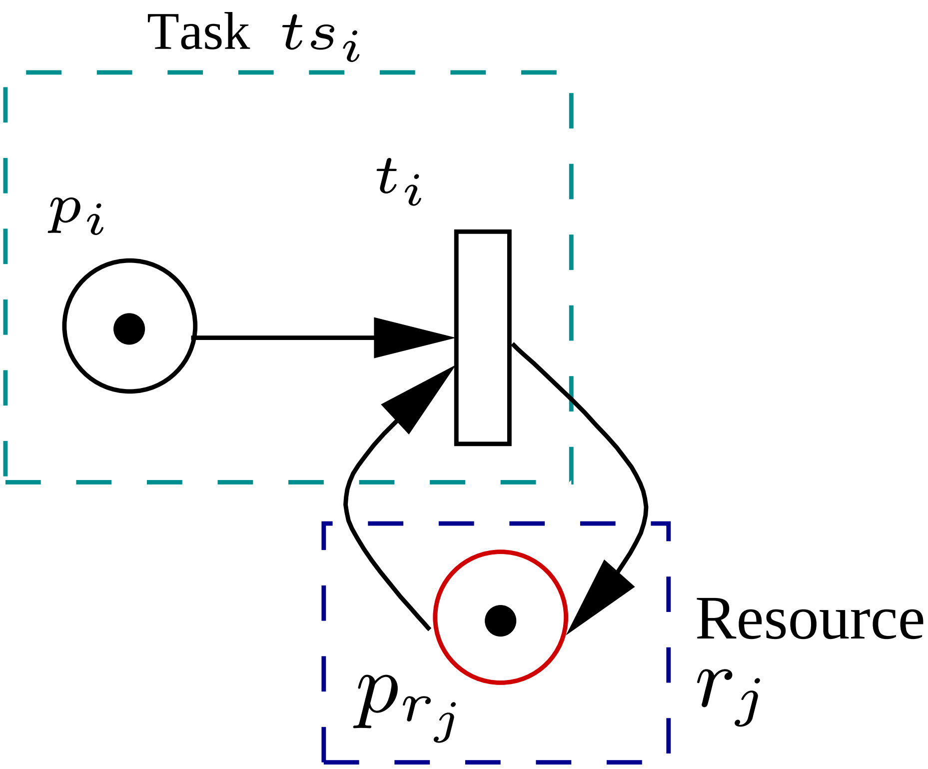


Figure 4.4: Task and resource of $MWF - wR$ in Petri net

of first case in Petri nets, where the resource r_j is modeled by place p_{r_j}, and the task ts_i is modeled by place p_i and transition t_i. The place p_{r_j} contains a token, which is used to model the cost c_j of the resource.

The representation of the second case is shown in figure 4.5. In this case, the task ts_i is executed by a set of resources. Two resources r_1 and r_2 are used to execute the task ts_i and they are represented by places p_{r_1} and p_{r_2} respectively. The task ts_i is denoted by the place p_i and the transition t_i, p_{r_1} when it is processed by resource r_1, or place p_i and transition t_i, p_{r_2} when it is processed by resource r_2, and by place p_i and transition $t_i, (p_{r_1}, p_{r_2})$ when it is executed by both resources r_1 and r_2.

3. **Processing time mapping**

   The task ts_i in Figure 4.4 is denoted by place p_i and transition t_i. The processing time s_i of the task is mapped to the firing rate λ_i of transition t_i, that is, $s_i = \lambda_i$. The firing delay of the transition t_i equals the processing time s_{ij} of task ts_i when the task is processed by resource r_j. So, we have $s_{ij} = \lambda_{ij}$, where λ_{ij} is the firing rate of transition t_i when it is connected to place p_{r_j}.

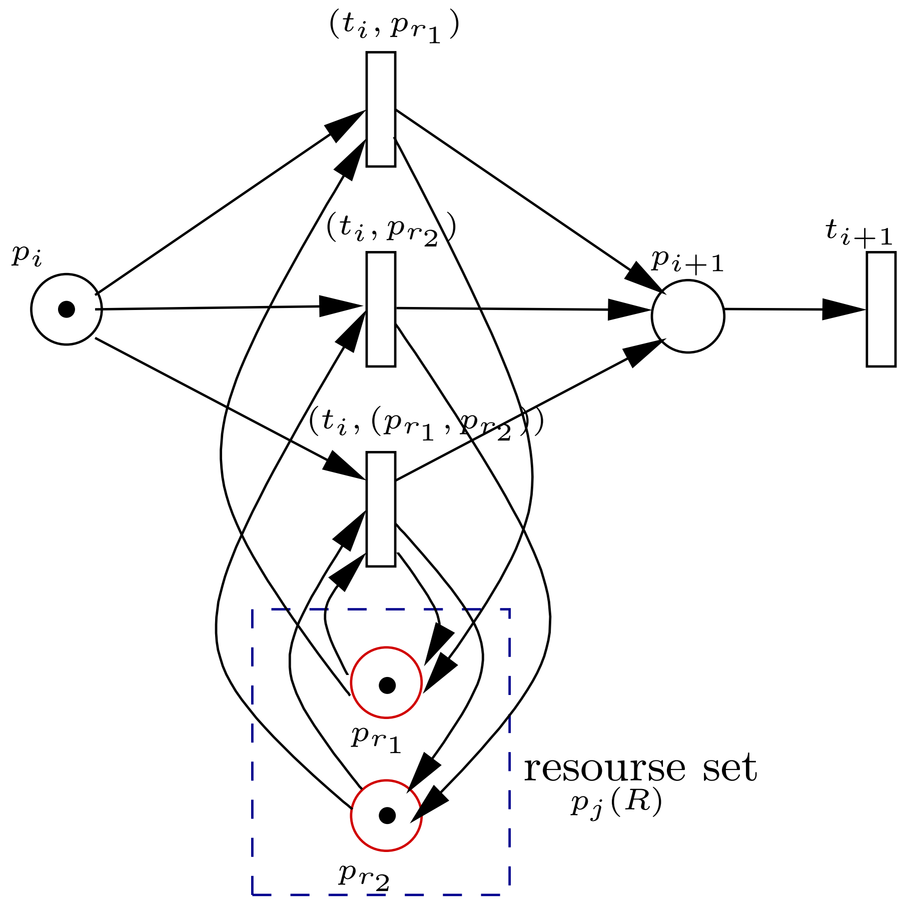


Figure 4.5: Multiple-resources of $MWF - wR$ in Petri net

Initially, it is supposed that there is one token in each resource places in the SPN. This token represents the cost of the resource and is mapped to c_j as the cost of resource r_j.

However, to demonstrate the mapping relations introduced in *MWF2SPN* in one complete example, example 4.1 is taken from Section 4.4 and the corresponding SPN model is presented in the following example 4.2.

Example 4.2:

Figure 4.6 shows the graphical Petri net model corresponding to example 4.1. The

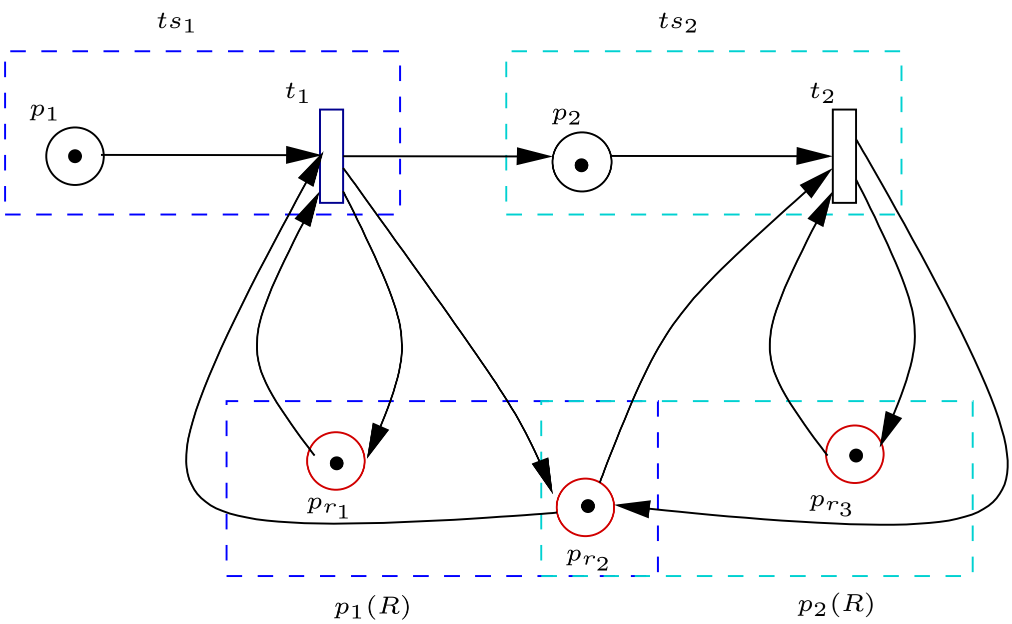


Figure 4.6: SPN model of example 4.1 $MWF - wR$ model

task ts_1 of the *MWF-wR* model is mapped into place p_1 and transition t_1 in the SPN model. The place p_1 is connected to transition t_1 by one input arc from p_1 to t_1. Task ts_1 followed by task ts_2. Task ts_2 is mapped into place p_2 and transition t_2 and connected by the input arc from p_2 to t_2. The flow relationship between these two tasks is mapped into an output arc from transition t_1 to place p_2. The resource set $p_1(R)$ consists of two resources r_1 and r_2 used to complete the processing of task ts_1, and they are mapped into two places p_{r_1} and p_{r_2} respectively. These two places are connected to transition t_1 which represents task ts_1 with two input arcs from p_{r_1} and p_{r_2} to t_1, and two output arcs from t_1 to p_{r_1} and p_{r_2}. The resource set $p_2(R)$ consists of two resources r_2 and r_3 used to complete the processing of task ts_2 which is represented by the transition t_2. These two tasks are mapped into two places p_{r_2} and p_{r_3} respectively. These two places are connected to transition t_2

by two input arcs from p_{r_2} and p_{r_3} to t_2, and by two output arcs from t_2 to p_{r_2} and p_{r_3}.

From the above description of the mapping of the tasks and resources of a business process to the places and transitions of *SPN*, the mapping relations can be distinguished between tasks and places and resources and places. The resource places are connected to transitions that represent the tasks using these resources in their processes with one input arc and one output arc each, while task places are connected to the transition of the task by only one input arc from a place to a transition. For instance, in Figure 4.6, the resource places p_{r_1} and p_{r_2}, each have one input arc and one output arc to transition t_1 of task ts_1, while place p_1 has only one input arc to transition t_1. And the same applies to the resource places p_{r_2} and p_{r_3} with the transition t_2 of task ts_2. The processing time for task ts_1 when it is processed by resource r_1 will be t_{11} and is mapped to the firing rate of transition t_1 when it connects to place p_{r_1}. Alternatively, it will be t_{12} when task ts_1 is processed by resource r_2 and mapped to the firing rate of the same transition t_1 when connected to place p_{r_2}. The firing rate is determined by adding a specific code for each resource to its rate of transition. The implementation of these mapping relations is presented in the next chapter.

4.7 Summary

This chapter has described the *MWF-wR* formalism for modelling business processes. The aim of the *MWF-wR* formalism is to allow quantitative evaluation algorithms to be applied as widely usable as possible. The main challenges faced in the design of the formalism are the existence of many types of algorithms and the absence of a standard model that can accommodate all of these types. This increases the difficulty and the complexity of constructing a formalism that meets these requirements. Section 4.1 of this Chapter presented an introduction. The fundamentals of modelling workflow were explained in Section 4.2, and the concepts of workflow models related to the *MWF-wR* formalism were explained in Section 4.2.1. The resources and tasks and relationships between them as well as the time they take are essential elements in performance analysis and the evaluation of business process models. Moreover, the characteristics and structure of business process workflows are explained in Sections 4.2.2 and 4.2.3 respectively. The requirements for modelling the

MWF-wR formalism were introduced in Section 4.3, which identified the parameters that influenced business process analysis. Section 4.4 focused on the definition of the proposed *MWF-wR* formalism. This definition specifies the main elements of business processes along with the relationships between them. For the purpose of performing the analysis and quantitative evaluation of the *MWF-wR* business processes in an automated way, the *MWF-wR* formalism needs to be mapped into SPN, and this issue was introduced in Section 4.5. before the mapping of the elements of the *MWF-wR* formalism into *SPN* was explained in details in Section 4.6. The next Chapter demonstrates the design and implementation of the software support tools for this framework used for business process modelling and quantitative evaluation.

Chapter 5

Software Tool Architecture

5.1 Introduction

This chapter gives a general review of the design and implementation of comprehensive support tools in the framework used for the quantitative evaluation of business process workflows. The chapter utilises the framework presented in Chapter 3 to design and implement software and support tools to provides a suitable environment for the implementation of the framework. The tools provide support for all of the steps involved in the implementation of the framework, and help the user for easily, efficiency and automated way of use. The main contribution of this chapter is to provide a structured framework for the implementation of the quantitative evaluation of business processes in the form of combined modules whose their elements are defined over the framework.

For the implementation of the framework described in Chapter 3, the main modules must be connected to provide an implementation layer for the quantitative evaluation of business process workflows. The framework consist of three main modules: a business process module, a stochastic model module and an algorithm module. The *MWF-wR* business processes model has been defined and then mapped into an *SPN* model using standard files to select the correct input to algorithms of quantitative evaluation and optimisation of business processes. A support tool is developed to map the *MWF-wR* for business processes into an *SPN* model, and the *CSPL* file of *SPN* is augmented to include the components of the model. This software tool provides the mapping relations from the *CSPL* file of *SPNP* to Matlab for the implementation of the algorithms. Mapping relations are adopted which perform all of the steps in an automated way, overcoming any incompatibility between the different formats of files used in each module.

To achieve this, several practical steps need to be conducted. This chapter present details of these steps. The remainder of this chapter is structured as follows. Section 4.2 presents a set of functional and non-functional requirements for the software of the support tool architecture in the framework. Section 4.3 discusses the assumptions made for the software architecture. The design of the software is then outlined in Section 5.4. Section 5.5 depicts the implementation of the design components. Finally, the chapter is briefly summarised and conclusions gives in Section 5.6.

5.2 Requirements of Support Tool for Framework Design

This section discusses the requirements for the design of software in the framework. The requirements of software architecture used in this work refer to "the determination of the information, processing and the characteristics of that information and processing required by the user of the system" [126]. These requirements can be classified into functional and non-functional aspects of the software architecture. Functional requirements denote the set of functions which summarise the information and processing required in the framework, whereas non-functional requirements denote the qualitative properties that can be used to assess the efficacy of the functionality provided by the framework [127, 128, 129]. The following section states the requirements of the software architecture explained in this chapter.

5.2.1 Functional requirements

The following are functional requirements for the design architecture of the software tool in the framework:

1. **Automated**:
   The tool has to be designed in such a way that the execution of the framework can be automated. It has to employ the framework modules of chapter 3 in software engines that perform the modules functionality automatically. The

automation can be achieved by using different parsers which implement the functions that each module requires. The output and input of these modules have to be available in a machine-readable format. This support to achieve a higher level of automation. Having such interchangeable inputs and outputs allows the parsers to update, read or write their contents automatically.

2. **Formal readable format of business processes**:

   A business processes model should be available in a formal, machine-readable and understandable format. The model must provide a language for the explicit representation of business processes formulated as clear semantics. The business processes should be model so that the components are defined in precise semantics and expressed in a verifiable and enforceable manner so as to be able to be designed and implemented in as automated a way as possible.

3. **A structured format for business process**

   The software architecture must provide a structured format for the explicit representation of business processes model.

4. **Stochastic tools**: The tool used to build a stochastic Petri net model must have standard structure file used to generate standard customer file for business process model.

5.2.2 Non-functional requirements

The non-functional requirements of the software tool are characterised as follows:

1. **Usability and simplicity**:

   The steps followed in the use of the tool have to be clear and sufficiently simple and have a logical sequence. The tool's interface also has to provide some description of what the user has to do in each step (i.e. user friendly).

2. **Extensibility**:

   The design of the tool should allow new components to be easily added which represent either business processes or algorithms. The framework architecture must be easily extensible to add new component for business process model, in addition, any new metrics used for quantitative evaluation must not require any changes to the framework.

3. **Adaptability to changes in objectives**

   One of the main contributions of this research that the framework used for the quantitative evaluation of business processes is a generic framework. This means that it should be compatible or work with any business process model. In other words, the framework should be has the capability to introduce the quantitative evaluation for any business processes.

5.3 Assumption

The design of the software support tools presented in this chapter is built upon set of assumptions based upon the requirements described in Section 5.2 and those of the framework defined in Chapter 3. The assumptions for the software architecture are presented in what follows:

- **Business process model**:

  A business process model is predefine in a formal technique. The model assumed to be define a set of outcomes business process components of algorithms of quantitative and optimisation evaluation.

- **Stochastic model**:

  A stochastic analysis tool assumed to be available and has standard file which is used for mapped the business process model onto *SPN*.

- **Mathematical language**:

  A high level mathematical language should be available to implement the algorithms.

5.4 Design Components

The design architecture of the software support tool describes the "modularization and detailed interfaces of the design elements, their algorithms and procedures, and the data types" [126]. This section, presents the design components of support tool that automates the framework presented in chapter 3. The components are depicted from a user's perspective in Figure 5.1, which consists of three modules (components) that flow from left to right. These are the business process workflow

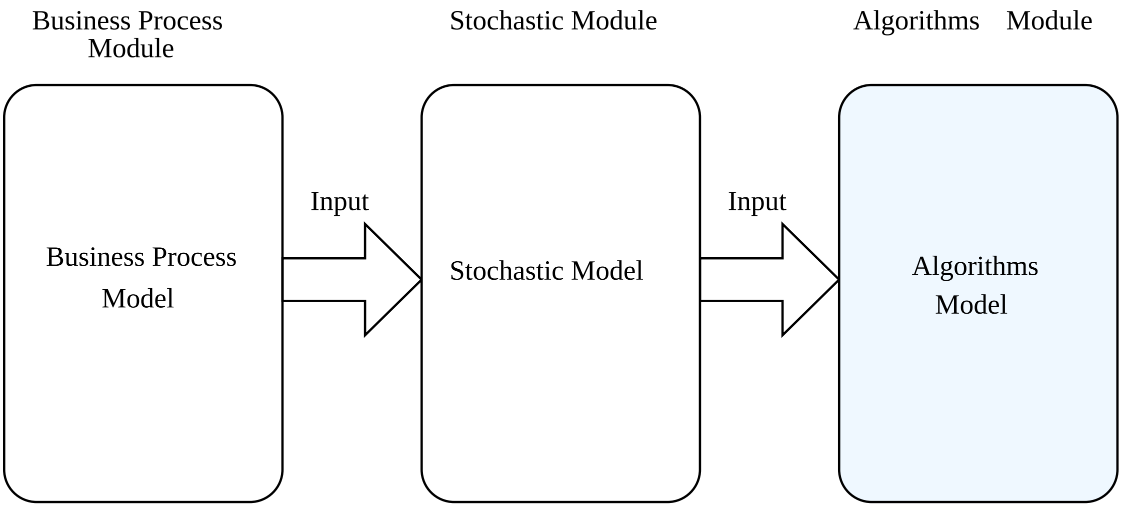


Figure 5.1: Framework component from engineering perspective

module, the stochastic module and the algorithm module. In this design, the business process mapped into stochastic model, then this model is mapped to algorithm model to conduct the quantitative and optimisation approach. Figure 5.2 depicts the components of the framework presented in chapter 3 and shown in Figure 5.1 from tool support perspective. The input, output and functionality of each component is explained in the following sections.

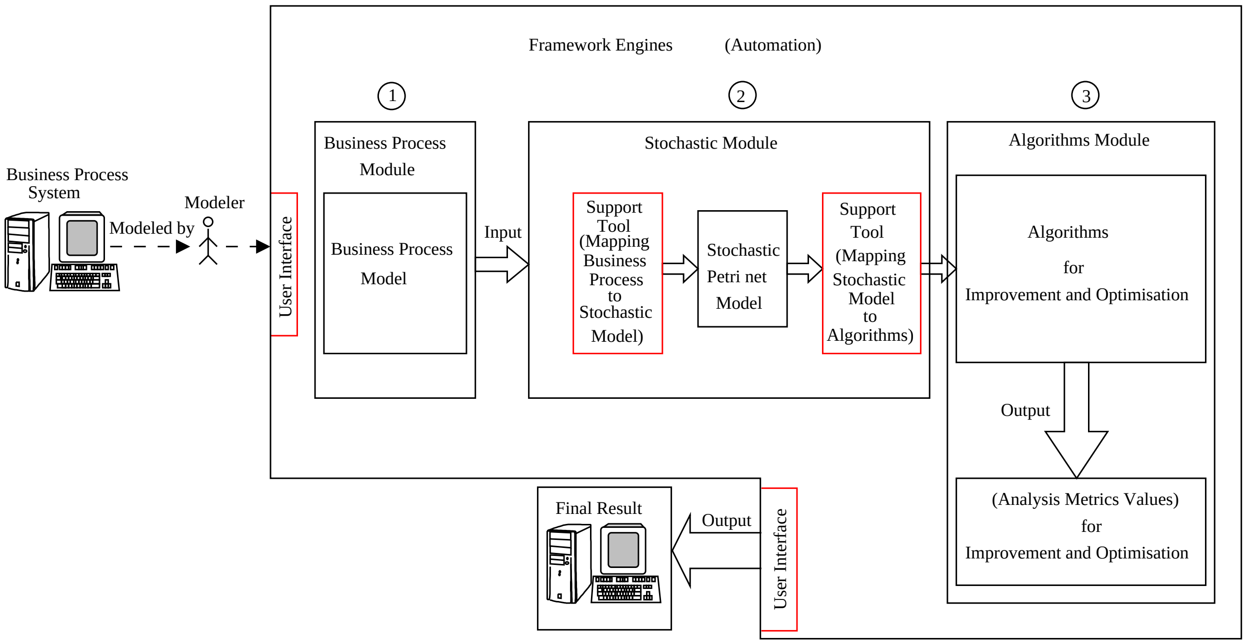


Figure 5.2: Framework components from support tool designer perspective

1. **Business process module**

   In this module, business process elements are specified, and the model is defined and created. There are many techniques and language for modelling business process. The model of business process is defined in this module to identify the relevant elements that be used for quantitative evaluation objectives. The *MWF-wR* formalism presented in Chapter 4 can be used as a generic formalism for modelling any business process. However, the input, output and functionality of this module are as follows:

- **Input**: The input to this module is any business process model used for quantitative and optimisation evaluation approach.
- **Output**: The machine readable business process model have formal syntax and semantic.
- **Functionality**: The business process model presented by modeler which represent business process system in accordance with selected techniques. Each business process model should include the following elements:
  - Tasks: Any business process model is designed to represent a business system which seeks to achieve specific goals. Each business process consists a number of tasks that need to be carried our according to set of conditions. The tasks are logical elements of business processes that are carried out by resources. Thus, tasks are regarded as essential elements of any business process model.
  - Resources: The resources are used to carry out tasks. A task can be carried out by one resource or a set of resources. But one resource can carry out only one task at a time.
  - Resource cost: For the purpose of the quantitative evaluation of business process, the cost of resources is important factor which should be considered, since the final aim of improvement and optimisation is to minimise costs.
  - Processing time: Each task is associated with a processing time to complete specific actions. Processing time depends on the resources used. This means that each task has a processing time associated with resources that have specific costs.

The output of this module is a business process model used as the input to the next module. The model will be mapped to a stochastic Petri net model to facilitate the automated implementation of the framework.

2. **Stochastic Petri net module**

   In this module, a business process model is mapped into a stochastic Petri net model that has a standard output file format. This model is used as a solver to map the elements of a business process model to corresponding elements in this file. The stochastic Petri net model is generated by using appropriate stochastic analysis tool. From a multitude of general formalisms which can specify stochastic Petri net models, stochastic Petri net package (SPNP) were chosen in this study. Stochastic models such as stochastic Petri nets are powerful enough techniques with the formal formalism required to map business processes into. The input, output and functionality involved in mapping business process model into the stochastic Petri net model are as follows:

   - **Input**: The input is a business process model.
   - **Output**: A stochastic Petri net model have standard structure file and contain business process information.
   - **Functionality**: The components of the business process model are mapped into stochastic Petri net form. Each task in the business processes model is mapped into places and transitions, each resource is mapped into a place, and the relationships between tasks are mapped into arcs.The relations for mapping business process components to the *SPN* are as follows:

     (a) **Mapping of task and processing time**:

     The tasks and processing time in a business process model are mapped into *SPN* in the following way:

     • Each task is mapped to one place and one transition. One arc connects each place and transition. The processing time for the task is mapped to a rate of transition of the task.

     (b) **Mapping resources and costs**:

     The resources in a business process are mapped to *SPN* in the following way:

     • Each resource is mapped to one place. The resource processes the task in a specific time. Two arcs connect the place of a resource to the transition of a task. One is the input arc and the other is the output arc. The cost of the resource mapped to the token of the place. These mapping relations are explained later.

3. **Algorithm module**

   In this module, the outcome of the stochastic Petri net model is used as the input for algorithms used to improve and optimise business processes. The input, output and functionality of the mapping relation in this module are as follows:

   - **Input**: Stochastic Petri net which comprises information of business process components .
   - **Output**: The final results of quantitative evaluation of quantitative and optimisation approach of business processes.
   - **Functionality**: The output file from the stochastic Petri net module has standard structure file. The file contains all information from the business process model mapped into this file in the previous module. The elements of this file are then mapped to the corresponding parameters of algorithms of quantitative and optimisation evaluation of business processes. The elements of stochastic Petri net model file and parameters of business process algorithms are as follows.
     - *place* and *transition* in the *SPN* are mapped to *task* in the business process model algorithms.
     - The *rate value* of a *transition* is mapped to the *processing time* of the corresponding *task*.
     - *place* in *SPN* is mapped to *resource* in business process algorithms.
     - *place marking* is mapped to the *cost* of the corresponding *resource*

The user will only upload the business process document which describes the business processes model and all of the details of the mapping process from business processes to the *SPN* and from the *SPN* to the algorithms module are hidden. The following section describes the implementation of each design component.

5.5 Implementation

This Section describes the implementation of the design components of the software support tools presented in Section 5.4. The implementation provides representation of the algorithms and the data types described in design section which allow the functional and non-functional requirements of the software to be satisfied [126]. The software is used for mapping between different types of formats of the components. The implementation of the design components of the framework is built in Java as object oriented programming language using the Eclipse platform. Figure 5.3 present the components of the framework from modeler perspective which are as follows.

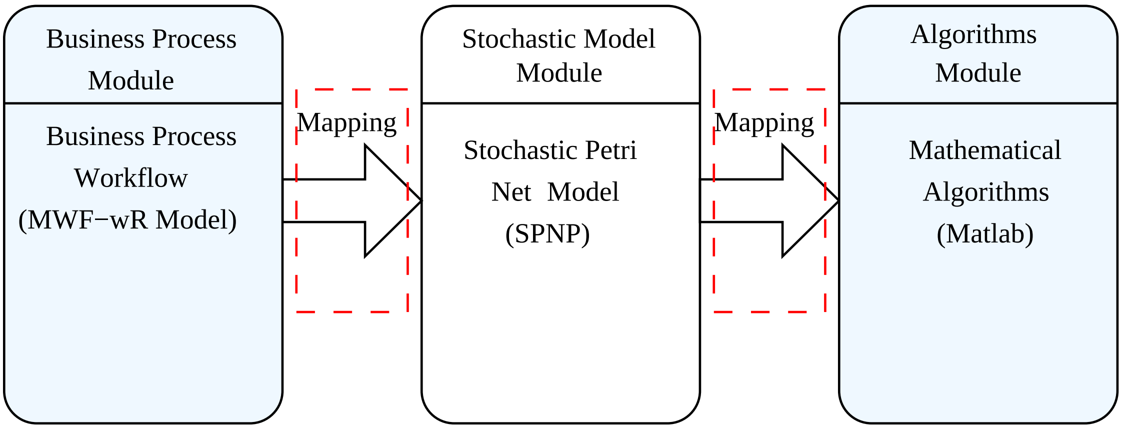


Figure 5.3: Software support components form modeler perspective

- **The MWF-wR** is used as a generic formalism for the modelling of any business process
- **Stochastic Petri nets** is used as a modelling tool for mapping the **MWF-wR** model into.
- **Stochastic Petri net Package** is used as a tool for build the *SPN* model
- **Matlab** is used as a high level mathematical language in implementing the different types of algorithm used for quantitative evaluation and optimisation of business processes.

Before describing the implementation of the design components, the requirements of this implementations are first explained.

5.5.1 Implementation requirements

The following are the requirements of the implementation of the design components presented in the previous section:

- **Business process model**: A business process model expressed in the MWF-wR form is used as a standard model for any business process. Therefore, a*MWF-wR* business process model needs to be created first.

- **Mapping the MWF-wR model into SPN**:
  An algorithm for mapping the MWF-wR business process model into stochastic Petri net is required.

- **Stochastic Petri net**:
  SPNP is used as a tool for *SPN* modelling. *SPNP* is based on standard file which is a *CSPL* (C-based SPN Language) file.

- **Mapping SPNP into algorithms**:
  Mapping relations are used to map the *CSPL* file of the *SPNP* tool into the parameters of algorithms expressed in Matlab.

5.5.2 The implementation of the components

Figure 5.4 depicts the implementation of the three modules of the framework from support tools perspective. This figure consist of three modules that flow from left to right. The *MWF-wR* business process model is mapped into *SPN* according to the *MWF2SPN* algorithm presented in Chapter 4. The *SPN* model is represented by the *CSPL* file of *SPNP* tool which is used as a solver to map the *MWF-wR* into *SPN*, and then to map the *SPN* into Matlab. The outcomes of the *SPNP* file is used as an input parameters to the algorithms of business process quantitative evaluation and optimisation approach.

5.5.2.1 Mapping *MWF-wR* model into *SPN*

This section describes the implementation of the mapping of a business process defined in *MWF-wR* into an *SPN* model. This tool used a *MWF2SPN* algorithm designed to map the *MWF-wR* model by means of a set of mapping relations and operations. The implementation of mapping relations was conducted by using a

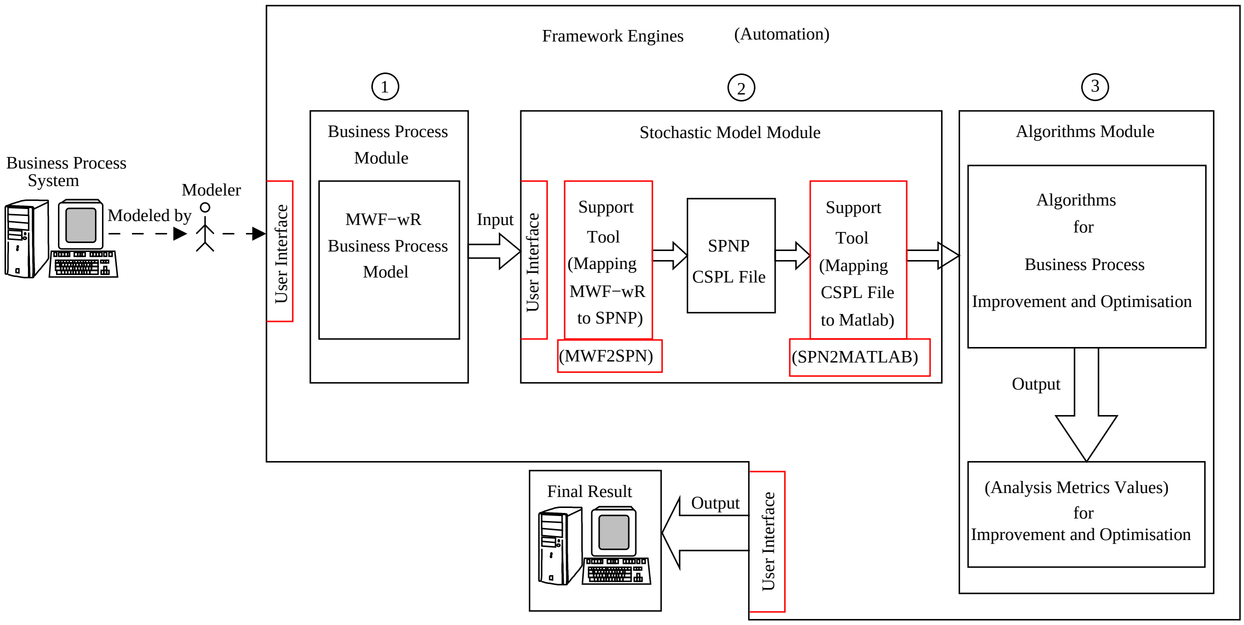


Figure 5.4: The framework implementation components

stochastic Petri net package *(SPNP)* as modelling tool to build the stochastic Petri net model. *SPNP* creates files of a standard structure using a *CSPL* (C-based SPN language) specific to *SPNP*, which, for ease of reference, is here called the *SPNP* file. A textual *SPNP* interface is required for model definition, since the *MWF-wR* component is eventually mapped into the textual code of an *SPNP* file . This can be done through a dedicated graphical user interface (GUI), which allows the modeler to insert the *MWF-wR* model and automatically produce the *CSPL* file. The *CSPL* output file contain all of the available functions and variables as well as those inserted by the modeler. The output *CSPL* file describes the *SPN* model which contains all of the information about the business process from the *MWF-wR* model with all *CSPL* functions and variables used to describe the model.

Mapping the *MWF-wR* components into *CSPL* file elements requires the elements of the file to be identified which used will be in the standard file to represent the *SPN* model. The stochastic Petri net package (SPNP) is a tool used in the modelling and analysis of stochastic Petri nets. It creates files with specific standard structure in a *CSPL*. The file must contain five basic functions [113], each of which is designed to carry out one or several tasks by using some relevant functions. The five functions used in the mapping relations are described as follows:

- *option()*: used to carry out certain tasks by calling some relevant function which affect the way in which the stochastic reward net (SRN) is described and solved.

- *net()*: used to define an SRN.
  This function is calling the following functions:
  - *place()* and *init()*: the former is used to define a place with a name p; and the later defines the initial number of tokens in place p to be n.
  - *imm()*: this function defines the time transition
  - *rateval()* and *probval()*: the first of these functions defines the firing rate of the timed transition t and the firing weight or probability of an immediate transition t as a constant value val. The function *probval()* needs to be used only if the value of the firing weight of the immediate transition is different from the default value 1.0.
  - *priority()*: this defines the priority of transition.
- *assert()*: a boolean marking function used to check the validity of each newly found marking.
- *ac_init()*: used to call a set of functions before starting the construction of the reachability graph to output information about the model to the "*.out*" file of *SPNP*.
- *ac_reach()*: used to call a set of functions after the construction of the reachability graph is completed to output information about it to the "*.out*" file of *SPNP*.
- *ac_final()*: this function calls a set of functions designed for the user to flexibly define outputs; for example, a function to solve the Markov chain numerically at time t or for *steady state analysis*, and a function to output data about the Markov chain and its solution.

The *MWF-wR* business process model has an output file in Matlab format defined in Table 5.1. The file comprises components defined in definition 4.1 in Chapter 4. Having the *CSPL* file of *SPNP* and the *MWF-wR* file, the implementation of the mapping relations for each component from *MWF-wR* to a *CSPL* file using the *MWF2SPN* algorithm presented in Chapter 4 are now described.

Table 5.1: MWF-wR file

```
S = ([structure])
field1().string = 'task';  value1 =  string();
filed2().string = 'resource';  value2 = string();
field3().double = 'process time'; value3 = double();
field4().double = 'cost';  value4 = double();
```

S (MWF−wR) =$('task'_i, 'value'_i, 'resour'_{ij}, 'value'_{ij} cost'_j, 'value'_j, 'processtime'_{ij}, 'value'_{ij})$

- **Task mapping into CSPL file**

  Each task ts_i of the *MWF-wR* model is mapped to a place ts_i in the *CSPL* file. Place ts_i is connected by an input arc to transition ts_i in the *void net*() function. Table 5.2 shows the format of a *CSPL* file for mapping the task of *MWF-wR* into *SPN*. The following mapping relations are used:

  1. The task ts_i is mapped to the *CSPL* file at place ("ts_i") in line 9 of the function *void net*() in line 7, and to the timed transition ts_i of the function rateval ("ts_i", process time"s_i") in line 13.
  2. The input arc in the function iarc "*transition*(ts_i)","*place*(ts_i)" in line 16 of the same function *void net*() is connect the place ts_i to the transition ts_i.
  3. The processing time s_i of the task ts_i is signified by the rate of the time transition "ts_i" in the second variable *process time"s_i"* of the function *rateval()* in line 13. This is defined as a variable process time"s_i" of the function *global variable* in line 2.

- **Resource mapping into the CSPL file**

  Each resource r_j of the *MWF-wR* model is mapped to a place r_j in the *CSPL* file and connected by input and output arcs to the same transition ts_i. This transition stands for task ts_i which uses resource r_j for processing. The cost c_j of resource r_j is mapped to a reward function defined in the *C Function* of the *CSPL* file. The function returns a number of tokens to place r_j multiplied by a number that stands for the cost c_j of the resource. The value 1 is selected as a default value for the number of tokens in place r_j. Table 5.3 shows the format of the mapping of resource r_j used to process the task ts_i and the cost of the resource is c_j, and consists of the following mapping relations:

  1. The resource r_j is mapped to the function place "r_j" in line 11 which is

Table 5.2: *CSPL* file of mapping *MWF-wR* task

1. /* global variables */

2. double (process time $”s_i”$)=1;

3. /* Prototype for the function(s) */

4. /* ================ OPTION ================ */

5. void options () { }

6. /* ========= DEFINITION OF THE NET =============== */

7. void net () {

8. /* ====== PLACE ====== */

9. place $(”ts_i”)$;

10. /* ====== TRANSITION ====== */

11. /* Immediate Transition */

12. /* Timed Transition */

13. rateval $(”ts_i”, \ (\text{process time}”s_i”))$;

14. /* ====== ARC ====== */

15. /* Input Arcs */

16. iarc $(”transition(ts_i)”, ”place(ts_i)”)$;

17. /* Output Arcs */

18. }

related to the function *void net*() in line 8.

2. The input arc in function iarc (“$transition(ts_i)$”,”$place(r_j)$” in line 19, and the output arc in function iarc (“$transition(ts_i)$”,”$place(r_j$” in line 21 are connect the place r_j to the timed transition ts_i in line 15.

3. The function *double ()* in line 4 defined by the function *double cost*(c_j) *() return (mark(“place*(r_j)*”) * c_j);* in lines 24 and 25 which returns the number of token in place r_j multiplied by the variable c_j which stands for the cost c_j of resource r_j.

Table 5.3: Format for mapping *MWF-wR* resource into *CSPL* file

1. /* global variables */
2. double process time"s_i" $= 1$;
3. /* Prototype for the function(s) */
4. double $cost(c_j)$ ();
5. /* ================ OPTION ================ */
6. void options () { }
7. /* ========= DEFINITION OF THE NET =============== */
8. void net () {
9. /* ====== PLACE ====== */
10. place ("ts_i");
10. init ("ts_i", 1);
11. place ("r_j");
10. init ("r_j", 1);
12. /* ====== TRANSITION ====== */
13. /* Immediate Transition */
14. /* Timed Transition */
15. rateval ("ts_i", (process time"s_i"));
16. /* ====== ARC ====== */
17. /* Input Arcs */
18. iarc ("$transition(ts_i)$", "$place(ts_i)$");
19. iarc ("$transition(ts_i)$", "$place(r_j)$");
20. /* Output Arcs */
21. oarc ("$transition(ts_i)$", "$place(r_j)$");
22. }
23. /* C Functions */
24. double $cost(c_j)$ () {
25. return (mark ("$place(r_j)$")) $* \; c_j$)
26. }

The mapping relations given above show how a business processes defined in *MWF-wR* can give rise to a *CSPL* file of the *SPNP*. The file contains all of the functions and variables that are used to define the components of the *MWF-wR* business process model. The tool parses the file and extracts the information required to select the correct input parameters for algorithms used for business process improvement and optimisation. These parameters need to be mapped into Matlab, because this high mathematical language is used for the implementation of different types of algorithms of business process quantitative evaluation and optimisation approaches. The next section describes the mapping of the *CSPL* file into Matlab.

5.5.3 Mapping the business process model into Matlab

Having created an enriched *CSPL* file with information from the business process model, they need to be fed into many different types of Matlab algorithms for quantitative evaluation and optimisation. So, an algorithm to mapping the *CSPL* file into Matlab is required. This section describes the design and implementation of the *SPN2MATLAB* algorithm, which has been designed to achieve the necessary mapping. *SPNP* produces a *CSPL* customer file model which is a C-base standard file containing a description of of the *MWF-wR* business process model. The mapping tool reads and parses this file to identify the information relevant to each Matlab parameter. Firstly, the parameters for the algorithms that are required in the present study of business process analysis and optimisation are identified. Two approaches are selected to demonstrate implementation. The "Availability weak point analysis over an SOA " [20], and the "Grid workflow scheduling in wose" [21]. These are chosen rather arbitrarily. However, each approach has specific algorithm parameters, they have common parameters. The following are the parameters common to both:

1. A set of tasks $\{ts_i\}$ defined in the business process model.
2. A set of resources $\{r_j\}$ available in the business process model.
3. A subset of resources $p(R)$ associated with each task; for example, the $p_i(R)$ resource set used to complete the processing of task ts_i.
4. The value of the cost of each resource of the business process; for example, c_j is the cost of resource r_j.
5. The processing time s of each task processed by a specific resource; for example, s_{ij} is the processing time of task ts_i by resource r_j.

The complete parameters of each approach are explained in the next chapter. The algorithms are implemented in Matlab, and the *SPN2MATLAB* algorithm is introduced to specify the methodology used for mapping the *CSPL* file into Matlab. The algorithm is shown in Table 5.4.

Table 5.4: SPN2MATLAB algorithm for mapping *CSPL* to Matlab

Step 1: Read file (CSPL file of SPNP)
Step 2: choose tasks count M and resource count N
Step 3: For each line of file until (End of file) do:
Step 4: For all $Place()$ function do:
Step 5: For all $iarc()$ function do:
Step 6.1: If word ("place name") = word ("place name")
Step 6.1.1: For all $oarc()$ function do:
Step 6.1.2: If word ("place name") = word ("place name")
Step 6.1.3: ("place name") = ("resource name"),
Step 6.1.4: ("transition name") = ("task name")
Step 6.1.5: Else ("transition name") = ("task name")
Step 7.1: For all $rateval()$ function do:
Step 7.2: ("transition name") = ("task name")
Step 7.3: ("place name") = ("resource name")
Step 7.4: ("rate value") = ("processing time")
Step 8.1: For all C functions do:
Step 8.2: ("place name") = ("resource name")
Step 8.3: ("mark value") = ("resource cost")
Step 9: End

To illustrate the implementation of the *SPN2MATLAB* algorithms, the *CSPL* file used as example 4.1 in Section 4.4 is again considered. Example 4.1 is a *MWF-wR* business process model which consist of two tasks and three resources used to process these tasks. The first task ts_1 is processed by resources r_1 and r_2, and the second task ts_2 is processed by resources r_2 and r_3. Table 5.5 depicts the *CSPL* file of this example. The following explains the mapping of each elements of the file into Matlab parameters:

1. The place “p_1“ in line 11 is connected to the transition “t_1“ by only one input

Table 5.5: CSPL file of Example 4.1

/* global variables */

/* Prototype for the function(s) */

double cost_r_1 ();
double cost_r_2 ();
double cost_r_3 ();
/* ================ OPTION ================ */

void options () { }

/* ========= DEFINITION OF THE NET =============== */
void net () {

/* ====== PLACE ====== */

place ("p_1"),
init ("p_1", 2);
place ("p_2",
init ("p_2", 2);
place ("p_{r_1}");
init ("p_{r_1}", 1);
place ("p_{r_2}");
init ("p_{r_2}", 1);
place ("p_{r_3}");
init ("p_{r_3}", 1);

/* ====== TRANSITION ====== */

/* Immediate Transition */

/* Timed Transition */

rateval ("t_1", 8, "p_{r_1}");
rateval ("t_1", 6, "p_{r_2}");
rateval ("t_2", 7, "p_{r_2}");
rateval ("t_2", 2, "p_{r_3}");

/* ====== ARC ====== */
/* Input Arcs */
iarc ("t_1", "p_1");
iarc ("t_2", "p_2");
iarc ("t_1", "p_{r_1}");
iarc ("t_1", "p_{r_2}");
iarc ("t_2", "p_{r_2}");
iarc ("t_2", "p_{r_3}");

/* Output Arcs */
oarc ("t_1", "p_2");
oarc ("t_1", "p_{r_1}");
oarc ("t_1", "p_{r_2}");
oarc ("t_2", "p_{r_2}");
oarc ("t_2", "p_{r_3}");
}

/* C Functions */
double cost_ r_1 () {
return (mark (("p_{r_1}") * c_1)
}
double cost_ r_2 () {
return (mark (("p_{r_2}") * c_2)
}
double cost_ r_3 () {
return (mark (("p_{r_3}") * c_3)
}

arc iarc (“t_1“, “p_1“) in line 30. So, this place stands for the task identified by ts_1. This place is mapped to the task ts_1 in Matlab.

2. The same applies to place “p_2“ in line 13 and transition “t_2“, which are connected by only one input arc iarc (“t_2“, “p_2“) in line 31. Therefore, this place stands for the task identified by ts_2 and mapped to task ts_2 in Matlab.

3. The place “p_{r_1}“ in line 15 is connected to transition “t_1“ by two arcs. The first is the input arc iarc (“t_1“, “p_{r_1}“) in line 32, and the second is the output arc oarc (“t_1“, “p_{r_1}“) in line 38. So, this place stands for the resource identified by r_1 used for the process task ts_1. This place is mapped to the resource r_1 in Matlab.

4. The place “p_{r_2}“ in line 17 is connected to the transition “t_1“ by two arcs. The first is the input arc iarc (“t_1“, “p_{r_2}“) in line 33, and the second is the output arc oarc (“t_1“, “p_{r_2}“) in line 39. So, this place stands for the resource identified by r_1 and used for the process task ts_1. In addition, this place if connected to the transition “t_2“ by two arcs. The first is the input arc iarc (“t_2“, “p_{r_2}“) in line 34, and the second is the output arc oarc (“t_2“, “p_{r_2}“) in line 40. This means that the place “p_{r_2}“ which stands for the resource identified by r_1 is used for process task ts_2 as well. This place is mapped to the resource r_2 in Matlab.

5. The place “p_{r_3}“ in line 19 is connected to transition “t_2“ by two arcs. The first is the input arc iarc (“t_2“, “p_{r_3}“) in line 35, and the second is the output arc oarc (“t_2“, “p_{r_3}“) in line 41. So, this place stands for the resource identified by r_3 and used for the process task ts_2. This place is mapped to the resource r_3 in Matlab.

6. The number m of all tasks in the model equals the sum of all tasks specified in the model.

7. The number n of all resources in the model equals the sum of all resources specified in the model.

8. The number of resources used to process each task, subset p(R)) is equals to the number of tokens in each task place. For example, in line 12, the second variable of function *init* “p_1“, 2, the number 2 denotes the number of resources

in the subset $p_1(R)$ that is used to process task ts_1 represented by this place. And the same applies to the second variable of the function *init* "p_2", 2 in line 14, where the number 2 equals the number of resources in the subset $p_2(R)$ that is used to process task ts_2 represented by place "p_2".

9. The cost of each task is mapped to the variable of the function defined in *C Functions* as *double ()*. For each resource place a function is defined that returns the number of tokens in the place. In line 44, the function *double cost_*r_1() of *C Functions* in line 43 is defined in *Prototype for the function(s)* as *double* $cost_{r_1}$ in line 3. This returns the number of token in the place "p_{r_1}" multiplied by the variable c_1. This variable stands for the cost c_1 of the resource r_1. The same is true for the function *double cost_*r_2() in line 47 of *C Functions* in line 43 which is defined in *Prototype for the function(s)* by *double* $cost_{r_2}$ in line 4. It returns the number of tokens in place "p_{r_2}" multiplied by the variable c_2, the cost of r_2. And the function *double cost_*r_3() in line 50 of *C Functions* in line 43 is defined in *Prototype for the function(s)* as *double* $cost_{r_3}$ in line 5. This returns the number of token in place "p_{r_3}" multiplied by the variable c_3, which stand for the cost of the resources r_3.

10. The processing time of each task is mapped to the rate of transition that represents this task. The processing time s_{ij} denotes the processing time of task ts_i by resource r_j. In line 24, the second variable of the function *rateval* ("t_1", 8*;* is mapped to the processing time s_{11} of task ts_1 when it is processed by resource r_1. This is defined by adding the place "p_{r_1}" of resource r_1 to this function as shown in line 24. The same applies to processing time s_{12} of task ts_1 when it is processed by resource r_2. It is mapped to the number 6, which is the second variable of the function *rateval* ("t_1", 6)*;* which is distinguished by adding the place "p_{r_2}" of resource r_2 to this function, as shown in line 25. The same is also true for s_{22} and s_{23}, the processing times of task ts_2 when processed by resources r_2 and r_3 respectively. In line 26, the second variable of the function *rateval* ("t_2", 7) is mapped to the s_{22} processing time of task ts_2 by resource r_2, and the number 2, the second variable in line 27 of the function *rateval* ("t_2", 6 is mapped to the s_{23} processing time of task ts_2 by resource r_3.

After mapping the elements of the *CSPL* file to Matlab parameters and creating a file which contains these parameters, the file will be used as input for algorithms for availability enhancement [20] and workflow scheduling [21]. In the next chapter, implementations of the two approaches above are used as case studies to investigate the viability and validity of the framework proposed in this study.

5.6 Summary

This chapter has presented the software support tool developed for the proposed framework for the quantitative evaluation of business processes. The design and implementation of the software provides an environment where the output results of algorithms for the quantitative evaluation and optimisation of business processes to be automatically produced. The main contribution of this chapter is develop software support tools used to implement the framework in automated way. This is carried out in two main stages. Firstly, a support tool for the *MWF2SPN* algorithm which maps the *MWF-wR* business process model into a *CSPL* file of *SPN*. This is achieved by the design and implementation of a format to map *MWF-wR* business process model into *SPN* so as to produce a *CSPL* file with a standard structure. This file is used as the input to the second stage. The second stage is a support tool designed for the *SPN2MATLAB* algorithm used for mapping the *CSPL* file into Matlab. This stage is achieved by the design and implementation of a format to map the *CSPL* file into Matlab and to produce a file which contains the necessary parameters used by algorithms for business process quantitative evaluation and optimisation. In the next chapter, the efficacy of the support software tool is demonstrated in an empirical study of two selected approaches for enhance the availability and scheduling of resources of business process workflows.

Chapter 6

Experimental Results

6.1 Introduction

This chapter describes the experimental studies conducted to demonstrate the capability of the general solution presented in Chapter 3. They demonstrate the operations of the framework's mechanisms and the performance of the tools developed. Many techniques are used in the literature for the purpose of business process improvement and optimisation, and each technique uses different algorithms. The experiments described here two different algorithms. The first study was conducted to validate the framework by conducting an analysis of the availability of business process resources using the research published by Xie et al. on availability weak point analysis over service-oriented architecture deployment [20] as a practical example. The second study was conducted on workflow scheduling used on the work by Patel et al., grid workflow scheduling in workflow optimisation services for e-science (WOSE) [21] as the other practical example. In both cases, the methods proposed do not support the analysis of all business processes in a systematic manner, which why these two techniques were chosen. However, the methodology used for conducting those two published studies is applied based on the methodology used in the main framework proposed in this research as described in Chapter 3. The aims of the experiments are to demonstrate the capability of the framework and the performance of the support tools developed to get the same results of the examples used in this work. Here the business process model is mapped into a stochastic Petri net using the *MWF2SPN* algorithm, and then the stochastic Petri net model is mapped into the algorithm module using the *SPN2MATLAB* algorithm. The methodologies applied in these two experimental studies is explained below.

The remainder of this chapter is organised as follows: Section 6.2 outlines the first study, availability weak point analysis over an SOA deployment scenario which is used as the first experimental study. Section 6.3 introduces the modules of the framework applying to the first study concerned with resource availability and the results of its implementation are presented. Section 6.4 explain the second experimental study, grid workflow scheduling in WOSE Scenario. Section 6.5 describes the steps involved in applying the framework to workflow scheduling and the results. Section 6.6 then summarises this chapter.

6.2 Enhancement of Availability of Resources

The validity of the framework was firstly tested by applying it in the availability enhancement as a quantitative evaluation approach to improve business processes. We chose the study presented in [20] as example for this approach. The implementation of the study, introduced a work titled "automated generic weak point analysis of business process" [18]. Business process services doing works and have users often spread out across the globe and requiring near 24/7 availability. Presented high availability of resources is one main approaches to improve business processes. This allows IT managers to balance the operational and economic costs of business processes and the resources used in organisations. Delivering the right level of availability of IT infrastructure allows an organization to balance the reliability of the execution of business process with the cost of resources. Exceeding the appropriate level of availability results in more expensive services, while insufficient availability results in costly outages. Organizations seek to be able to predict the reliability of business processes by ensuring the availability of distributed resources while keeping the overall costs close to minimum possible.

6.2.1 Availability weak point analysis over an SOA deployment scenario

In this section, a methodology used in business process performance evaluation to enhance the availability of workflows resources is described. A method called (availability weak point analysis over an SOA deployment framework) is discussed to determine the critical components for the successful completion of a business process [20]. The requires the modeller to identify the frequency with which components in the workflow are used, but this task is not automated and may require considerable computational effort. The analyst needs to identify by hand how often a business process uses certain resources. The framework used in this study improves this situation by automatically deriving the usage patterns of resources from business process specifications.

The availability of a single resource can affect the overall availability of resources. The weak point methodology determines the resources of the workflow and satisfy the predefined availability requirement level. The modeler first extracts the relevant information from the business process specifications and uses them as input along with the availability requirements for each workflow. The workflow specifications concerning the resources involved determines how often they are used. The relationship between workflow and resources is specified in a workflow-resource mapping matrix. Based on this mapping matrix and information on the current availability of resources, the analysis calculates whether or not the specified availability level has been reached.

The methodology used in this analysis can be explained as follows. Suppose there exist n workflows, $W_1, W_2, \cdots, W_n$, and the availability requirements for each workflow are $P_1, P_2, \cdots, P_n$. Suppose there exist m resources, $r_1, r_2, \cdots, r_m$, and the availability of each resource is $P(r_1), P(r_2), \cdots, P(r_m)$. The mapping matrix showing relationships between workflows and resources is depicted in Table 6.1. The

Table 6.1: Workflow-resource relationship matrix

	r_1	r_2	r_3	$\cdots$	r_m
W_1	R_{11}	R_{12}	R_{13}	$\cdots$	R_{1m}
W_2	R_{21}	R_{22}	R_{23}	$\cdots$	R_{2m}
W_3	R_{31}	R_{32}	R_{33}	$\cdots$	R_{3m}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
W_n	R_{k1}	R_{k2}	R_{k3}	$\cdots$	R_{nm}

relationship between workflow W_i and resource r_j is R_{ij}, where R_{ij} is an integer value depicting the reference number of resource r_j from workflow W_i in the

Table 6.2: Workflow-resource relationship matrix

	r_1	r_2	r_3	r_4	r_5
W_1	1	0	0	0	0
W_2	0	0	1	0	0
W_3	0	0	0	1	0
W_4	0	0	0	0	1

workflow-resource matrix. R_{ij} is set to 0 for unreferenced resources. For example, let four workflows W_1, W_2, W_3, W_4, be processed by five IT resources r_1, r_2, r_3, r_4, r_5. The reference numbers of each workflow are R_1, R_2, R_3 and R_4 respectively and the availability of each resource is $P(r_1), P(r_2), P(r_3), P(r_4)$ and $P(r_5)$ respectively. For the workflow-resource matrix shown in Table 6.2, resource 2 is not included in the resource list of the workflows, and so $(R_{11}, R_{23}, R_{34}, R_{45})$ are set to 1 while the unreferenced resource is set to 0. Table 6.3 shows the components failure and the cost of each component of the resource, specifying the fundamental parameters for resource availability and cost. Failure behavior (MTTR, MTTF) and the cost elements ColdCost, ActiveCost, RepairCost are allocated for each component of the resources. The parameters of component availability are $MTTR$, which specifies the mean time to repair after each failure, and $MTTF$ which specifies the mean time between failures. These parameter specify the availability of each component of the resource. The availability of a single component is calculated as MTTF / (MTTF + MTTR). The cost of a resource is equal to the sum of its components.

Table 6.3: Component failure and costs.

Component	ColdCost	ActivCost	RepairCost	MTTR	MTTF
$Comp_1$	c_{11}	c_{12}	c_{13}	MTTR_1	MTTF_1
$Comp_2$	c_{21}	c_{22}	c_{23}	MTTR_2	MTTF_2
$Comp_3$	c_{31}	c_{32}	c_{33}	MTTR_3	MTTF_3
...	...	...	...	...	...
$Comp_j$	c_{j1}	c_{j2}	c_{j3}	MTTR_j	MTTF_j

The cost parameters for each resource are specified by various costs associated with the components of the resource as follows:

- *ColdCost*: specifies the cost when component is powered off.
- *ActiveCost*: specifies the cost when component is powered on.

- *RepairCost*: specifies the cost to repair the component when the component is down.

Table 6.3 shows the cost of resource components. The cost c_i of resource r_i equals the sum of component cost c_{ij}, where j is the number of components of resource c_i. The current availability of resource P_{r_j} and for workflow $P(w_i)$ is calculated with respect to the availability behaviour of the resource components, as explained below. The current availability of resources for workflow $P(W_i)$ is then compared with the workflow availability requirement $P_{W_{req}}$: and if $P(W_i) \geqslant P_{W_{req}}$ the required availability of resources is fulfil; otherwise the availability is unsatisfied and some resources in the resource list of workflow W_i need to have their availability enhanced. The identification of weak point depends on the behaviour and cost of resource components. The methodology is called weak point analysis, because it determines the relevant resources which do not meet the required level of in a workflows or which do not keep cost as low as possible.

Existing approaches [20] requires the modeller to identify the frequency with which components in the workflow are used (the usage of resources has to be known by the analyst), but this task is not automated and may require considerable computational effort. In this experiment, the proposed framework used to automatically determine the frequency with which resources in the workflow are being used are based on the specifications of the business process. This achieved by applying the support tools to the output of the (*CSPL* file) how often resources are used can be determined by running the availability algorithm [20]. This represents a considerable improvement on previous study [20] where the modeler needs to input resource usage by hand, which requires significant computational effort and leads to inaccuracies.

6.3 The Framework of Weak Point Analysis Methodology

This section presents the framework of the methodology used for weak point analysis which is implemented according to the main framework proposed in this study and presented in Chapter 3. The framework consists of three main modules which are shown in Figure 6.1 and illustrated below.

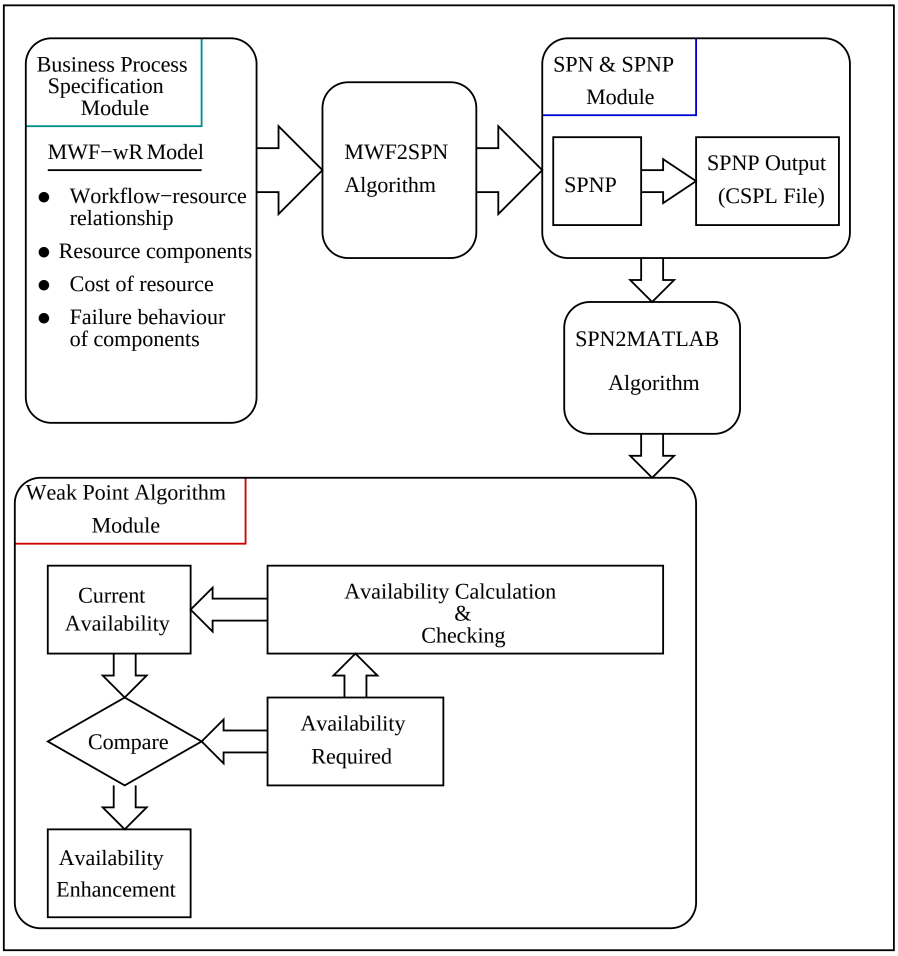


Figure 6.1: Framework for weak point analysis methodology

6.3.1 The *MWF-wR* business process specification module

This section describes the *MWF-wR* business processes specification module of the weak point analysis approach. This is the first module in the proposed framework used in this research and is presented in Section 3.3.1. The components of the *MWF-wR* model define the process and the services used in order to conduct weak point analysis of the availability of resources. It contains the following components:

- The relationship between the workflow and resources in the business process. Each workflow is associated with a subset of resources used for the execution of tasks. The relationship is described as given in Table 6.1 to produce the workflow-resource matrix in Table 6.2 which determines the relationship between the workflows and resources.

- The components of resources. Each resource has a number of components as given in Table 6.3 which determine its behaviour.

- The cost and the failure behaviour of the components. The components of resource cost and failure behaviour are described as given in Table 6.3 which determine the availability and cost of each resource.

Table 6.3 depicts the cost and the failure behavior of the components of resources which are used as the fundamental parameters for calculating the availability and cost of resources. Determination of the above components is the key factor for build the structure of the *MWF-wR* business process file and the basic step of this module. The structure of the *MWF-wR* specification file used for the weak point analysis is depicted in Table 6.4., which represent the *MWF-wR* business process components that described in Section 4.4. These components will be mapped into the *CSPL* file of the *SPNP* according to the *MWF2SPN* algorithm which is presented in Section 4.5.1. The next section describes the *SPN* module including the mapping relation.

Table 6.4: MWF-wR file for weak point analysis

DSR = structure
DSR = structure(field,value)
field1 = 'workflow'; value1 = 'string';
filed2 = 'resource'; value2 = 'string';
field3 = 'availability'; value3 = 'double';
field4 = 'cost'; value4 = 'double';
DSR(i).MWF-wR = ($'workflow'_i$, 'a', DSR(i).resource'($'resource'_j$, 'b', $'cost'_j$,'0.00', $'availability'_j$,'0.00')

6.3.2 Stochastic Petri nets module

This module represents the second module of the framework proposed in this research which is presented in Section 3.3.2. The key contribution of this module is the mapping of the *MWF-wR* business process model into the *SPN* model. The components of the weak point analysis specified in Section 6.3.1 are mapped into a *SPN* model using the *MWF2SPN* algorithm which is defined in Section 4.5.1. The components are mapped to a *CSPL* file as depicted in Section 5.5.2. and according to the following *CSPL* file formula.

- ***CSPL* file of *MWF-wR* for availability weak point analysis**:

#include <stdio.h>
#include <math.h>
#include <user.h>
/* global variables */
/* Prototype for the function(s) */
double C_j (); (j = 1 : n , n = number of resources)
/* ========== OPTIONS ========== */
void options() {
}
/* ========== DEFINITION OF THE NET ========== */
void net() {
/* ===== PARAM VARIABLES ==== */
/* ========== PLACE ========== */
Place("p_i"); (i = 1 : (m + n), m = number of workflows and n = number of resources)
init("p_i", M_i);
/* ========== TRANSITION ========== */
/* Immediate Transitions */
/* Timed Transitions */
rateval("t_i",Av_{ij},"p_j"); (i = 1:m, m = number of workflows and j = 1:k, k = number of resources associated with workflow represented by t_i)
/* ====== ARC ====== */
/* Input Arcs */
iarc("t_i","$p_{i(j+1)}$"); (i = 1 : m , m = number of workflows & j = 1 :k, k = number of resources associated with workflow represented by t_i)
/* output Arcs */
oarc("t_i","$p_{i(j+1)}$"); (i = 1 : m , m = number of workflows & j = 1 :k, k = number of resources associated with workflow represented by t_i)
}

```
/* C Functions */
double C_j () {
return(mark("p_j") * c_j)          (j = 1 : n , n = number resources)
}
```

This *CSPL* file formula determines the mapping of fundamental components of the *MWF-wR* business process to the parameters used by the algorithm of the weak point analysis which are described as follows.

- A workflow of business process is specified by the function *Place ()* and the function *rateval()* of CSPL file, where each workflow is mapped to one place and one transition.
- A resource of business process is specified by the function *Place ()* of CSPL file, where each resource is mapped to one place.
- The relationship between the workflow and the relevant resources of business process is specified by the functions *iarc()* and *oarc()* of CSPL file, where each resource used to complete the workflow is connected to it by one input arc and one output arc.
- The current availability of each resource of business process is specified by the function *rateval()* of CSPL file, where the availability is mapped to the rate of the transition associated to the corresponding resource place in this function.
- The cost of each resource of business process is identified by a function defined in the *C Function* of CSPL file, where the cost of resource is mapped to the variable of the defined function which returned the marking of the place of resource.

Having mapped the *MWF-wR* business process components to the *CSPL* file, the next module is used to conduct the weak point analysis of the availability of resources using the information contained in this file.

6.3.3 Availability weak point analysis algorithm module

This module represents the algorithm module, the third module of the framework which is presented in Section 3.3.2. The module carries out the analysis of weak point in resources availability by running the algorithm shown in table 6.5. The algorithm aims at perform optimal solution for enhance the availability of resources while keep overall cost as minimum as possible. The analysis is carried out on the

Table 6.5: High availability weak point analysis algorithm

```
MinCost=MaxNumber
OptimalSolution=Null
for every Resource r_i in ResourceList do
    if NumofCandidateResource r_i > 1 then
       for every CandidataResource rR_j in Resour r_i
       do
      rR_j =GetCandidateResource (r_i)
        ResourceList=GenerateResourceList rR_j, r_i
        UtilityFunction=GenerateUtilityFunction rR_j, r_i
        AddtoResourceListPool(ResourceList)
        AddtoUtilityFunctionPool(UtilityFunction)
    end if
end for
for every ResourceList and counterpart UtilityFunction in
ResourceListPool and UtilityFunctionPool do
Cost,Solution = WeakPointAnalysis(ResourceList, UtilityFunction,
Topology, WorkflowList)
    if Cost<MinCost then
        OptimalSolution=Solution
    end if
end for
Output MinCost,OptimalSolution
```

CSPL file of the *SPN* model constructed in previous module. According to the workflow-resources matrix extracted from the SPNP file, the current availability of each resource in the workflow is calculated. The single component availability *Comp(Av)* is calculated according to the following equation:

$$Comp(Av) = MTTF/(MTTF + MTTR) \tag{6.1}$$

where $MTTF$ specifies the mean time to failure of the component and $MTTR$ specifies the mean time to repair after each failure. So, resource availability lies in the range from 0 and 1. The cost c_i of the resource r_i is equal to the sum of the cost of each component c_{ij} where j is the number of components of resource c_i. Table 6.3 shows the cost of the components of resource, where the cost of resource r_i calculated as:

$$c_i = \sum_{j=1}^{|j|} c_{ij} \tag{6.2}$$

where $|j|$ is the number of resource components.

The current availability of resources for each workflow is calculated as state in the following equation:

$$P(W_i) = \prod_{j=1}^{m} (P(r_j)^{R_{ij}}), \tag{6.3}$$

where, $P(W_i)$ is the current availability for the workflow W_i, and $P(r_j)$ is the availability of the resource r_j. The current availability capability for workflow $P(W_i)$ is then compared with the workflow availability requirement $P(w_{req}$: and if $P(W_i) \geqslant P(w_{req})$, the requirement is met; otherwise, the availability requirement is unsatisfied and some resources in resource list of workflow W_i need to have their availability enhanced.

The practical example used the experimental scenario presented in previous study [20]. The example consists of four resources r_1, r_2, r_3, r_4 and two specified business process workflows W_1 and W_2 over the application and resources. Table 6.6 shown the workflow and resources component failure and costs are showing in Table 6.7

Table 6.6: Resource to workflow mapping

BP Workflow	Resource
workflow1	r_1
	r_2
	r_3
	r_4
workflow2	r_1
	r_2
	r_4

where the parameters are the failure behavior (MTTR, MTTF) and the cost components (ColdCost, ActiveCost, RepairCost). The availability parameter MTTR specifies the mean time to repair after each failure, and the parameter MTTF specifies the mean time between failures. The availability of resources depends on the availability of their components,which are depicted in Table 6.8. The availability of each is calculated according to equation (6.1) and are shown in Table 6.9. The availability of the resources calculated using equation (6.3) and shown in Table 6.10.

Table 6.7: Component failure and costs.

Component	ColdCost	ActivCost	RepairCost	MTTR	MTTF
x86Server	£2400	£2640	£300	3600_{sec}	75_{days}
POWERServer	£85000	£93500	£1500	3600_{sec}	150_{days}
LinuxOS	£0	£0	£0	120_{sec}	45_{days}
WindowsOS	£0	£200	£0	120_{sec}	30_{days}
AIXOS	£0	£400	£0	240_{sec}	100_{days}
WASServer	£0	£100	£0	100_{sec}	30_{days}
IHSServer	£0	£45	£0	50_{sec}	25_{days}
DB2Server	£0	£60	£0	80_{sec}	30_{days}

Table 6.8: Resource components

Resource	Servier	OS	Hardware
r_1	IHSServer	LinuxOS	x86Server
r_2	WASServer	LinuxOS	x86Server
r_3	WASServer	AIXOS	POWERServer
r_4	DB2Server	WindowsOS	x86Server

Table 6.9: Components of resources availability

Component	Availability
x86Server	%99.944
POWERServer	%99.972
LinuxOS	%99.996
WindowsOS	%99.995
AIXOS	%99.997
WASServer	%99.996
IHSServer	%99.997
DB2Server	%99.996

Table 6.10: Resources availability

Resource	Availability
r_1	%99.937
r_2	%99.936
r_3	%99.965
r_4	%99.935

The cost parameters *ColdCost*, *ActiveCost*' *RepairCost* of each resource specify the various costs associated with the components shown in Table 6.7. The cost of each resource is the sum of the costs of components and is calculated using equation (6.2). The cost of resources are obtained as shown in Table 6.11. The current availability of workflow resources is calculated using equation (6.2) and shown in Table

Table 6.11: Resources cost

Resource	cost (c_i)
r_1	£5385
r_2	£5440
r_3	£180500
r_4	£5600

6.12. The components of the business process for weak point and the relationships

Table 6.12: Workflow availability

Business process	Availability
Workflow1	%99.773
Workflow2	%99.808

between them are specified and they are then mapped to the *SPN* model for automated analysis. The *SPN* model consists of six places and two timed transitions, which represent the two workflows W_1 and W_2. The four places denote the four resources r_1, r_2, r_3, r_4 connected to the above two timed transitions by input and output arcs according to the workflow-resource relationships presented in Table 6.6. The following is the *CSPL* file of the *SPN* model in this example:

- ***CSPL* file of example for weak point analysis**

```
#include <stdio.h>
#include <math.h>
#include <user.h>
/* global variables */
/* Prototype for the function(s) */
double C_r1 ();
double C_r2 ();
double C_r3 ();
double C_r4 ();
/* ========== OPTIONS ========== */
void options() {
}
/* ========== DEFINITION OF THE NET ========== */
void net() {
```

```
/* ===== PARAM VARIABLES ==== */
/* ========== PLACE ========== */
Place("ts1");
init("ts1",4);
Place("ts2");
init("ts2",3);
Place("r1");
init("r1",1);
Place("r2");
init("r2",1);
Place("r3");
init("r3",1);
Place("r4");
init("r4",1);
/* ========== TRANSITION ========== */
/* Immediate Transitions */
/* Timed Transitions */
rateval("t1",0.99937,"r1");
rateval("t1",0.99936,"r2");
rateval("t1",0.99965,"r3");
rateval("t1",0.99935,"r4");
rateval("t2",0.99937,"r1");
rateval("t2",0.99936,"r2");
rateval("t2",0.99935,"r4");
/* ====== ARC ====== */
/* Input Arcs */
iarc("t1","ts1");
iarc("t2","ts2");
iarc("t1","r1");
iarc("t1","r2");
iarc("t1","r3");
iarc("t1","r4");
iarc("t2","r1");
iarc("t2","r2");
```

iarc("t_2","r_4");
/* output Arcs */
oarc("t_1","ts_2");
oarc("t_1","r_1");
oarc("t_1","r_2");
oarc("t_1","r_3");
oarc("t_1","r_4");
oarc("t_2","r_1");
oarc("t_2","r_2");
oarc("t_2","r_4");
}
/* C Functions */
double C_{r_1} () {
return(mark("1") $*$ 5385)
}
double C_{r_2} () {
return(mark("1") $*$ 5440)
}
double C_{r_3} () {
return(mark("1") $*$ 180500)
}
double C_{r_4} () {
return(mark("1") $*$ 5600)
}

The elements of the *CSPL* file are described as follows:

- The functions *(Place"ts_1")* and *iarc*("t_1", ts_1) of CSPL stand for the workflow W_1 of business process where the place ts_1 is connected to the transition t_1 by only the input arc *iarc*("t_1", ts_1).

- The function *init("ts_1",4)* of CSPL specifies the number of resources in the workflow W_1 resource list of business process, which is in this case is 4 resources.

- The function *rateval("t_1",0.99937,"r_1")* of CSPL stands for the availability of the resource r_1 of business process which is in the resource list of W_1.

- The function *rateval("t_1",0.99936,"r_2")* of CSPL specifies the availability of the resource r_2 of business process which is in the resource list of W_1.
- The function *rateval("t_1",0.99965,"r_3")* of CSPL specifies the availability of the resource r_3 of business process which is in the resource list of W_1.
- The function *rateval("t_1",0.99935,"r_4")* of CSPL specifies the availability of the resource r_4 of business process which is in the resource list of W_1.
- The functions *(Place"ts_2")* and $iarc("t_2", "ts_2")$ of CSPL stand for the workflow W_2 of business process where the place t_2 is connected to the transition t_2 by only the input arc $iarc("t_2", ts_2)$.
- The function *init("ts_2",3)* of CSPL specifies the number of resources in the workflow W_2 resource list of business process, which in this case is 3 resources.
- The function *rateval("t_2",0.99937,"r_1")* of CSPL stands for the availability of the resource r_1 in which is in the resource list of W_2 of business process.
- The function *rateval("t_2",0.99936,"r_2")* of CSPL specifies the availability of the resource r_2 which is in the resource list of W_2 of business process.
- The function *rateval("t_2",0.99935,"r_4")* of CSPL specifies the availability of the resource r_4 which is in the resource list of W_2.
- The function *(Place"r_1")* of CSPL stands for the resource r_1. This resource is in the resource lists of both W_1 and W_2 of business process, and so it is connected to W_1 by one input arc of function $iarc("t_1", "r_1")$ and one output arc of function $oarc("t_1", "r_1")$, and connected to W_2 by one input arc of function $iarc("t_2", "r_1")$ and one output arc of function $oarc("t_2", "r_1")$.
- The function *(Place"r_2")* of CSPL stands for the resource r_2. The resource is in the resource lists of both W_1 and W_2 of business process, and so it is connected to W_1 by one input arc of function $iarc("t_1", "r_2")$ and one output arc of function $oarc("t_1", "r_2")$, and connected to W_2 by one input arc of function $iarc("t_2", "r_2")$ and one output arc of function $oarc("t_2", "r_2")$.

- The function *(Place"r_3")* of CSPL stands for the resource r_3, which is in the resource list of W_1 of business process, and so is connected to W_1 by one input arc of function $iarc("t_1","r_3")$ and one output arc of function $oarc("t_1","r_3")$.
- The function *(Place"r_4")* of CSPL stands for the resource r_4. The resource is in the resource lists of both W_1 and W_2 of business process, and so it is connected to W_1 by one input arc of the function $iarc("t_1","r_4")$ and one output arc of the function $oarc("t_1","r_4")$, as well connected to W_2 by one input arc of the function $iarc("t_2","r_4")$ and one output arc of the function $oarc("t_2","r_4")$.
- The function *double* C_{r_1} of CSPL specifies the cost of resource r_1 of business process which is return the (marking of place $r_1 = 1$) multiplied by (c_1 the cost of $r_1 = 5385$).
- The function *double* C_{r_2} of CSPL specifies the cost of resource r_2 of business process which is return the (marking of place $r_2 = 1$) multiplied by (c_2 the cost of $r_2 = 5440$).
- The function *double* C_{r_3} of CSPL specifies the cost of resource r_3 of business process which is return the (marking of place $r_3 = 1$) multiplied by (c_3 the cost of $r_3 = 180500$).
- The function *double* C_{r_4} of CSPL specifies the cost of resource r_4 of business process which is return the (marking of place $r_4 = 1$) multiplied by (c_4 the cost of $r_4 = 5600$).

The *CSPL* file contained the information in which it used to provide the correct input parameters to the Matlab to conduct the weak point analysis of the availability of resources. The *CSPL* file is mapped into Matlab using the *SPN2MATLAB* general algorithm presented in Section 5.5.3 which is designed for mapping the *SPN* model to Matlab. The *SPN2MATLAB* algorithm which is used for the enhancement of the availability of resources approach is described as follows:

- **The generic *SPN2MATLAB* algorithm for mapping *SPN* into Matlab for enhancement resource availability approaches)**

Step 1 : Read file (CSPL file of SPN)

Step 2 : While not EOF () do:

Step 3 : Read word Place("x")

Step 4 : Extract value of x

Step 5 : Put x in list of Places

Step 6 : Read word init("x",y)

Step 7 : Extract values of x and y

Step 8 : Put x in list of init1

Step 9 : Put y in list of init2

Step 10 : Read word rateval("x",y,"z")

Step 11 : Extract values of x, y and z

Step 12 : Put x in list of rateval1

Step 13 : Put y in list of rateval2

Step 14 : Put z in list of rateval3

Step 15 : Read word iarc("x","y")

Step 16 : Extract value of x and y

Step 17 : Put x in list of iarc1

Step 18 : Put y in list of iarc2

Step 19 : Read word oarc("x","y")

Step 20 : Extract value of x and y

Step 21 : Put x in list of oarc1

Step 22 : Put y in list of oarc2

Step 23 : Read word return(mark("x"*y)

Step 24 : Extract values of x and y

Step 25 : Put x in list of return1

Step 26 : Put y in list of return2

Step 27 : If list Places = iarc2

Step 28 : Put iarc1 in list of iarc3

Step 29 : Put iarc2 in list of iarc4

Step 30 : If list iarc3 = oarc2 and iarc4 = oarc1

Step 31 : Put oarc1 in list of resouces1

Step 32 : Put oarc2 in list of resouces2

Step 33 : Else

Step 34 : Put iarc3 in list of Task1

Step 35 : Put oarc2 in list of Task2

Step 36 : If list resouces2 = init1

Step 37 : Put init1 in list of resouces3

Step 38 : Put init2 in list of resouces4

Step 39 : If list resouces4 = return1

Step 40 : Put return2 in list of resouces5

Step 41 : for i = 1: number of resources

Step 42 : DSR(i).Resource = 'string'

Step 43 : Put resource2 in list of Resource

Step 44 : for i = 1: number of resources

Step 45 : DSR(i).Cost = '0.00'

Step 46 : Put resouces5*resouces4 in list of Cost

Step 47 : for i = 1: number of Tasks

Step 48 : DSR(i).Task = 'string'

Step 49 : Put Task1 in list of Task

Step 50 : If list Resource = rateval3

Step 51 : Put rateval2 in list of Availability

Step 52 : for i = 1: number of resources

Step 53 : DSR(i).Availability = '0.00'

Step 54 : Put (Task, Resource, Availability, Cost) in list of AVAILABILITY

Step 55 : for i = 1: number of Tasks

Step 56 : DSR(i).AVAILABILITY = struct('Task','string','Resource','string', 'Availability','0.00','Cost','0.00')

The mapping algorithm *SPN2MATLAB* creates an output file. The results shown in Table 6.13 are exactly the same values of the parameters that have been obtained from the example used in weak point analysis algorithm which is used in this work. The optimal solution for weak point analysis is found using the algorithm by identifying the resources in the workflow lists for which availability needs to be enhanced. More information about weak point analysis can be found in the original study [20].

6.4 Business process workflow scheduling

The second experiment used to demonstrate the framework's validity by applying it in the scheduling of workflow as quantitative evaluation approach to optimise the business process workflows. Business process workflow scheduling is an optimisation

Table 6.13: The file of the example of high availability

W_1	r_1	0.99937	5385
W_1	r_2	0.99936	5440
W_1	r_3	0.99965	180500
W_1	r_4	0.99935	5600
W_2	r_1	0.99937	5385
W_2	r_2	0.99936	5440
W_2	r_4	0.99935	5600

problem concerned with the optimal allocation of resources to tasks over time. A previously published study [21] has been chosen as an example of this approach. A stochastic scheduling algorithm is used to optimise overall workflow execution time and to minimise overall workflow costs by allocating the available resources to tasks.

6.4.1 Grid workflow scheduling in WOSE Scenario

This section describe a methodology in business process performance evaluation used to scheduling workflows proposed in a previous study [21] as example. The method is called grid workflow scheduling in WOSE which consists of two stages. In the first stage a set of equations may be solved using integer linear programming (ILP). The second stage uses stochastic ILP to solve the proposed stochastic scheduling algorithm. The used of the algorithm is intended to optimise overall workflow execution time and to minimize overall workflow cost while ensuring the workflow system satisfies QoS requirements. So, resources are allocated to tasks such that their execution is completed within given maximum deadlines as well within an overall workflow completion time. Existing approaches [21] required the modeller to specify the usage of resources and their elements by hand, which requires considerable computational effort and is prone to error. The analytic framework used here improves this situation by automatically deriving the usage resources from business process specification.

The main approach for conducting the workflow scheduling used to map the business process model to an (*CSPL* file) of *SPNP* tool as given in previous tools support chapter. Applying the support tool to the output of the *CSPL* file) the used resources and their elements can introduces as the parameters of the scheduling algorithm.

The parameters are fed into Matlab programs to conduct the automated workflow scheduling. In the work presented in [21], the modeler needs to input resource usage by hand, which requires significant computational effort and leads to inaccuracies. The present study make considerable improvement on previous study [21] by proposed a support tools that used to introduce the parameters of the workflow scheduling algorithm in automated way.

6.5 The framework for Business Process Workflow Scheduling

This section presents the methodology used for workflow scheduling which is implemented according to the main framework proposed in this study and presented in Chapter 3. The framework is shown in Figure 6.2 and consists of three main modules illustrated below.

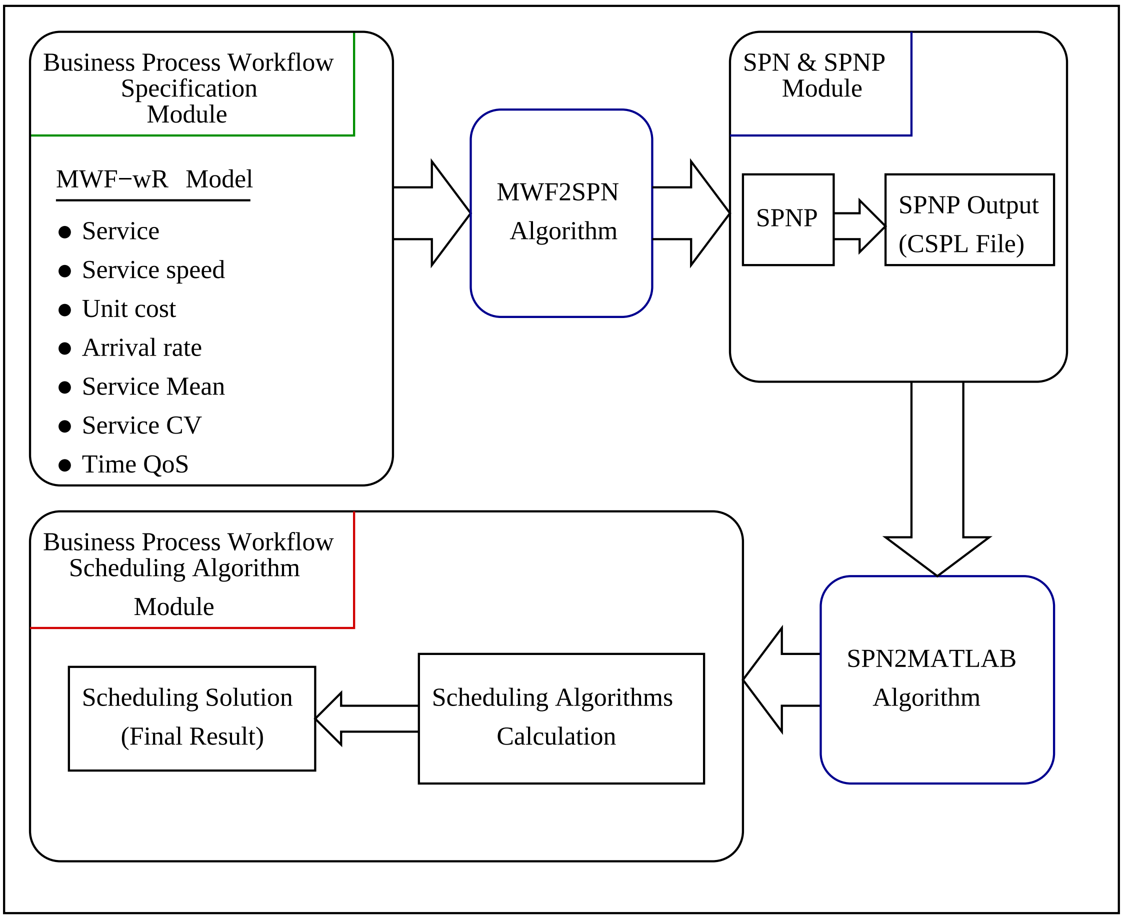


Figure 6.2: Framework for business process workflow scheduling

6.5.1 The *MWF-wR* business process specification for workflow scheduling module

This section describes the *MWF-wR* business process specification module which represent the information of the workflow scheduling model. This represents the first module of the proposed framework used in this research and presented in Section 3.3.1. The components of the *MWF-wR* and their relationships are defined. The components signify the parameters used to conduct workflow scheduling. These parameters are shown in Table 6.14 and described as follows:

Table 6.14: The parameters used by the workflow scheduling algorithm

Scheduling Parameters	
A_i	Services
a_{ir}	Expected time
c_{ir}	Expected cost
x_{ir}	Selection variable
$time_{QoS}$	Maximum time of workflow execution
$deadline_i$	Expected time to complete A_i
|A|	Number of service
$\|a_i\|$	Number of Web Services matching A_i

- A finite number of services A_i. It represents a finite set of task $\{ts_i\}$ of workflow model.
- a_{ir} is the expected time associated with the r^{th} web service match the service A_i. It denotes the expected time s_{ij} associated with the resource r_j used to complete the task ts_i or in other words the deadline assigned to each task. The resource $r_j \in p_i(R)$, where $p_i(R)$ is subset of resources may used to complete the task ts_i.
- c_{ir} is the expected cost associated with the r^{th} web service match the service A_i. It denotes the expected cost associated with the resource r_j used to complete the task ts_i.

- x_{ir} the selection variable associated with the r^{th} web service match the service A_i. It denotes the resource r_j from the subset $p_i(R)$ used to complete the task ts_i.

- $time_{QoS}$ denotes the maximum time within which all the tasks of the workflow should be executed.

- $deadline_i$ is the time within which the service A_i is expected to be complete. It denote in workflow, the time within which the task ts_i is expected to be complete.

- $|A|$ is the number of abstract services. It represents finite number m of the tasks ts_i that the workflow model consists of.

- $|a_i|$ is a finite number of Web services match the service A_i. It represents the number of resources in the subset $p_i(R)$ used to complete the task ts_i.

The scheduling parameters are mapped into the *SPN* model using the *MWF2SPN* algorithm which is presented in Section 4.5.1. The *CSPL* file of an *SPNP* which is described in Section 5.5.2 used to represent the *SPN* model. The next module is used to conduct the mapping relations from *MWF-wR* model to *SPN* model.

6.5.2 Stochastic Petri Nets Module

This module represents the second module of the main framework presented in Section 3.3.2. An *SPN* model is generated by mapping the *MWF-wR* business process workflow scheduling model into *SPN*. The components of the workflow scheduling are mapped into the *CSPL* file using the format presented in Section 5.5.2. and according to the following:

- **Generic *CSPL* file for mapping workflow scheduling components**:

```
#include <stdio.h>
#include <math.h>
#include <user.h>
/* global variables */
/* Prototype for the function(s) */
```

double C_j (); (j = 1 : n , n = number of resources)

/* ========== OPTIONS ========== */

void options() {

}

/* ========== DEFINITION OF THE NET ========== */

void net() {

/* ===== PARAM VARIABLES ==== */

/* ========== PLACE ========== */

Place("p_i"); (i = 1 : (m + n), m = number of tasks and n = number of resources)

init("p_i", M_i);

/* ========== TRANSITION ========== */

/* Immediate Transitions */

/* Timed Transitions */

rateval("t_i",s_{ij},"p_j"); (i = 1:m, m = number of tasks & j = 1:k, k = number of resources associated with task represented by t_i)

/* ====== ARC ====== */

/* Input Arcs */

iarc("t_i", "$p_{j(k+1)}$"); (j = 1 : m , m = number of tasks & j = 1 :k, k = number of resources associated with task represented by t_i)

/* output Arcs */

oarc("t_i", "$p_{j(k+1)}$"); (j = 1 : m , m = number of tasks & j = 1 :k, k = number of resources associated with task represented by t_i)

}

/* C Functions */

double C_j () {

return(mark("p_j") $*$ c_j) (j = 1 : n , n = number of resources)

}

The formula determines mapping the workflow components to the *CSPL* file. The file is depicted in what follow:

- The set of tasks TS of workflow are specified by the functions *Place ()* and *rateval()* of CSPL file, where each task ts_i is mapped into one place p_i and one transition t_i.

- The set of resources R of workflow are specified by the function *Place ()* of CSPL file, where each resource r_i is mapped into one place p_i.
- The relationship between the tasks and relevant resources of workflow are specified by the functions *iarc()* and *oarc()* of CSPL file, where each resource r_j used to complete the task ts_i is connected to the task by one input arc and one output arc.
- The processing time s_{ij} of each task of workflow is specified by the function *rateval()* of CSPL file, where each processing time is mapped to the rate of the transition that represents the task. The transition is connected to corresponding place of the resource used to complete this task.
- The cost c_j of each resource r_j of workflow is identified by a function in the *C Function* of CSPL file. Each resource cost is mapped to a variable of the function which is return the marking of place of the resource.

Having mapped the component of workflow to the *CSPL* file, the next module conducts the workflow scheduling.

6.5.3 Business process workflow scheduling module

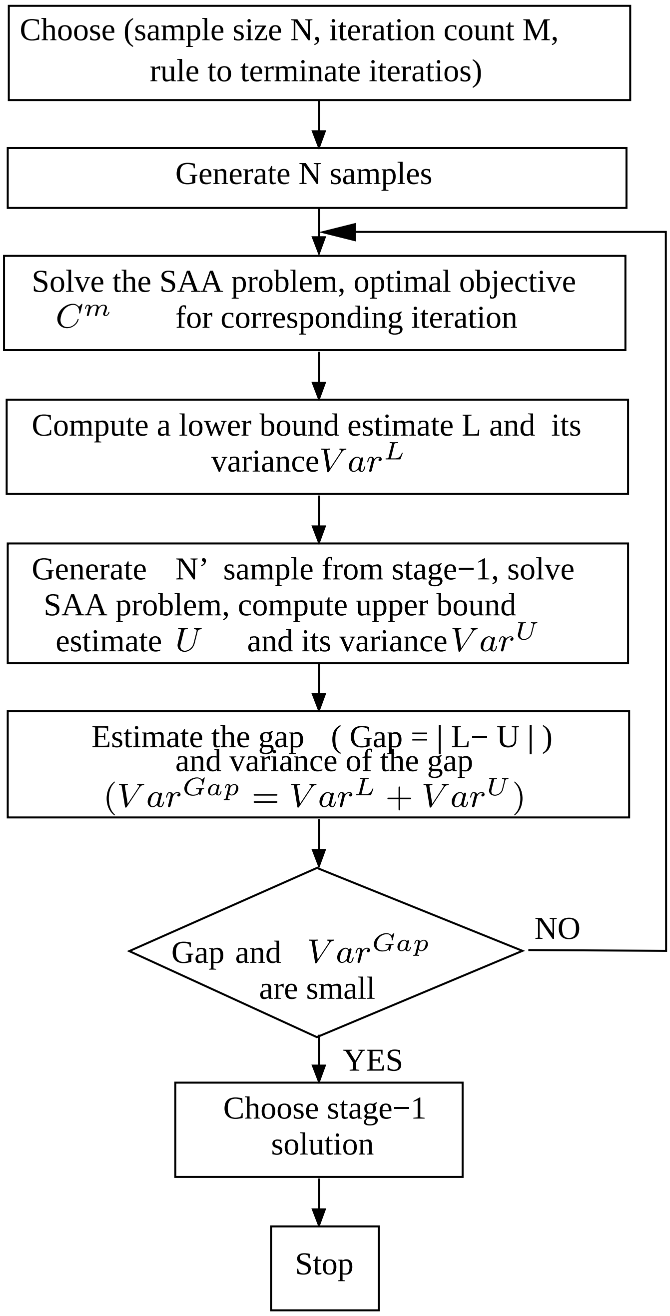
This module represents the third module of the framework, the algorithm module, which is presented in Section 3.3.2. This module carries out the workflow scheduling by running the algorithm shown in table 6.14. The algorithm aims to optimise workflow and perform solution to allocate resources of workflow in away that keep overall cost as minimum as possible and satisfy QoS requirements. The example used in present study conducting the workflow scheduling in two stages. The first stage uses deterministic ILP to minimise the overall workflow costs, and the second stage uses stochastic ILP to minimise future costs. The two stages described as follow:

- Stage-1

$$Cost = minimise\,[C] \tag{6.4}$$

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{6.5}$$

Table 6.15: Scheduling workflow algorithm



Here, C is the cost associated with web services, and the following constraints are applied:

$$\forall i, \sum_{j=1}^{n} x_{ij} = 1 \tag{6.6}$$

$$x_{ij} \in \{0, 1\} \tag{6.7}$$

$$\sum_{j=1}^{n} a_{ij} x_{ij} \leqslant ddl_i \tag{6.8}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_{ij} \leqslant ddl \tag{6.9}$$

where m is the number of services, and n is the number of web services matching i^{th} services, a_{ij} is the expected time variable associated with the j^{th} web service matching the i^{th} service, c_{ij} denotes the expected cost variable associated with the j^{th} web service matching the i^{th} service, x_{ij} is the selection variable associated with the j^{th} web service matching the i^{th} service, ddl is the maximum time in which the workflow should be executed and ddl_i is the time in which the i^{th} service is expected to be complete.

• Stage-2

Stage two aims to minimise future costs.

$$Cost = minimise\,[C + E(Q(x_s, w))] \tag{6.10}$$

where, w is the vector of random variables of runtimes and costs of services. The x_s vector denotes the solutions to stage1. The function E is the expected objected value of stage2. The approach proposed previously [21] is improved by using a framework to automated the scheduling procedure. The analyst needs to identify by hand how often a business process uses certain resources and the elements of these resources. The approach is improved by deriving the usage resources for complete the task from business process specification automatically. Given the framework of workflow scheduling, the components of the business process are mapped to the *CSPL* file of the *SPNP*. The support tools determine how often resources are used based on the specification of the business process workflow . However, in the previous study [21], the resources, tasks and the relationships between them has to be known by the modeller. This study presents an implementation tool to determine usage resource and introduce the parameters of the scheduling algorithm. The parameters are fed into the Matlab programs to conduct the automated workflow scheduling.

The parameters of workflow scheduling in the example are presented in Table 6.16. The Table summarises the functional parameters of the model, which are mapped into a *CSPL* file. The following is the *CSPL* file for workflow scheduling:

Table 6.16: The parameters of the example for workflow scheduling

Workflow Parameters
Services match A_i Service speed (kMIPS) Unit cost (per sec) Arrival Rate λ (per sec) A_i Mean (μ) (k MI) A_i CV = σ/μ $time_{QoS}$ (sec)

- ***CSPL* file for business process workflow scheduling**:

```
#include <stdio.h>
#include <math.h>
#include <user.h>
/* global variables */
/* Prototype for the function(s) */
double Arrival Rate ();
double Unit cost_j ();
double Service speed_j ();
/* ========== OPTIONS ========== */
void options() {
}
/* ========== DEFINITION OF THE NET ========== */
void net() {
/* ===== PARAM VARIABLES ==== */
/* ========== PLACE ========== */
Place("task_i");
init("task_i", 1);
Place("resource_j");
init("resource_j", 1);
/* ========== TRANSITION ========== */
/* Immediate Transitions */
/* Timed Transitions */
```

rateval("$task_i$",$Mean(\mu_i)$);

/* ====== ARC ====== */

/* Input Arcs */

iarc("$task_i$","p_i");

iarc("$task_i$","$resource_{ij}$");

/* output Arcs */

oarc("$task_i$",'p_{i+1});

oarc("$task_i$","$resource_{ij}$");

}

/* C Functions */

double *Arr-Rate* () {

return(mark("ts_1") * *Arrival Rate*)

}

double $Cost_j$ () {

return(mark("$resource_j$") * $Unit\ cost_j$)

}

double $Speed_j$ () {

return(mark("$resource_j$") * $Service\ speed_j$)

}

The practical example presented in previous study [21] used a lot of information which made it complex and it is not appropriate to use this range of parameters to illustrate the efficiency of the framework. Therefore, only a sample of parameters is used in the example. The selected parameters are shown in Table 6.17. The

Table 6.17: The selected parameters for workflow scheduling example

Scheduling Parameters		
Services matching ts_1	r_1, r_3, r_4	
Services matching ts_2	r_2, r_4	
	Service speed (kMIPS)	Unit cost (per sec)
r_1	10	9
r_2	14	15
r_3	5	22
r_4	12	7
Arrival Rate (λ) (per sec)		3.5
ts_1 Mean (μ) (kMI)		7.5
ts_2 Mean (μ) (kMI)		16.5

parameters mapped into *SPN* model using the *SPNP* as a tool to represent the model. The following *CSPL* file is generated as a result of the mapping relations:

- ***CSPL* file of example for workflow scheduling**:

```
#include <stdio.h>
#include <math.h>
#include <user.h>
/* global variables */
/* Prototype for the function(s) */
double $Arr - Rate$ ();
double $Cost_1$ ();
double $Cost_2$ ();
double $Cost_3$ ();
double $Cost_4$ ();
double $Speed_1$ ();
double $Speed_2$ ();
double $Speed_3$ ();
double $Speed_4$ ();
/* ========== OPTIONS ========== */
void options() {
}
/* ========== DEFINITION OF THE NET ========== */
void net() {
/* ===== PARAM VARIABLES ==== */
/* ========== PLACE ========== */
Place("$ts_1$");
init("$ts_1$",1);
Place("$ts_2$");
init("$ts_2$",1);
Place("$r_1$");
init("$r_1$", 1);
Place("$r_2$");
init("$r_2$", 1);
Place("$r_3$");
init("$r_3$", 1);
```

Place("r_4");

init("r_4", 1);

/* ========== TRANSITION ========== */

/* Immediate Transitions */

/* Timed Transitions */

rateval("t_1",7.5);

rateval("t_2",16.5);

/* ====== ARC ====== */

/* Input Arcs */

iarc("t_1","ts_1");

iarc("t_2","ts_2");

iarc("t_1","r_1");

iarc("t_1","r_3");

iarc("t_1","r_4");

iarc("t_2","r_2");

iarc("t_2","r_4");

/* output Arcs */

oarc("t_1",'ts_2);

oarc("t_1","r_1");

oarc("t_1","r_3");

oarc("t_1","r_4");

oarc("t_2","r_2");

oarc("t_2","r_4");

}

/* C Functions */

double $Arr - Rate$ () {

return(mark("t_1") * 3.5)

}

double $Cost_1$ () {

return(mark("r_1") * 9)

}

double $Cost_2$ () {

return(mark("r_2") * 15)

}

double $Cost_3$ () {

return(mark("r_3") * 22)

}

double $Cost_4$ () {

return(mark("r_4") * 7)

}

double $Speed_1$ () {

return(mark("r_1") * 10

}

double $Speed_2$ () {

return(mark("r_2") * 14)

}

double $Speed_3$ () {

return(mark("r_3") * 5)

}

double $Speed_4$ () {

return(mark("r_4") * 12)

}

The example consists of two tasks ts_1 and ts_2 and four resources r_1, r_2, r_3 and r_4. The three resources r_1, r_3, r_4 used to complete the task ts_1, and the two resources r_2, r_4 used to complete the task ts_2. Description of the *CSPL* demonstrated in what follows:

- The functions *(Place"ts_1")* and *iarc*("t_1", ts_1) of CSPL file stand for the task ts_1 of workflow where the place ts_1 is connected to the transition t_1 by only the input arc *iarc*("t_1", ts_1).

- The function *init("ts_1",1)* of CSPL file specified the marking of the place ts_1 which is the place of the first task of workflow and is used to define the *Arrival Rate* of the tasks of the workflow.

- The function *rateval("t_1",7.5)* of CSPL file stands for the *Mean (μ_1)* of the task ts_1 of workflow.

- The function *rateval("t_2",16.5)* of CSPL file represents the *Mean (μ_2)* of the task ts_2 of workflow.

- The functions *(Place"ts_2")* and *iarc*("t_2","ts_2") of CSPL file stand for the task ts_2 of workflow where the place ts_2 is connected to the transition t_2 by only the input arc *iarc*("t_2", ts_2).

- The function *(Place"r_1")* of CSPL file stands for the resource r_1 of workflow. The resource used to complete the task ts_1, so it connected to ts_1 by one input arc of the function *iarc*("t_1","r_1") and one output arc of the function *oarc*("t_1","r_1").

- The function *(Place"r_2")* of CSPL file stands for the resource r_2 of workflow. The resource used to complete the tasks ts_1 and ts_2, so it is connected to ts_1 by one input arc of the function *iarc*("t_1","r_2") and one output arc of the function *oarc*("t_1","r_2"), and it is connected to ts_2 by one input arc of the function *iarc*("t_2","r_2") and one output arc of the function *oarc*("t_2","r_2").

- The function *(Place"r_3")* of CSPL file stands for the resource r_3 of workflow. The resource used to complete the task ts_1, so it is connected to ts_1 by one input arc of the function *iarc*("t_1","r_3") and one output arc of the function *oarc*("t_1","r_3").

- The function *(Place"r_4")* of CSPL file stands for the resource r_4 of workflow. The resource used to complete the tasks ts_1 and ts_2, so it is connected to ts_1 by one input arc of the function *iarc*("t_1","r_4") and one output arc of the function *oarc*("t_1","r_4"), and it is connected to ts_2 by one input arc of the function *iarc*("t_2","r_4") and one output arc of the function *oarc*("t_2","r_4").

- In the *C Functions* of CSPL file:

  - The function *double Arr-Rate ()* specified the *Arrival Rate* of the tasks to the workflow system. It returns the marking of the place $ts_1 = 1$, which is the place representing the first task, multiplied by 7.5 , which is the *Arrival Rate* of tasks.

  - The function *double* $Cost_1$ specified the cost of the resource r_1. It returns the marking of the place $r_1 = 1$ multiplied by 9 which is the cost of r_1.

  - The function *double* $Cost_2$ specified the cost of the resource r_2. It returns the marking of the place $r_2 = 1$ multiplied by 15 which is the cost of r_2.

- The function *double* $Cost_3$ specified the cost of resource r_3. It returns the marking of the place $r_3 = 1$ multiplied by 22 which is the cost of r_3.
- The function *double* $Cost_4$ specified the cost of the resource r_4. It returns the (marking of the place $r_4 = 1$) multiplied by 7 which is the cost of r_4.
- The function *double* $Speed_1$ specified the speed of the resource r_1. It returns the (marking of the place $r_1 = 1$) multiplied by 10 which is the speed of r_1.
- The function *double* $Speed_2$ specified the speed of the resource r_2. It returns the (marking of the place $r_2 = 1$) multiplied by 14 which is the speed of r_2.
- The function *double* $Speed_3$ specified the speed of the resource r_3. It returns the (marking of the place $r_3 = 1$) multiplied by 5 which is the speed of r_3.
- The function *double* $Speed_1$ specified the speed of the resource r_4. It returns the (marking of the place $r_4 = 1$) multiplied by 12 which is the speed of r_4.

The *CSPL* file provides an input parameters to the Matlab programs to conduct workflow scheduling. The file is mapped into Matlab according to the *SPN2MATLAB* algorithm which is described in Section 5.5.3. The algorithm designed to map the *CSPL* file into Matlab for workflow scheduling approach is describe as follows:

- **Generic *SPN2MATLAB* algorithm for mapping *SPN* into Matlab for workflow scheduling approach.**

Step 1 : Read file (CSPL file of SPN)
Step 2 : While not EOF () do:
Step 3 : Read word Place("x")
Step 4 : Extract value of x
Step 5 : Put x in list of Places
Step 6 : Read word init("x",y)
Step 7 : Extract values of x and y
Step 8 : Put x in list of init1
Step 9 : Put y in list of init2
Step 10 : Read word rateval("x",y)

Step 11 : Extract values of x and y

Step 12 : Put x in list of rateval1

Step 13 : Put y in list of rateval2

Step 14 : Read word iarc("x","y")

Step 15 : Extract value of x and y

Step 16 : Put x in list of iarc1

Step 17 : Put y in list of iarc2

Step 18 : Read word oarc("x","y")

Step 19 : Extract value of x and y

Step 20 : Put x in list of oarc1

Step 21 : Put y in list of oarc2

Step 22 : Read word return(mark("x")*y)

Step 23 : Extract values of x and y

Step 24 : Put x in list of Arr-Rate1

Step 25 : Put y in list of Arr-Rate2

Step 26 : Read word return(mark("x")*y)

Step 27 : Extract values of x and y

Step 28 : Put x in list of return1

Step 29 : Put y in list of return2

Step 30 : Read word return(rate("x")*mark("y")*z)

Step 31 : Extract values of y and z

Step 32 : Put y in list of speed1

Step 33 : Put z in list of speed2

Step 34 : Read word return(rate("x")*y)

Step 35 : Extract values of x and y

Step 36 : Put x in list of mean1

Step 37 : Put y in list of mean2

Step 38 : If list Places = iarc2

Step 39 : Put iarc1 in list of iarc3

Step 40 : Put iarc2 in list of iarc4

Step 41 : If list iarc3 = oarc2 and iarc4 = oarc1

Step 42 : Put oarc1 in list of resouces1

Step 43 : Put oarc2 in list of resouces2

Step 44 : Else

Step 45 : Put iarc3 in list of Task1

Step 46 : Put oarc2 in list of Task2

Step 47 : If list return1 $\neq$ speed1

Step 48 : DSR(1).Arr-Rate = ’0.00’

Step 49 : Put Arr-Rate2 in list of Arr-Rate

Step 50 : If list resouces2 = init1

Step 51 : Put init1 in list of resouces3

Step 52 : Put init2 in list of resouces4

Step 53 : If list resouces3 = return1

Step 54 : Put return2 in list of resouces5

Step 55 : for i = 1: number of resources

Step 56 : DSR(i).Resource = ’string’

Step 57 : Put resource2 in list of Resource

Step 58 : for i = 1: number of resources

Step 59 : DSR(i).Cost = ’0.00’

Step 60 : Put resouces5*resouces4 in list of Cost

Step 61 : for i = 1: number of resources

Step 62 : DSR(i).Speed = ’0.00’

Step 63 : Put speed2 in list of Speed

Step 64 : for i = 1: number of Tasks

Step 65 : DSR(i).Task = ’string’

Step 66 : Put Task1 in list of Task

Step 67 : for i = 1: number of Tasks

Step 68 : DSR(i).Mean = ’0.00’

Step 69 : Put mean2 in list of Mean

Step 70 : for i = 1: number of Resources

Step 71 : DSR(i).SCHEDULING = struct(’Task’,’string’,’Mean’,’0.00’,
’Resource’,’string’,’Cost’,’0.00’,’Speed’,’0.0’,,,’Arr-Rate’,’0.00’)

The output file of the *CSPL* as shown in Table 6.18 contains the parameters required to calculate workflow scheduling. Providing a solution and the details of functions of grid workflow scheduling in WOSE methodology are out of scope of the thesis. More information can be found in the original publication [21].

Table 6.18: The file of workflow scheduling example

T_1,	7.5,	r_1,	9,	10,	3.5
T_1,	7.5,	r_3,	22,	5,	
T_1,	7.5,	r_4,	7,	12	
T_2,	16.5,	r_2,	15,	14,	
T_2,	16.5,	r_4,	7,	12,	

6.6 Summary

This chapter has presented two examples to demonstrate the applicability of the framework utilising the software support tool proposed in Chapter 5. The experiments were conducted using realistic business processes models. The results have shown that the generic framework proposed in this study is capable of providing the required functionalities. The results show how some commonly used parameters of algorithms of performance evaluation and optimisation of business processes can be extracted and automated performance evaluations and optimisation can then be conducted. The quantitative evidence supports the conclusion that the proposed framework is feasible and effective. The experiments conducted concerning the availability of resources and workflow scheduling have clearly demonstrated the benefits of using the framework for real-world business process applications. The next chapter presents a summary of the research and the contributions made by this thesis, as well as suggestions for possible future study.

Chapter 7

Conclusions and Future Work

This chapter concludes the thesis with a discussion of the findings of this research and with future work. The discussion involves the key observations made throughout the course of this research, and identifies its main contributions. The limitations of the proposed approach to the quantitative evaluation of business processes are also underlined and corresponding future activities that can push forward research in the area are highlighted. Section 7.1 summarises the research reported in each chapter. Section 7.2 then outlines the contributions made by this thesis. Section 7.3 presents the limitations of this research. Finally, how the knowledge gained in this study can be applied in the future to improve the quantitative evaluation of business processes is discussed in Section 7.4.

7.1 Summary of the Research

In order to design a generic modelling framework for the quantitative evaluation of business processes, suitable modelling techniques, algorithm solutions and software support tools are required. In this thesis, a new modelling framework is developed for quantitative evaluation of business process based on a novel engineering methodology. The methodology allows for any quantitative evaluation or optimisation algorithm to be applied to as broad a class of business processes as possible. The framework consists of three main modules: a business process specification module, a stochastic Petri net module, and an algorithms module. The business process specification module defines a business model in an *MWF-wR* formalism in order to identify its quantitative specifications. The second module includes the mapping of the business process model into a stochastic Petri net model enriched with business process information. The third module presents algorithm solutions for the performance analysis and optimisation of business processes by extracting appropriate parameters from the *SPN* models. The framework integrates the three modules through the development of software support tools able to automatically conduct performance analysis and optimisation.

7.2 Summary of Contributions

The framework is capable of representing business process in a *MWF-wR* model, and leveraging many techniques to analyse models for a set of performance and optimisation metrics. Any analysis algorithm for any business process can be plugged in and then a stochastic Petri net models were generated which can provide the parameters needed for conducting the performance analysis and optimisation approach.

This research has contributed to the field of the quantitative evaluation of business processes through the following advances:

- **A generic framework for modelling business processes**
  The generic framework for the quantitative evaluation of business processes is a comprehensive methodology due to the integration of three main modules. Each module fulfils a specific function of the methodology. The modules are:
  - **Business process specification module**:
    This module defines business process elements specified in the developed formalism *MWF-wR*. This generic describes business processes and specifies the relationships between their elements which are the basic parameters in techniques used to conduct performance analysis and optimisation.
  - **Stochastic Petri net module**:
    In this module, a stochastic Petri net model is generated. The model is a results of mapping a *MWF-wR* business process model into an *SPN* model. A new algorithm, *MWF2SPN*, has been designed to complete the mapping relations. The *SPN* is used as intermediate solver between the business process specifications and the application of performance analysis and optimisation techniques. It is used to provide the correct input parameters for the algorithms used in these techniques.

  - **Algorithms module**:

    This module represents performance analysis and optimisation techniques where any algorithm can be plugged in. The output of the *SPN* model is mapped into Matlab for the execution of the algorithms. A new algorithm, *SPN2MATLAB*, has been designed to complete the mapping relations.

- **The Modelling Workflows with Resources *(MWF-wR)* model**

  The *MWF-wR* is a novel generic formalism capable of modelling business processes and representing their quantitative aspects. It allows the main elements of business processes and the functional relationships between them to be defined.

- **Software support tools**

  The software support tools were developed in this research are based on the requirements of the framework which is built by using Java as object oriented programming language. They integrate the functionalities of the three modules in the framework and facilitate its automated working. These software support tools include the following:

  - **A user-friendly GUI**:

    A user-friendly GUI interface is designed in which to insert information from the business process models and to display the results of performance analysis and optimisation.

  - **MWF2SPN algorithm**

    The *MWF2SPN* is an algorithm used to automate the mapping relation between the *MWF-wR* business process model and the *SPN* model.

  - **SPN2MATLAB algorithm**

    This algorithm automatically maps the *SPN* model into Matlab and generates the correct input parameters for different types of algorithms used in business process analysis and optimisation.

- **Current state of the art of business process modelling and analysis**

  A survey of the business process and workflow literature was carried out as part of this research. Existing definitions, modelling, analyses of business process and methodologies of performance analysis and optimisation discussed. Based on the literature survey, gaps in knowledge are highlighted including the lack of a comprehensive quantitative approach for the performance analysis and optimisation of business processes.

7.3 Limitations of the Research

The research described in this thesis has attempted to design a methodology which is as generic as possible, and which can be used to automatically conduct the performance analysis and optimisation of business processes. However, this research inevitably has some limitations, the most important of which are as follows:

- The framework presented in this work has covered important aspects of the quantitative evaluation of business processes. However, it is difficult to design a single format capable of covering all solution algorithms. This is due to the existence of different types of modelling techniques required for each different algorithm, as shown in Chapter 2.
- The proposed *MWF-wR* formalism focuses only on the acyclic business process workflows. Therefore, it may not be suitable correct for cyclic workflows.
- The *MWF2SPN* algorithm for mapping the relations between business process models and *SPN* depends on the type of modelling technique used, which makes it difficult to map some types of business process models to *SPN* in a single format.
- Because of the existence of different types of algorithms for business process analysis and optimisation, different formats are required for the *SPN2MATLAB* algorithm to map the *SPN* into Matlab.
- Different algorithms pose different restrictions on the class of business process models to which the algorithms can be applied.

7.4 Suggestions for Future Work

The proposed generic framework for the quantitative evaluation of business process is flexible, and there are many ways in which it could be extended in future research. Several promising potential extensions are outlined below:

- The Business Process Execution Language (BPEL) and Business Process Modelling Notation (BPMN) have been extensively used in business process applications in the past few years. The framework developed in the present study can now be used in future research involving the analysis of business processes using these two modelling techniques. Therefore, it would be well worth investing some effort in improving the applicability of the framework to include the applications of which use BPEL and BPMN.

- The current *MWF-wR* formalism can be extended to include the cyclic workflows.

- The methodology can be extended by designing a generic algorithm which would fully automate the mapping of *SPN* models into different Matlab algorithms of performance analysis and optimisation.

- Algorithms could be classified based on the class of business models to which they apply. This would allow the systematic tailoring of our generic methodology to each class.

- The experimental studies carried out here clearly demonstrate the capability of the proposed framework to be used in conducting resources availability enhancement and the scheduling of business process workflows. However, further experiments could be carried out for additional evaluation and optimisation algorithms.

References

[1] M. Hammer and S. A. Stanton, *The Reengineering Revolution: A Handbook*. New York: HarperBusiness, 1995.

[2] G. T. Lumpkin and G. G. Dess, "E-business strategies and internet business models: How the internet adds value," *Organizational Dynamics*, vol. 33, No. 2, pp. 161–173, 2004.

[3] S. M. Siha and G. H. Saad, "Business process improvement: empirical assessment and extensions," *Business Process Management Journal*, vol. 14, no. 6, pp. 778–802, 2008.

[4] K. K. Ryan, S. L. Stephen, and W. L. Eng, "Business process management (BPM) standards: a survey," *Business Process Management Journal*, vol. 15, no. 5, pp. 744–791, 2009.

[5] H. Smith and P. Fingar, *Business Process Management: The Third Wave*. Meghan Kiffer Press,, 2003.

[6] M. Hammer, "Reengineering work: dont automate, obliterate," *Harvard Business Review*, 1990.

[7] T. H. Davenport and J. E. Short, "The new industrial engineering: Information technology and business process redesign," *Sloan Management Review*, vol. 31, pp. 11–27, 1990.

[8] W. M. P. van der Aalst and M. Weskez, "Advanced topics in workflow management: Issues, requirements, and solutions," *Journal of Integrated Design and Process Science*, 2003.

[9] G. Kappel, S. Rausch-Schott, and W. Retschitzegger, "A framework for workflow management systems based on objects, rules and roles," *ACM Comput. Surv.*, vol. 32, March 2000.

[10] U. Bititci, A. Carrie, and T. Turner, "Integrated performance measurement systems: Structure and dynamics," *, in Business Performance Measurement: Theory and Practice. Neely, Andrew, editor. Cambridge University Press*, pp. 174–197, 2002.

[11] H. Jonkers and H. M. Franken, "Quantitative modelling and analysis of business processes," *in A. Bruzzone and E. Kerckhoffs, eds., Simulation in Industry: Proceedings 8th European Simulation Symposium*, vol. 1, pp. 175–179, 1996.

[12] Y. Ding, "A prescriptive approach to business process modelling," *Proceeding of Information Resources Management Association (IRMA), USA*, 2007.

[13] A. van Moorsel, "Metrics for the Internet age: Quality of experience and quality of business," *Technical Report HPL-2001-179, HP Labs*, August 2001.

[14] S. Fenz, A. Ekelhart, and T. Neubauer, "Business process-based resource importance determination," *Springer-Verlag, Berlin, Heidelberg*, pp. 113–127, 2009.

[15] "Business process modeling language (BPML)," *Business Process Management Institute. Available at http://www.bpmi.org*, 2003.

[16] "Business process modeling notation v1.1 (BPMN)," *OMG document formal*, 2008.

[17] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Business Process Execution Language for Web Services Version 1.1," 5 May 2003. [Online]. Available: ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf

[18] J. Al-Tuwaijari and A. P. A. van Moorsel, "Automated generic weak point analysis of business process." *proceedings of the 26th UK Performance Engineering Workshop,*, UK. 2010.

[19] ——, "Generic business process model for quantitative evaluation," *Poster Session at The Newcastle Connection 2012, UK*, August, 2012. [Online]. Available: http://www.cs.ncl.ac.uk/publications/trs/papers/1348.pdf.

[20] L. Xie, J. Luo, J. Qiu, J. A. Pershing, Y. Li, and Y. Chen, "Availability weak point analysis over an SOA deployment framework." *IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubioquitous Networks and Services, NOMS 2008, Salvador, Bahia, Brazil*, 2008. [Online]. Available: http://dblp.uni-trier.de/db/conf/noms/noms2008.html#XieLQPLC08

[21] Y. Patel, A. S. M. Gough, and J. Darlington, "Grid workflow scheduling in WOSE," *In UK e-Science Programme All Hands Meeting Conference, AHM2006*, 2006.

[22] K. Vergidis, A. Tiwari, and B. Majeed, "Business process improvement using multi-objective optimisation," *BT Technology Journal*, vol. 24, no. 2, pp. 229–235, Apr. 2006. [Online]. Available: http://dx.doi.org/10.1007/s10550-006-0065-2

[23] P. Feiler and W. Humphrey, "Software process development and enactment: concepts and definitions," *Second International Conference on the Continuous Software Process Improvement,1993.*, pp. 28 –40, feb 1993.

[24] V. Grover and W. Kettinger, *Business Process Change: Reengineering Concepts, Methods and Technologies.* Hershey, PA, USA: IGI Publishing, 1995.

[25] M. Havey, *Essential Business Process Modeling.* USA: O'Reilly Media, Inc., 2005.

[26] J. Z. Li, R. B. Maguire, and Y. Yao, "A business process centered software analysis method." *International Journal of Software Engineering and Knowledge Engineering*, pp. 153–168, 2003.

[27] D. G. Platt, "Process modelling and process support environments for design management," *PhD thesis, Univ. of Bristol, England*, 1994.

[28] R. Medina-Mora, T. Winograd, and R. Flores, "Action workflow as the enterprise integration technology," *Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society*, vol. 16, no. 2, pp. 219–229, June 1993.

[29] T. H. Davenport and D. B. Stoddard, "Reengineering: Business change of mythic proportions?" *MIS Quarterly*, vol. 18, no. 2, pp. 121–127, 1994.

[30] B. Warboys, P. Kawalek, I. Robertson, and M. Greenwood, *Business Information Systems: a Process Approach.* McGraw-Hill Publishing Company, 1999.

[31] G. A. Pall, *Quality Process Management.* Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[32] M. Hammer and J. Champy, *Re-engineering the Corporation; A Manifesto for Business Revolution.* London, U.K.: Brealey, 1993.

[33] F. Hewitt, "Business process innovation in the mid-1990s." *Integrated Manufacturing*, vol. 6, pp. 18–26, 1995.

[34] H. A. Reijers, *Design and control of workflow processes: business process management for the service industry.* Berlin, Heidelberg: Springer-Verlag, 2003.

[35] V. Irani, Zahir Hlupic and G. M. Giaglis, "Business process reengineering: an analysis perspective," *International Journal of Flexible Manufacturing Systems*, vol. 14, pp. 5–10, 2002.

[36] J. Becker, M. Kugeler, and M. Rosemann, *Process management : a guide for the design of business processes / Jörg Becker, Martin Kugeler, Michael Rosemann, editors.* Springer, Berlin , Heidelberg, 2003.

[37] J. Lonchamp, "A structured conceptual and terminological framework for software process engineering," *in Proceedings of the Second International Conference on the Software Process*, 1993.

[38] P. J. Agerfalk, G. Goldkuhl, and S. Cronholm, "Information systems actability engineering - integrating analysis of business processes and usability requirements," *In Proceedings of the 4th International Workshop on the Language Action Perspective on Communication Modelling Copenhagen, Denmark*, pp. 1245–1252, 1999.

[39] Y. Fan, *Fundamental of Workflow Management Technology.* Springer-Verlag, New York, 2001.

[40] K. B. C. Saxena, "Re-engineering public administration in developing countries," *Long Range Planning*, vol. 29, no. 5, pp. 703–711, 1996.

[41] M. Zairi, "Business process management: a boundaryless approach to modern competitiveness," *Business Process Management Journal*, vol. 3, no. 1, pp. 64–80, 1997.

[42] S. Alter, *Information Systems: A Management Perspective.* Menlo Park, Ca: The Benjamin Cummings Publishing Company Inc, 1996.

[43] A. Lindsay, D. Downs, and K. Lunn, "Business processes–attempts to find a definition," *Information and Software Technology*, vol. 45, no. 15, pp. 1015–1019, December 2003.

[44] P. Volkner and B. Werners, "A decision support system for business process planning," *European Journal of Operational Research*, vol. 125, no. 3, pp. 633–647, September 2000.

[45] N. Melao and M. Pidd, "A conceptual framework for understanding business process and business processing modelling," *Information Systems Journal*, vol. 10, pp. 105–129, 2000.

[46] W. M. P. van der Aalst, *A class of Petri nets for modeling and analyzing business processes*, ser. Computing science reports 95/26. Eindhoven University of Technology, Department of Mathematics and Computing Science, Eindhoven, 1995. [Online]. Available: http://books.google.co.uk/books?id=yTzTGwAACAAJ

[47] L. V. Orman, "A model management approach to business process reengineering," *Journal of Management Information System.*, vol. 15, pp. 187–212, June 1998.

[48] T. Basten and W. M. van der Aalst, "Inheritance of behavior," *Journal of Logic and Algebraic Programming*, vol. 47, no. 2, pp. 47 – 145, 2001.

[49] E. A. Stohr and J. L. Zhao, "Workflow automation: Overview and research issues," *Information Systems Frontiers*, vol. 3, no. 3, pp. 281–296, 2001.

[50] W. M. P. van der Aalst, "The application of Petri nets to workflow management," *The journal of Circuits, Systems and Computers*, vol. 8 No. 1, pp. 21–66, 1998.

[51] S. Biazzo, "Approaches to business process analysis: a review," *Business Process Management Journal*, vol. 6, no. 2, pp. 99–112, 2000.

[52] W. M. P. van der Aalst and K. M. V. Hee, "Business process redesign: a Petri-net-based approach," *Computers in Industry*, vol. 29, no. 1-2, pp. 15–26, 1996.

[53] M. Castellanos, F. Casati, U. Dayal, and M.-C. Shan, "A comprehensive and automated approach to intelligent business processes execution analysis," *Distrib. Parallel Databases*, vol. 16, pp. 239–273, November 2004.

[54] W. M. P. van der Aalst, A. H. M. T. Hofstede, and M. Weske, "Business process management: A survey." Springer-Verlag, 2003, pp. 1–12.

[55] W. M. P. van der Aalst, "Making work flow: On the application of Petri nets to business process management." *23rd International Conference on Applications and Theory of Petri Nets*, vol. 2360, pp. 1–22, jun 2002.

[56] B. Wilson, *Systems: concepts, methodologies, and applications (2nd ed.).* New York, NY, USA: John Wiley & Sons, Inc., 1990.

[57] S. Beer, C. Strachey, R. Stone, Torrance., M. in Bullock, A., Stallybrass, O., and S. Trombley, *The Fontana Dictionary of Modern Thought.* revised edition, Fontana, London, 1988.

[58] R. S. Aguilar-Savn, "Business process modelling: Review and framework," *International Journal of Production Economics*, vol. 90, no. 2, pp. 129 – 149, 2004.

[59] M. Cumberlidge, *Business Process Management with JBoss jBPM: A Practical Guide for Business Analysts.* Packt Publishing, 2007.

[60] W. J. Kettinger, J. T. C. Teng, and S. Guha, "Business process change: A study of methodologies, techniques, and tools," *Journal of Management Information Systems Quarterly*, vol. 21, no. 1, pp. 55–80, 1997.

[61] P. Kueng, P. Bichler, P. Kawalek, and M. Schrefl, "How to compose an object-oriented business process model?" *Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering : principles of method construction and tool support: principles of method construction and tool support*, pp. 94–110, 1996.

[62] G. M. Giaglis, "A taxonomy of business process modelling and information systems modelling techniques," *Information Systems Journal*, vol. 13, no. 2, pp. 1–34, 2001.

[63] K. Vergidis, A. Tiwari, and B. Majeed, "Business process analysis and optimization: Beyond reengineering," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 1, pp. 69 –82, jan. 2008.

[64] J. Koenig, "JBoss JBPM white paper, version 2004," November, 2004.

[65] W. M. P. van der Aalst and A. H. M. ter Hofstede, "YAWL: yet another workflow language," *Information Systems*, vol. 30, no. 4, pp. 245–275, Jun. 2005.

[66] M. Koubarakis and D. Plexousakis, "A formal framework for business process modelling and design," *Information Systems*, vol. 27, pp. 299–319, July 2002.

[67] W. M. P. van der Aalst, "Challenges in business process management: Verification of business processes using Petri nets," *Bulletin of the European Association for Theoretical Computer Science*, vol. 80, pp. 174–198, 2003.

[68] I. Hofacker and R. Vetschera, "Algorithmical approaches to business process design," *Computers and Operations Research*, vol. 28, no. 13, pp. 1253–1275, Nov. 2001.

[69] A. Tiwari, K. Vergidis, and B. Majeed, "Evolutionary multi-objective optimization of business processes," *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 3091 –3097, 2006.

[70] S. G. Powell, M. Schwaninger, and C. Trimble, "Measurement and control of business processes," *System Dynamics Review*, vol. 17, no. 1, pp. 63–91, 2001.

[71] G. Valiris and M. Glykas, "Critical review of existing BPR methodologies: The need for a holistic approach," *Business Process Management Journal*, vol. 5, no. 1, pp. 65–86, 1999.

[72] N. Russell, Arthur, D. Edmond, and W. M. P. van der Aalst, "Workflow Resource Patterns," Eindhoven University of Technology, Tech. Rep., 2004.

[73] M. Wilson, "Walking the BPM talk how Appian uses its own technology to drive its business," *International Journal of Operations & Production Management*, 2009. [Online]. Available: http://www.bptrends.com/publicationfiles/09-08-CS-Appian-MarcWilson.doc-cap-82708.pdf

[74] J. Westerman, "The case for business process management," 2009. [Online]. Available: http://www.bptrends.com/publicationfiles/04-09-CS-Case-for-BPM-TIBCO. doc.pdf.

[75] S. D. P. Flapper, L. Fortuin, and P. P. Stoop, "Towards consistent performance management systems," *International Journal of Operations & Production Management*, vol. 16, pp. 27–37, 1996.

[76] M. H. Jansen-Vullers, M. W. N. C. Loosschilder, P. A. M. Kleingeld, and H. A. Reijers, "Performance measures to evaluate the impact of best practices," *Proceedings of Workshops and Doctoral Consortium of the 19th International Conference on Advanced Information Systems Engi-neering (BPMDS workshop),*, pp. 359–368, 2007.

[77] A. Neely, M. Gregory, and K. Platts, "Performance measurement system design: A literature review and research agenda," *International Journal of Operations & Production Management,*, vol. 25, no. 12, pp. 1228–1263, 2005.

[78] A. De Toni and S. Tonchia, "Performance measurement systems: models, characteristics and measures," *International Journal of Operations & Production Management*, vol. 21, no. 1/2, pp. 46–70, 2001.

[79] H. A. Reijers, *Design and control of workflow processes: business process management for the service industry.* Berlin, Heidelberg: Springer-Verlag, 2003.

[80] W. M. P. van der Aalst, "Business process management: A personal view," *Business Process Management Journal*, vol. 10, no. 2, pp. 135 –139, 2004.

[81] J. Li, Y. Fan, and M. Zhou, "Performance modeling and analysis of workflow," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, no. 2, pp. 229 – 242, March 2004.

[82] K. Salimifard and M. Wright, "Petri net-based modelling of workflow systems: An overview," *European Journal of Operational Research*, vol. 134, no. 3, pp. 664–676, 2001.

[83] Y. Zhou and Y. Chen, "Project-oriented business process performance optimization," *IEEE International Conference on Systems, Man and Cybernetics.*, vol. 5, pp. 4079 – 4084, October 2003.

[84] K. Vergidis, A. Tiwari, B. Majeed, and R. Roy, "Optimisation of business process designs: An algorithmic approach with multiple objectives." *International Journal of Production Economics*, vol. 109, no. 1-2, pp. 105–121, 2008.

[85] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various qos requirements," *IEEE Transactions on Systems Management and Cybernetics Part C*, vol. 39, pp. 29–43, January 2009.

[86] K. V. Hee, H. Reijers, E. Verbeek, and L. Zerguini, "On the optimal allocation of resources in stochastic workflow nets," *In:Djemame K, Kara M, editors. Proceedings of the Seventeenth UK Performance Engineering Workshop.*, pp. 23–34, 2001.

[87] F. Kamrani, R. Ayani, and F. Moradi, "A framework for simulation-based optimization of business process models," *SIMULATION: Transactions of The Society for Modeling and Simulation International, Special Issue. SIMULATION published online 13 September 2011, Available from: http://sim.sagepub.com/content/early/2011/09/10/0037549711417880. (Accessed on 28th February, 2012).*

[88] W. M. P. van der Aalst, "Petri net based scheduling," *OR Spektrum*, vol. 18, no. 4, pp. 219–229, 1995.

[89] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review," *Computers and Chemical Engineering*, vol. 28, no. 11, pp. 2109–2129, 2004.

[90] P. Senkul and I. Toroslu, “An architecture for workflow scheduling under resource allocation constraints,” *Information Systems Journal*, vol. 30, no. 5, pp. 399–422, 2005.

[91] C. Floudas and X. Lin, “Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications,” *Annals of Operations Research*, vol. 139, pp. 131–162, 2005.

[92] J. Kallrath, “Planning and scheduling in the process industry,” *OR Spectrum*, vol. 24, no. 3, pp. 219–250, 2002.

[93] J. M. Pinto and I. E. Grossmann, “Assignment and sequencing models for the scheduling of process systems,” *Annals of Operations Research*, 1998.

[94] J. Yu, R. Buyya, and C. K. Tham, “Cost-based scheduling of scientific workflow application on utility grids,” *Proceedings of the First International Conference on e-Science and Grid Computing*, pp. 140–147, 2005.

[95] P. Senkul, M. Kifer, and I. H. Toroslu, “A logical framework for scheduling workflows under resource allocation constraints,” *Proceedings of the 28th international conference on Very Large Data Bases*, pp. 694–705, 2002.

[96] C. Combi and G. Pozzi, “Task scheduling for a temporalworkflow management system,” *Proceedings of the Thirteenth International Symposium on Temporal Representation and Reasoning*, pp. 61–68, 2006.

[97] K. M. v. Hee and H. A. Reijers, “Using formal analysis techniques in business process redesign,” *Business Process Management, Models, Techniques, and Empirical Studies*, pp. 142–160, 2000.

[98] S. Li and H. Zhu, “Generalized stochastic workflow net-based quantitative analysis of business process performance,” *Science And Technology*, pp. 1040–1044, 2008.

[99] K. S. Trivedi, G. Ciardo, M. Malhotra, and R. A. Sahner, “Dependability and performability analysis,” *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance ’93 and Sigmetrics ’93*, pp. 587–612, 1993.

[100] M. Jansen-Vullers and M. Netjes, "Business process simulation - a tool survey," *Management*, vol. 834, no. 1-2, pp. 77–96, 2006.

[101] A. Gunasekaran and B. Kobu, "Modelling and analysis of business process reengineering," *International Journal of Production Research*, vol. 40, no. 11, pp. 2521–2546, 2002.

[102] W. M. P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell, "Business process simulation," *Handbook on Business Process Management 1*, pp. 313–338, 2010.

[103] N. R. Adam, V. Atluri, and W.-K. Huang, "Modeling and analysis of workflows using Petri nets," *Journal of Intelligent Information Systems*, vol. 10, pp. 131–158, March 1998. [Online]. Available: http://dl.acm.org/citation.cfm?id=290056.290062

[104] W. M. P. van der Aalst, "Three good reasons for using a Petri-net-based workflow management system," *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, pp. 179–201, 1996.

[105] V. Atluri and W.-K. Huang, "A Petri net based safety analysis of workflow authorization models," *The Journal of Computer Security*, vol. 8, pp. 209–240, August 2000.

[106] K. Sarshar, T. Theling, P. Loos, and M. Jerrentrup, "Integrating process and organization models of collaborations through object Petri nets," *In Proc. of the Third GI-Workshop XML4BPM XML Integration and Transformation For Business Process Management*, 2006.

[107] J. Dehnert and P. Rittgen, "Relaxed soundness of business processes," *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*, pp. 157–170, 2001.

[108] H. Schlingloff, A. Martens, and K. Schmidt, "Modeling and model checking web services," *Electronic Notes in Theoretical Computer Science*, vol. 126, pp. 3–26, March 2005.

[109] Y. Tang, L. Chen, K.-T. He, and N. Jing, "SRN: An extended Petri-net-based workflow model for web service composition," *In Proc. of the IEEE International Conference on Web Services*, pp. 591–599, 2004.

[110] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING*, vol. 1, pp. 11–33, January 2004.

[111] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets.* New York, NY, USA: John Wiley & Sons, Inc., 1995.

[112] M. Ajmone Marsan, G. Conte, and G. Balbo, "A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems," *ACM Transactions on Computer Systems*, vol. 2, pp. 93–122, May 1984.

[113] G. Ciardo, J. Muppala, and K. Trivedi, "SPNP: Stochastic Petri Net Package," *Proceedings of the Third International Workshop on Petri Nets and Performance Models PNPM89*, vol. 96, pp. 142–151, 1989.

[114] K. S. Trivedi, *SPNP Users Manual, version 6.0 ed.* Duke University, 1999.

[115] Matlab, "The language of technical computing, 2012," *Available from: http://www.mathworks.com/products/ matlab/.*

[116] Sun Microsystems. Java, 2012. http://www.java.com.

[117] M. A. Marsan, "Stochastic petri nets: an elementary introduction," *In Advances in Petri Nets, Springer-Verlag New York, Inc.*, pp. 1–29, 1990.

[118] D. Hollingsworth, "Workflow management coalition - the workflow reference model," Workflow Management Coalition, Tech. Rep., Jan. 1995.

[119] E. Oren and A. Haller, "Formal frameworks for workflow modelling," *Digital Enterprise Research Institute, National University of Ireland, Galway*, Apr. 2005.

[120] W. M. P. van der Aalst and K. M. van Hee, *Workflow Management: Models, Methods, and Systems.* Cambridge, MA, USA: MIT Press, 2004.

[121] W. M. P. van der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow patterns," *Distributed Parallel Databases*, vol. 14, no. 1, pp. 5–51, Jul. 2003.

[122] H. Lin, Y. Fan, and T. Loiacono Eleanor, "A practical scheduling method based on workflow management technology," *The International Journal of Advanced Manufacturing Technology*, vol. 24, no. 11-12, pp. 919–924, dec 2004.

[123] N. R. Adam, V. Atluri, and W.-K. Huang, "Modeling and analysis of workflows using Petri nets," *Journal of Intelligent Information Systems*, vol. 10, no. 2, Mar. 1998.

[124] B. R. Haverkort, *Performance of Computer Communication Systems: A Model-Based Approach.* New York, NY, USA: John Wiley & Sons, Inc., 1998.

[125] M. K. Molloy, "Performance analysis using stochastic Petri nets," *IEEE Transactions on Computers*, vol. 31, no. 9, pp. 913–917, Sep. 1982. [Online]. Available: http://dx.doi.org/10.1109/TC.1982.1676110

[126] D. E. Perry and A. L. Wolf, "Foundations for the study of software architecture," *SIGSOFT Software Engineering Notes*, vol. 17, no. 4, pp. 40–52, Oct. 1992.

[127] J. McCall, *Factors in Software Quality: Preliminary Handbook on Software Quality for an Acquisiton Manager.* General Electric, November 1977, vol. 1-3, no. ADA049055.

[128] R. B. Grady and D. L. Caswell, *Software metrics: establishing a company-wide program.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1987.

[129] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice (2nd Edition).* Addison-Wesley Professional, Apr. 2003. [Online]. Available: http://www.worldcat.org/isbn/0321154959