MODELLING AND EXECUTION OF ASSET MANAGEMENT PROCESS VIA WORKFLOW AUTOMATION

A THESIS
Submitted in partial fulfilment of the requirements for the award of the degree of MASTER OF ENGINEERING (RESEARCH) in SCHOOL OF ENGINEERING SYSTEMS

by
SRIMANTH LINGAMANENI
B.Tech (C.S.E)

Faculty of Built Environment and Engineering Queensland University of Technology BRISBANE, AUSTRALIA May- 2010
I hereby certify that the work which is being presented in the thesis, entitled *Modelling and Execution of Asset Management Processes via Workflow Automation* in partial fulfilment of the requirements for the award of the degree of Master of Engineering (Research) and submitted in the School of Engineering Systems of the University is an authentic record of my own work carried out during a period from August, 2007 to December, 2009 under the supervision of Prof. Lin Ma, Prof. Prasad Yarlagadda, and Dr. Chun Ouyang, Queensland University of Technology.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other University.

(Srimanth Lingamaneni)
ACKNOWLEDGEMENT

This thesis is a living testimony to the numerous contributions of a galaxy of distinguished personalities whom I had the good fortune of being associated with. I deem it an honour and duty to acknowledge all help I received from these luminaries.

I am awed and overwhelmed as I bow to my most revered ‘mentor’ Prof. Lin Ma, Faculty of Built Environment and Engineering, QUT. There is much for me to learn from her artistic touch to academics and her meticulousness. It is an eternal honour to have worked as her student for such a long spell. Her support, personal guidance, thought provoking discussions and encouragement helped me glide through the upheavals, which are inevitably in-built into a research work.

Words desert me when I rise to offer my humble respects to my second guide, Prof. Prasad Yarlagadda, Faculty of Built Environment and Engineering, QUT. “It is the master who makes things easy” holds true for him. It is a great honour to work under his supervision. His fathomless knowledge always turned a pearl of advice to satiate my academic inquisitions. The support and kindness that he has laid on to me is much appreciated.

I take this as an honour to offer my humble respects to my third guide Dr. Chun Ouyang, Faculty of Information Technology, QUT. The endless streams of queries emerging from the bottom of my mind always had an ever-ready quest-quencher in her. I appreciate the kindness and affirm that she showed upon me. She always willingly snatched out time for me. I, Once again, bow my head in eternal reverence to all of my supervisors.

I heartily thankful the CIEAM and QUT for the scholarship they provided to me. Without their support, it may not be possible to conduct this research in a smooth fashion.

I take this as honour to thank Dr. Yong Sun for his suggestions and help provided through out this Research.
I avail the privilege to pour on paper, my regards to my parents, Venkatarao Lingamaneni and Bhaskaramba Lingamaneni. Their blessings, love, care, inspiration, seen and unseen blessings kept me sailing through the storms.

The active support centred by my revered brother Veerendra Lingamaneni, who has been a parent, a guide, a patron also a trouble saver. His untiring organizing efforts and his way of life will always be my driving force.

I thank all the souls who helped me in this herculean task.

Finally, I thank God, for all the blessings he has showered on me and helped me to achieve my true potential in this temporal world.

Srimanth Lingamaneni
ABSTRACT

An Asset Management (AM) life-cycle constitutes a set of processes that align with the development, operation and maintenance of assets, in order to meet the desired requirements and objectives of the stakeholders of the business. The scope of AM is often broad within an organization due to the interactions between its internal elements such as human resources, finance, technology, engineering operation, information technology and management, as well as external elements such as governance and environment. Due to the complexity of the AM processes, it has been proposed that in order to optimize asset management activities, process modelling initiatives should be adopted.

Although organisations adopt AM principles and carry out AM initiatives, most do not document or model their AM processes, let alone enacting their processes (semi-) automatically using a computer-supported system. There is currently a lack of knowledge describing how to model AM processes through a methodical and suitable manner so that the processes are streamlined and optimized and are ready for deployment in a computerised way. This research aims to overcome this deficiency by developing an approach that will aid organisations in constructing AM process models quickly and systematically whilst using the most appropriate techniques, such as workflow technology.

Currently, there is a wealth of information within the individual domains of AM and workflow. Both fields are gaining significant popularity in many industries thus fuelling the need for research in exploring the possible benefits of their cross-disciplinary applications. This research is thus inspired to investigate these two domains to exploit the application of workflow to modelling and execution of AM processes. Specifically, it will investigate appropriate methodologies in applying workflow techniques to AM frameworks.

One of the benefits of applying workflow models to AM processes is to adapt and enable both ad-hoc and evolutionary changes over time. In addition, this can automate an AM process as well as to support the coordination and collaboration of people that are
involved in carrying out the process. A workflow management system (WFMS) can be used to support the design and enactment (i.e. execution) of processes and cope with changes that occur to the process during the enactment.

So far few literatures can be found in documenting a systematic approach to modelling the characteristics of AM processes. In order to obtain a workflow model for AM processes commonalities and differences between different AM processes need to be identified. This is the fundamental step in developing a conscientious workflow model for AM processes. Therefore, the first stage of this research focuses on identifying the characteristics of AM processes, especially AM decision making processes. The second stage is to review a number of contemporary workflow techniques and choose a suitable technique for application to AM decision making processes. The third stage is to develop an intermediate ameliorated AM decision process definition that improves the current process description and is ready for modelling using the workflow language selected in the previous stage. All these lead to the fourth stage where a workflow model for an AM decision making process is developed. The process model is then deployed (semi-) automatically in a state-of-the-art WFMS demonstrating the benefits of applying workflow technology to the domain of AM.

Given that the information in the AM decision making process is captured at an abstract level within the scope of this work, the deployed process model can be used as an executable guideline for carrying out an AM decision process in practice. Moreover, it can be used as a vanilla system that, once being incorporated with rich information from a specific AM decision making process (e.g. in the case of a building construction or a power plant maintenance), is able to support the automation of such a process in a more elaborated way.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>TITLE</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANDIDATE’S DECLARATION</td>
<td>2</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>3</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>11</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>14</td>
</tr>
<tr>
<td>NOMENCLATURE AND ACRONYMS</td>
<td>15</td>
</tr>
<tr>
<td><strong>CHAPTER 1: INTRODUCTION</strong></td>
<td>17-24</td>
</tr>
<tr>
<td>1.1 Preamble</td>
<td>17</td>
</tr>
<tr>
<td>1.2 STATE of Research in Asset Management</td>
<td>18</td>
</tr>
<tr>
<td>1.3 Problem Statement</td>
<td>19</td>
</tr>
<tr>
<td>1.4 Motivation</td>
<td>22</td>
</tr>
<tr>
<td>1.5 Thesis Outline</td>
<td>23</td>
</tr>
<tr>
<td><strong>CHAPTER 2: LITERATURE REVIEW AND PROBLEM FORMULATION</strong></td>
<td>25-55</td>
</tr>
<tr>
<td>2.1 Overview</td>
<td>25</td>
</tr>
<tr>
<td>2.2 Asset management processes</td>
<td>25</td>
</tr>
<tr>
<td>2.2.1 Introduction</td>
<td>26</td>
</tr>
<tr>
<td>2.2.2 Asset management principles</td>
<td>27</td>
</tr>
<tr>
<td>2.2.3 Asset management processes</td>
<td>30</td>
</tr>
<tr>
<td>2.2.4 Data Collection for the Asset management processes</td>
<td>32</td>
</tr>
</tbody>
</table>
2.3 Gaps and Opportunities – Modelling and Execution of Asset Management Processes 37

2.4 Business Process Management and Workflow Technology 38

2.4.1 Introduction 39

2.4.2 Process Modelling and Execution 40

2.4.3 Workflow Technology 44

2.4.4 Languages, Systems and Tools 45

2.5 Gaps and Opportunities – Applications of Workflow Technology to Asset Management Processes 52

2.6 Problem Formulation 54

CHAPTER 3: METHODOLOGY AND TECHNIQUES 56-95

3.1 Overview 56

3.2 Methodology and Approach 56

3.3 Asset Management Split Decision Making (AMSDM) Process 63

3.3.1 Description of the AMSDM Process 63

3.3.2 Analysis of the AMSDM Process 72

3.3.3 Characteristics of the AMSDM Process 73

3.4 Association between AMSDM Process and WFM Technique 76

3.5 Yet Another Workflow Language (YAWL) 79

3.5.1 YAWL Principles 81

3.5.2 The YAWL System 82

3.5.3 Process Modelling and Execution using YAWL 86

3.6 Summary 95
CHAPTER 4: Ameliorating the AMSDM Process Definition  96-123

4.1 Introduction  96
4.2 Overview of Ameliorated Process Definitions  97
4.3 Control Flow Definition of the Ameliorated AMSDM process  105
4.4 Data Flow Definition of the Ameliorated AMSDM process  107
  4.4.1 Data Collection for the Complete AMSDM process  108
    4.4.1.1 Data Collection for the Ameliorated AMSDM Process  108
    4.4.1.2 Data Collection for the Ameliorated AM Decision Objective Identification Sub process  111
    4.4.1.3 Data Collection for the Ameliorated Option Ranking Sub process  112
4.4.2 Data Analysis for the AMSDM Process  115
4.5 Initial Resource modelling of the AMSDM process  117
  4.5.1 Organizational Structure  118
  4.5.2 Role-based Task Allocation  120
4.6 Summary  122

CHAPTER 5: Modelling and Execution of the AMSDM Process using Yet Another Workflow Language  124-168

5.1 A Workflow Model of the AMSDM Process  124
  5.1.1 AMSDM Process Control Flow Definition  125
  5.1.2 AMSDM Process Data Flow Modelling  135
    5.1.2.1 Data Definitions  135
    5.1.2.2 Data Mapping  142
5.1.3 AMSDM Process Resource Modelling

5.1.3.1 Organizational Data

5.1.3.2 Task-Resource Allocation Scheme

5.2 AMSDM Process Execution

5.3 Summary

CHAPTER 6: Conclusions and Scope of Future Work

6.1 Introduction

6.2 Summary of Findings

6.3 Scope for Future Work

REFERENCES

APPENDICES

Appendix A – YAWL Model for the AM Decision Objective Identification Sub-Process

Appendix B – YAWL Model for the Option Selection Sub process
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Relation between different decision making levels and the</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Corresponding detail and amount of needed data</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>Business Process Management Life-Cycles</td>
<td>39</td>
</tr>
<tr>
<td>3.1</td>
<td>A typical multi-scale decision making framework</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Concept of the split decision making process</td>
<td>60</td>
</tr>
<tr>
<td>3.3</td>
<td>Concept of the multi-agent system based split decision support system</td>
<td>62</td>
</tr>
<tr>
<td>3.4</td>
<td>The notations of Figures 2 to 6</td>
<td>66</td>
</tr>
<tr>
<td>3.5</td>
<td>Asset Management split decision making process</td>
<td>67</td>
</tr>
<tr>
<td>3.6</td>
<td>Asset management decision objectives identification process</td>
<td>68</td>
</tr>
<tr>
<td>3.7</td>
<td>High level asset condition assessment and prediction process</td>
<td>69</td>
</tr>
<tr>
<td>3.8</td>
<td>Option ranking process</td>
<td>70</td>
</tr>
<tr>
<td>3.9</td>
<td>Optimization process</td>
<td>71</td>
</tr>
<tr>
<td>3.10</td>
<td>Association between AMSDM process and YAWL workflow model</td>
<td>77</td>
</tr>
<tr>
<td>3.11</td>
<td>YAWL system architecture</td>
<td>83</td>
</tr>
<tr>
<td>3.12</td>
<td>Schema definitions for YAWL abstract syntax</td>
<td>89</td>
</tr>
<tr>
<td>3.13</td>
<td>Modelling elements in YAWL</td>
<td>91</td>
</tr>
<tr>
<td>3.14</td>
<td>State Transition diagram for Work distribution</td>
<td>93</td>
</tr>
<tr>
<td>4.1</td>
<td>Notations for the ameliorated AMSDM process</td>
<td>98</td>
</tr>
<tr>
<td>4.2</td>
<td>Ameliorated AM Split Decision Making process</td>
<td>100</td>
</tr>
<tr>
<td>4.3</td>
<td>Ameliorated AM decision objective identification process</td>
<td>101</td>
</tr>
<tr>
<td>4.4</td>
<td>Ameliorated AM objective identification process</td>
<td>102</td>
</tr>
<tr>
<td>4.5</td>
<td>Ameliorated Asset Condition Assessment and Prediction process</td>
<td>103</td>
</tr>
</tbody>
</table>
5.24 Manage Resourcing wizard for the step 1 of interaction point
5.25 Manage resourcing wizard for the step 5 of interaction point
5.26 Admin form for loading and launching the Case
5.26(a) Admin form for launching three cases
5.27(a) Administration tools in YAWL
5.27 Admin Screen for offering task to the Participants
5.28 List of Participants shown
5.29 Admin Screen for Work Queues while Executing
5.30 Execution of the task “Identify an AM decision”
5.31 Check box for decision making task
5.32 Output Screen after executing the task “Gather decision objectives information”
5.33 Output screen for the task “Gather asset conditions”
5.34 Output screen for the task “Define decision options”
A1 Control flow for the task “Define the attributes for each goal”
A2 Control flow for the “Audit the AM activity goals”
A3 Control flow for the “AM decision objective identification process”
A4 Control flow for the “AM objective identification process”
A5 Parameter definitions for “Defining attributes for each goal”
A6 XQuery for the input parameter of the task.
A7 Data passing for the task “Audit the AM activity goals”
A8 Variables and XQuery expression for “AM objective identification process”
A9 Output of the task “Define the goals for the AM activity”
A10 Output for the task “Define the attributes for each goal”
B1 Control flow for the “Option selection sub process”
B2  The task variables defined in the Option selection process  197
B3  Data passing information for the task “Choose an analytical model”  198

LIST OF TABLES

4.1  Goals and attributes for the specific AM activity  112
# NOMENCLATURE AND ACRONYMS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AASHTO</td>
<td>American Association of State Highway and Transportation Officials</td>
</tr>
<tr>
<td>AHP</td>
<td>Analytic Hierarchy Process</td>
</tr>
<tr>
<td>AM</td>
<td>Asset Management</td>
</tr>
<tr>
<td>AMSDM</td>
<td>Asset Management Split Decision Making</td>
</tr>
<tr>
<td>APWA</td>
<td>American Public Works Association</td>
</tr>
<tr>
<td>ARTBA</td>
<td>American Road and Transportation Builders Association</td>
</tr>
<tr>
<td>BP</td>
<td>Business Process</td>
</tr>
<tr>
<td>BPEL</td>
<td>Business process Execution Language</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business process Modelling Notation</td>
</tr>
<tr>
<td>BPR</td>
<td>Business Process Reengineering</td>
</tr>
<tr>
<td>CID</td>
<td>Case Identifier</td>
</tr>
<tr>
<td>CMMS</td>
<td>Computerised Maintenance Management Systems</td>
</tr>
<tr>
<td>CIEAM</td>
<td>Cooperative Research Centre for Integrated Engineering Asset Management</td>
</tr>
<tr>
<td>FHWA</td>
<td>Federal Highway Administration</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GPR</td>
<td>Ground Penetrating Radar</td>
</tr>
<tr>
<td>IDC</td>
<td>International Data Corporation</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>MCFE</td>
<td>Multistage Comprehensive Fuzzy Evaluation</td>
</tr>
<tr>
<td>NAMS</td>
<td>National Asset Management Strategy Committee</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>OECD</td>
<td>Organization of Economic Cooperation and Development</td>
</tr>
<tr>
<td>QC</td>
<td>Quality Control</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>TAC</td>
<td>Transportation Association of Canada</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
<tr>
<td>WERD</td>
<td>Western European Road Directors</td>
</tr>
<tr>
<td>WF</td>
<td>Workflow</td>
</tr>
<tr>
<td>WFM</td>
<td>Workflow Modelling</td>
</tr>
<tr>
<td>WS-CDL</td>
<td>Web Services Choreography Description Language</td>
</tr>
<tr>
<td>XPDL</td>
<td>XML Process Definition Language</td>
</tr>
<tr>
<td>YAWL</td>
<td>Yet Another Workflow Language</td>
</tr>
</tbody>
</table>
1.1 Preamble

In scientific enquiry, an experiment is a method of investigating particular types of research questions or solving particular types of problems. The experiment is a cornerstone in the empirical approach to acquire deeper knowledge about the world and is used in all branches of science and engineering. An experiment can be defined as a method of investigating less-known fields, solving practical problems and proving theoretical assumptions. The theoretical models are important in coming up with newer designs. Nevertheless they also should be subject of experimentation for a real test as they are based on certain assumptions.

Discovery of a novel scientific idea seldom happens. Serendipity along with life-long involvement of many researchers results in the birth of a new concept. However, the process of understanding and deriving applications continues. This leads to continuation of the procedure termed as research (rather “re-search”). The wheel was invented once. Nevertheless, research on the wheel continues until date. Need combined with a zeal to make things better, is reason behind research. Entities, either natural or manufactured, follow certain rules. A combination of many entities makes a system. Thus, research can be related with understanding of such systems even with the behaviour of the entities is well understood. Research is based on ‘Cause and Effect Analysis’. Experimentation on physical models along with fundamental mathematical concepts, leads to analytical or empirical expressions. The Cause and Effect Analysis combined with computational capabilities enhances the understanding of systems by development of process models. Thus development of a model became an important tool for research. In case of engineering processes, the role of process model becomes more significant as in becomes instrumental in productivity enhancement.

Engineering Asset Management (or Asset Management, in short), as technology, process or system, has also benefited by the exercise of research. The last ten years have tremendous growth in this area of research. A number of research articles dealing with
engineering asset management were published in various journals and conference proceedings. A deliberation on the publications reveals that a major portion within asset management related studies is covered by various investigations into technologies like Reliability, Process modelling, Life prediction, System thinking, etc. Various tools are being used to carry out the above stated investigation. These tools are Predicting of failure probability, asset life cycle, simulation analysis, Condition based monitoring, etc and computations intelligence based tools like Computerised maintenance management systems (CMMS), Cost-benefit analysis, Fuzzy evaluation, etc. The present state of research in engineering asset management is discussed in the following section.

1.2 State of Research in Asset Management

The state of research in asset management (AM) can be classified in to four categories as following:

1) Existing processes applied to existing techniques
2) Existing processes applied to new techniques
3) New processes applied to existing techniques
4) New processes applied to new techniques

Here, existing processes refer to those, which are successfully being used in the industries. New processes are those, which are either at a laboratory stage or comparatively new. Conventional processes like condition monitoring systems, asset life prediction, condition based monitoring, Information Technology (IT) asset management process, decision-making, etc and the unconventional processes like business processes, workflow processes, etc can be considered as existing processes. The processes like reliability analysis processes, intelligent maintenance decision systems and further modifications can be considered as new processes.

A majority of research work has been reported for existing techniques and existing processes whereas the new techniques and new process combination has been least reported. The other two categories come in between. Thus the direction of research in asset management is driven by industrial requirement of making the existing process better as well as it is self driven to search new avenues. As far as industrial requirements
are concerned, development of new or alternative techniques and processes plays an important role. The growing trend in asset management area is bi-directional, i.e. continuously evolving conventional management technologies along with development of new processes.

1.3 Current Directions

New processes are required to cater to the ever-increasing needs, and investigation is required for existing techniques to process them. In addition investigation regarding improvement or modifications of these processes will enhance their industrial applicability. Among the processes, asset management processes are generating extensive interest in diverse fields like reliability, system thinking, life-prediction, etc. These processes have different characteristics of process length of life, safety and risk mitigation, planning horizons, stakeholder involvement; etc [8]. An AM process is a set of linked activities and the sequence of these activities that are necessary for collectively realising asset management goals.

Existing techniques like Business Process Management (BPM) or Workflow techniques needs to be investigated in order to improve or to make modifications for the process to improve the industrial applications. BPM/Workflow techniques are creating interest in the fields like automation, IT, human co-ordination, etc. They have the following distinct characteristics [9]:

Ad-hoc workflows perform office processes, such as documentation, sales proposals, where there is no set of patterns for moving information among people. This type of workflows is not automated but instead controlled by humans.

Administrative workflows involve predictable processes with simple task coordination rules, such as routing an expense report. The ordering and coordination of tasks in this type of workflows can be automated.

Production workflows involve repetitive predictable business processes, such as loan applications or insurance claims. This type of workflows typically encompasses a complex information process involving access to multiple information systems. The ordering and coordination of tasks can be automated.
Samples from the research publications of AM were taken and analysed to find out the prevailing points:

- There are a number of processes, which have been used for development of asset management [7]
- The major research objective is to improve asset management and its processes by achieving uniform distribution of its various results [13]
- The research community has worked on different types of processes for a particular study objective. The overall domain for AM processes is very wide. AM decision making process has been a favoured choice [11]
- Data quality is a critical issue for effective AM. This data quality problems can result in severe negative consequences [17]
- Everyone is now faced with cost pressure in almost every aspect of the business putting a strain in profitability by continually challenging the fixed and variable elements of the operating environment [20]
- Deregulation and an increasing competition in markets urge suppliers to optimize the utilization of their equipment, focusing on technical and cost-effective aspects with regard to the AM [18]
- International Data Corporation found that companies practicing AM process lowered their annual costs by nearly 20% on average [4]

Many organisations around the world are devoted to AM-related research and development. Among them, Cooperative Research Centre for Integrated Engineering Asset Management (CIEAM) is one of the leading organisations. CIEAM is conducting research in five themes which are listed below. Each research programme focuses on key disciplines in the engineering asset management field.

**Models and Decision Systems**: - This program is to develop and implement reliability and maintenance models, business processes and procedures using a holistic approach to asset maintenance and management.

**Advanced Sensors**: - This programme is to develop sensor technologies for intelligent maintenance systems to detect and quantify machine degradation and structural damage.

**Intelligent Diagnostics and Remnant Life Prediction**: - This programme is to develop technologies to streamline and automate fault diagnosis and prognosis in intelligent
maintenance systems; activities that are central to accurate maintenance planning and scheduling.

**System Integration and IT**: This programme is to develop standards and tools for data exchange and integration between management information systems and asset based technical systems to provide strategic management decision tools in maintenance and operations.

**Strategic Human Dimensions**: This programme is to develop best practice approach to efficiently transfer and integrate research into asset management business practice.

Among these various themes of research programmes, this thesis belongs to the research programme of “Models and Decision Systems”. The models integrate business goals, production and maintenance functions, taking into account total life cycle costs including acquisition, operations, maintenance and opportunity costs, and the cost of down time risks. The goals of this research programme are:

- To significantly advance data analysis and reliability models business architecture and strategic decisions systems suitable for the industry sector in engineering asset management.
- To develop and test intelligent maintenance decision systems for asset management processes.
- To develop the applicable reliability analysis models based on advanced data models and failure analysis algorithms.

These integrated research outputs will accommodate the requirements of individual industries.

Thus, it is clear from the above discussion that the development of AM processes is a new arena for desired possibilities.

The ability of Workflow modelling (WFM) techniques to model any process provides an interest over AM processes to model them [21]. The research on application of WFM techniques to AM processes has not been carried out before. As AM processes and WFM techniques, both concepts didn’t combine in the past. Hence this research will try to combine these two concepts by modelling the AM processes using WFM techniques.
1.4 Motivation

The discussion in the Section 1.3 clearly focuses on two facts, i.e. that the development of new techniques and improvements in the process modelling for the AM processes are current requirements. The development of AM process as well as their modelling using the workflow modelling technique is a promising field and still needs investigation at length. These two factors are prime motivators of this present work. The industrial applicability of the process can further be enhanced if the processing method is optimized and the end user is provided with some predictive models, which can be used at shop floor. In this regard, it is a good proposition to study holistically, the development of asset management process and its modelling by using workflow models. In particular, following gaps were observed (details provided in Chapter 2 under Literature Review), which in turn generated opportunities for investigation into the process.

- In recent times, AM processes (particularly AM decision making processes) have been increasingly used in industry or organizations. The characteristics of AM decision making processes like the structure, the level, the procedure, the information flows, decision making criteria, and the incentive structure for formulating and executing decisions make it suitable choice as a process model [14, 19]. AM process model is yet to be explored with full potential. The applicability of such models is limited by problems associated with AM and its processes, which is still in the stage of infancy. In this case, attention may be given for developing the process model of these AM processes.

- There is little effort directed towards improvement in process performance in terms of good modelling skilfulness with respect to AM processes. In such a case, it would be imperative to study comprehensively the AM process in terms of effective modelling.

- There is hardly any information, available regarding modelling of AM processes by combining WFM with it. It has been reported that modelling with WFM technique enhances the process. Thus, an opportunity arises to investigate WFM techniques to apply it to AM processes.
It had been a problem for researchers to obtain optimal and economical performance of AM processes. Thus, extensive research is needed to obtain the optimal set of AM process parameters that provide best performance of AM process with WFM.

Thus, the present investigation is defined as follows:

“Modelling and Execution of Asset Management Processes via Workflow Automation”

Following the stated definition, present investigation has been carried out. The subsequent section provides the overview of the thesis.

1.5 Thesis Outline

The present thesis is divided into following 6 chapters:

Chapter 1: Introduction
The chapter discusses the present state of research in Asset management and describes the motivation behind the present work and also deals with model organization of this thesis.

Chapter 2: Literature Review and Problem Formulation
Critical review of previous works (pertaining to the present work) is described in this chapter. Potential areas of research in AM, AM processes, BPM, WFM techniques have also been identified. In turn, gaps, opportunities and the problem for present thesis are given along with objectives and scope.

Chapter 3: Methodology and Techniques
The fundamental aspects of ‘methodology and techniques’ for modelling the AM process used in the present research are presented in this chapter. The chapter also presents details about the YAWL modelling technique.
Chapter 4: Ameliorating the AM Process Definition
Description regarding the work carried out for improvement, assessment of an Asset management split decision making process is given in this chapter.

Chapter 5: AMSDM Process Modelling and Execution using YAWL
The chapter presents the development strategy of the AMSDM process model, how this process is modelled and executed using Yet another workflow language (YAWL) modelling technique.

Chapter 6: Conclusions and scope of Future Work
The chapter draws overall conclusions of the work and also throws light on possibilities where the work can further be extended.
CHAPTER 2

LITERATURE REVIEW AND PROBLEM FORMULATION

2.1 Overview

This chapter is divided into five sections. A detailed account of background and previous investigations of asset management processes is presented in the Section 2.2. The potential research areas, identified in the asset management processes from the gaps available in literature are presented in the Section 2.3. The Section 2.4 details about the background and review of investigations regarding the business process management. The gaps and opportunities regarding executable process models have been discussed in the Section 2.5. In Section 2.6, the problem formulation for the present work is presented, which covers problem definition, objectives.

2.2 Asset Management Processes

We are in a rapid period of change. Change is all around us, in our political system, in our economic system, in institutional relationships, in technology, in public attitudes and in customer’s expectations. We need not only to be a part of change; also be leading the change. Otherwise, we will be falling behind; and we will not have the support that we need for highway programs in future. Hence the genesis of the movement towards AM in the globe has been an understanding of the need for it.

Industry has recently put a strong emphasis on to the area of AM. In order for organisations to generate revenue they need to utilize assets in an effective and efficient way [24]. Often the success of an enterprise depends largely on its ability to utilise assets efficiently. Therefore, asset management has been regarded as an essential business process in many organisations. As we enter the new century, AM is attracting the resurgence of interest; it optimises the use of assets for the delivery of the organisation objectives [25].
2.2.1 Introduction
Due to the broadness of the term Asset Management (AM), there are varieties of definitions, and are often specific to the various organisations.

The Federal Highway Administration (FHWA) defines Asset Management as follows:

“Asset Management is a systematic approach of maintaining, upgrading and operating physical assets cost effectively. It combines engineering principles with sound business practices and economic theory, and it provide tools to facilitate a more organized, logical approach to decision making. Thus, asset management provides a framework for handling both short and long range planning (FHWA 1999) [15].”

However, there have been many other definitions that consider different aspects of the business strategies pertaining to AM and that also widen its scope beyond solely physical assets. Examples of this sort include the definition from the Transportation Association of Canada (TAC):

“Asset Management is a comprehensive business strategy employing people information and technology to effectively and efficiently allocate available funds amongst valued and competing asset needs (TAC 1999) [23].”

And the one from the American Public Works Association (APWA):

“Asset Management is a methodology to efficiently and equitably allocate resources amongst valid and competing goals and objectives (Danyo and Lemer 1998) [26].”

And the one from the organisation of Economic Cooperation and Development (OECD) emphasises the service to the public, which is the end customer of the road agencies and administrations:

“Asset Management is a systematic process of maintaining, upgrading and operating assets, combining engineering principles with sound business practice and economic rationale, and providing tools to facilitate a more organised and flexible approach to making the decisions necessary to achieve the public’s expectations (OECD 2000) [27].”
And the definition derived in terms of AM process was:

“Asset Management process is, the set of processes aligning the development, operation and maintenance of assets, so as to meet the desired requirements and objectives of the shareholders in the organisation (AASHTO 2005) [28].”

Another Definition in terms of AM process was:

“An Asset Management process is a set of linked (often interrelated) activities and the sequence of these activities that are necessary for collectively realising AM goals, normally within the context of organisational structure and resource constraints (AM Processes: Modelling, Evaluation and Integration 2007) [7].”

From the joint committee of AASHTO-AGC-ARTBA [34], Asset management is defined as the strategic approach to the optimal allocation of resources for the management, operation, preservation, and maintenance of infrastructure. The concept of AM combines engineering, economic principles, and sound business practices to support decision making at the network and project level [6].

Based upon all of the above definitions, asset management is defined as the comprehensive and systematic approach for maintaining and operating of assets to meet the desired requirements and objectives, and to make the decisions necessary in a more organised manner.

2.2.2 Asset Management Principles

In its most general sense, AM is a business approach designed to align the management of asset-related spending to corporate goals. Typically, utilities adopt an AM approach to either reducing spending; more effectively manage risks, or drive corporate objectives throughout an organization [30]. Simply, AM is a corporate strategy that seeks to balance performance, cost and risk in order to ensure the optimum utilisation of asset. Achieving this balance requires the alignment of corporate goals, management decisions, and technical decisions [5]. It also requires the corporate culture, business processes and information systems capable of making decisions based on asset-level data. AM is
ambitious in scope and requires supporting metrics, organisational design, processes, information systems and corporate culture. Successful implementation can be quite disruptive and requires the involvement and support of top management, sufficient resources and effective change-management skills [5].

International Data Corporation found that companies practicing AM lowered their annual costs by nearly 20% on average [4]. With AM, they can track all the costs associated with an asset including initial price, depreciated value, service costs, add-on equipment etc.

There are five key components to any comprehensive asset management system:

- An asset inventory
- Methods of assessing current conditions and performance
- A process to determine and evaluate the future system needs
- Tools to evaluate and select appropriate strategies to address current and future needs
- Methods to evaluate the effectiveness of each strategy

A robust AM structure is supported by three pillars of competency: management, engineering and information. Building these competencies is daunting when reviewed in isolation. Far more difficult is developing cross functional expertise so that management, engineering and information skills can be addressed in a mutually supporting manner [5]. Management means strategy, organisational design, performance management, resource planning, decision analysis and financial risk. On the other hand, engineering covers the planning, operations, maintenance, reliability, protection and technical risk. The information entails system architecture, business intelligence, knowledge management, integration and asset registry. In this way three pillars of competency support asset management [5].

The world is constantly changing to have effective AM solutions. AM is based on a robust and accurate repository [4]. For this there is need to automate and regular input of reliable physical data, as well as newly acquired asset data. There is also a need to regular input of change data coming from consolidated service desk. The integration of reliable discovery tools into the process is critical to reconcile the physical view and the logical view within the AM repository. AM is spreading across many disciplines such as
economic, engineering, IT and different sectors within an organisation [8]. Truly, AM and quality are two sides of the same coin. The quality will be attained based on the leadership, strategic planning, information systems, process improvements, management and development of the human resources, focus on customers, and bottom line business results. These same “qualities” require AM [29].

**Benefits of Asset Management**

- Organisations waste money and slow down their business with inefficiencies that can be alleviated by effective asset management [4].
- Through Asset management we can monitor, define, optimise and track their business assets and processes by automating, which was previously manual [4].
- Enhanced objective information made available for better decision making [32].
- With asset management, track all the costs associated with an asset including initial price, depreciated value, service costs, add-on equipment and upgrades [36].
- The main benefits of AM are in terms of finance and time [4].
- Effective asset management is critical for successful customer service, regulatory compliance and the business viability [4].

From the “Asset Management Primer written by the U.S. Department of Transportation 1999” AASHTO and FHWA [15] are convinced that Asset Management can enhance business performance. An AM philosophy focuses on the benefits of investment, as well as its costs, and takes a comprehensive view of the entire portfolio of transportation resources. Objective, fact-based tools and techniques are systematically applied to determine how best to deploy available resources in order to achieve system-wide agency goals. AM is an improved way of doing business that responds to an environment of increasing system demands, aging infrastructure, and limited resources. AM also provides the ability to show how, when, and why resources were committed. Transportation officials are being held increasingly more accountable by their customers—the American public. The public demands a consistently high return on the portfolio of transportation assets, which, of course, represents a collection of public resources.


2.2.3 Asset Management Processes

Many organisations do not map out their AM processes using modelling techniques [5]. With AM playing a increasingly important role in organisations there is a need to be able to model AM processes in the most beneficial and efficient manner, to enable better decision making and the improvement of the processes being modelled. Asset Management has only recently gained sufficient attention in terms of realising the need to improve the processes within it. Therefore, the literature of asset management process improvement is quite sparse and describes only specific parts of asset management, whilst not fully conceptualising the broader asset management scope [7].

The AM Processes can be described by two different models namely the Functional model and the Business model [39]. While the functional model as proposed by CIGRE [41] is based on the categorisation of functions that take place within the organization by accountability. The business model, based on the organisational structure of the company was used for the development of the decision support.

Asset Management Decision Process

Asset management decision processes are the individual decisions that need be made in every level of decision making. The main levels of decision making are strategic, network, and project level. The strategic decision-making level is the broadest and most comprehensive. It pertains to strategic decisions concerning all types of assets and systems. The strategic level of decision making is concerned with generic and strategic resource allocation and utilisation decisions within the man-made environment.

Network level decision-making pertains to determining the overall agency-wide maintenance, rehabilitation and construction strategies and works programs. This decision level considers system-wide decisions but the scope is narrower than the strategic level.

The program decision-making level is concerned with the overall, network-wide programming of actions and allocations. It’s a level involving policy decisions and the aim is the system-wide optimization of funds allocated to rehabilitation, maintenance or new construction of infrastructure assets.
The *project selection level* is concerned with decisions on funding for projects or groups of projects. This level generates decisions at a higher level of aggregation than the project level but it requires more detailed information than the two previous ones. It serves as a link between the network level and the subsequent project level of analysis.

The *project level* of decision making and analysis pertains to the specific, mode-wise, asset-wise and geographically determined projects. It addresses the overall work plan that needs to be developed at this level, for the specifically selected project in order to meet the agencies’ performance measures. It is also called field level or operational level and refers to how the actual work is going to be done.

Although all the above hierarchical levels of decision making are clearly defined, there exists significant overlapping among what the management needs to do at every level. The identification of the actual data needed in every decision making level is a very challenging task. This is partly due to that significant overlapping between the various decision levels. Another important reason is also the lack of relevant research initiatives in this field up to date. Decision processes can therefore be concerned with budget allocations, network optimisation, work’s programming, and selection of alternative implementation methods.

Furthermore, different levels of decision making have different focus on the network. Higher levels are mostly concerned with overall budget allocations and system utilisation, while lower levels tend to focus more on the administration, funding and engineering of specific functions and processes. Also, people who make decisions often have different backgrounds and interests. As a result, the decisions at each level are different in scope and data aggregation level and so should be the corresponding detail and quantity of the collected data. This concept is illustrated in the Figure 2.1.
Figure 2.1: Relation between the different decision making levels and the corresponding detail and amount of needed data [14]

2.2.4 Data Collection for the Asset Management Process

Data collection, data management and data integration are essential parts of the AM framework that are critical to its success [14]. Timely and accurate data lead to information and form the basis for effective and efficient decision making. Data collection is very much dependant on the intended use of the data. It is obvious that the level of detail and the depth needed for the collected data varies according to the hierarchical level of the decisions that need to be made. Although all decision-making levels are undisputedly part of the overall AM process, data collection requirements have to specifically consider how the collected information is going to be used at the various management decision levels [17]. Data needs for supporting strategic, network, or project level are significantly different in terms of degree of detail and required accuracy. Broadly speaking the data collection requirements can be categorized in the following three groups [42]:

- **Location**: actual location of the asset as denoted using a linear referencing system or GPR coordinates.
Physical attributes: description of the considered assets that can include: material type, size, length, etc.

Condition: condition assessment data can be different from one asset category to another according to the set performance criteria. The data can be qualitative and generic (e.g., Good, Bad, etc) or detailed and/or quantitative in accordance to established practices and standards (e.g., Pavement Condition Index, bridge health indices, etc).

Data Collection Methods

Infrastructure data collection has been an ongoing process since the 1960’s. In the last decades the various methods and technologies used have shown a trend towards automation and computerisation.

Methods used for the collection of asset management data include: (1) manual, (2) automated, (3) semi-automated, and (4) remote collection. Regardless of the method used, the existence of an effective Quality Control and Quality Assurance (QC/QA) program is vital for the success and reliability of the collection. A brief description of each method is presented following [38]:

Manual collection: The method employs two or more data collectors and a distance measuring device. The collected data are documented either with pen and paper or in most recent cases with hand-held computers.

Automated collection: The method involves the use of a multipurpose vehicle which is equipped with a distance measuring device, digital video cameras a gyroscope, laser sensors, computer hardware, CMMS systems, SCADA systems and potentially GPS antennas in order to capture, store, and process the collected data.

Semi-automated collection: This method involves similar equipment as the completely automated method but with a lesser degree of automation. It is very popular within transportation agencies and yields comprehensive and accurate data collection when properly implemented.
Remote collection: This last method pertains to the use of satellite imagery and remote sensing applications. These methods involve high resolution images acquired through satellites or other types of images and scans obtained by remote sensing technologies (lasers, aerial photos, aerial GPR, etc).

Data Characteristics and Properties

The research of various sources on how agencies worldwide deal with AM Processes has brought to light particular attributes and characteristics that the collected data should possess in order to be useful for this purpose. Regardless of the particular type or category that the collected data fall into, it is of paramount importance that when incorporated in a database they exhibit the following characteristics [43]:

- **Integrity:** whenever two data elements represent the same piece of information, they should be equal;
- **Accuracy:** the data values represent as closely as possible the considered piece of information;
- **Validity:** the given data values are correct in terms of their possible and potential ranges of values; and
- **Security:** sensitive, confidential and important data are protected by restricting access to them and by properly ensuring systematic and frequent "backing-up" in other storage media.

In addition, the Western European Road Directors (WERD) [44] highlighted the importance of the following criteria when selecting data required by an agency/organisation:

- **Relevance:** every data item collected and stored should support an explicitly defined decision need,
- **Appropriateness:** the amount of collected and stored data and the frequency of their update should be based on the needs and resources of the agency/organisation,
- **Reliability:** the data should exhibit the required accuracy, spatial coverage, completeness and currency,
- **Affordability**: the collected data are in accordance with the agencies financial and staff resources.

According to the same source [44], agencies planning to engage in data collection should take into account and determine the following parameters:

- The specification of the data to be collected,
- Their frequency of collection,
- The accuracy and quality that the data should exhibit,
- Their completeness and currency.

As a general recommendation it is noted that the accuracy, quality and currency of the data should be decided based on the cost of the data collection and the value and benefit associated with the data in question. “Data should only be collected if the benefits that they provide outweigh the cost of their collection and maintenance” [44].

According to (FHWA 2001) [45] data integration alternatives include two main approaches: (1) a fused database and (2) many interoperable databases. In the first case the integration strategy leads to the creation of one database that contains all integrated data; in the second case existing or newly created databases are linked together and the integration of the data is achieved with the use of queries that provide a view of the linked data.

The choice among the two integration strategies depends on many factors and is clearly a judgment call for the agency officials. The factors to be considered include:

- Intended use of the integrated data (by whom and for what purpose),
- Characteristics of the already existing databases/information systems,
- Type and volume of the data that need to be integrated,
- Currently available information technology,
- Level of staff and resource allocation that will be dedicated to the process,
- Structure of the agency/organisation itself (business units and their roles, data needs, people and information systems).
As location is an important property of all transportation assets, it has served in many cases as the common platform used for data integration. For example, various state DOTs have used GIS and other geospatial tools for data integration [46]. GIS software and related functionalities can alternatively be incorporated in the databases as external software that enhances the analytical and reporting capabilities of the system [45].

Another aid for the integration and interoperability of databases is the use of commonly accepted data definitions and consistent formats across systems. A standard data dictionary or global standard for data definition, representation, storage and communication could be of vital help to the effort of data integration, regardless of the integration strategy implemented. However, there have been many challenges identified by agencies that have tried to develop and implement data standards and that have attempted to convert existing, legacy data to these new standards. These challenges include agreeing on suitable data formats, models and protocols when the existing databases present extreme diversity; achieving support from the agency staff and getting people to conform to the new standards; and reducing as much as possible the effort and resources needed in order to develop and implement the standards [47].

**Decision Processes and Data Collection**

Independent of the data integration strategy chosen and level of integration achieved, there are many dimensions inherent in the analytical and decision making processes concerning various assets that need be taken into account.

Decision processes can be either:

- At an operational level (e.g. how to repair an Engine) or
- At a more generalised strategic level (e.g. how often to remodel the engine).

Large and diverse amounts of data are needed in order to fully support the decision processes in all their possible dimensions and in all levels of decision making within the agencies. In addition, the resulting system’s complexity is big enough to intimidate even carefully designed strategies and high levels of data integration that are chosen to be implemented [45]. A carefully conceptualised thought process of rationalising which data are needed to support which type or level of decision processes needs to be developed.
Data should be collected according to their intended use and therefore data collection should be carefully planned according to these needs.

Decisions made at the different levels of Asset Management are heterogeneous and the supporting data needs are bound to be quite different. To systematically approach and identify the data needed to support Asset Management decision processes, it is necessary to first define the level of decision making these processes support. The data needed to support the various decisions at any of the various levels are different. Higher levels require more generalised information while lower ones tend to need more detailed and specific data [48].

2.3 Gaps and opportunities- Modelling and Execution of Asset Management Processes

Based upon the available literature review some gaps, i.e. the scope of research work are identified and described as follows:

1. In recent times, AM applications are increasingly used in industry. The continuation of research in the area of AM indicates that the area is still in the stage of infancy.

2. Many investigators have advocated the practical and economical benefits of AM. However the technical difficulties associated with AM resulted in one or the other defect.

3. As AM frameworks and methodologies are process-based, there is a need to model AM processes effectively to enable understandability between different industries and personnel from different backgrounds.

4. Although organisations adopt asset management principles and carry out AM initiatives, most do not map out or model their Asset Management processes. There is currently no body of knowledge that aids in modelling asset management processes in a methodical and suitable approach.
5. Which type of data is required for modelling the AM processes and how much detail this data should be?

6. Not all of the asset management process characteristics are suitable for applying executable process models.

The above mentioned gaps generate opportunities to carry out research in the development of AM process model.

1. The modelling in the development of AM processes is an area wide open to be investigated.

2. Modelling and execution of AM processes has great potential. As organisations become more complex and different people from different sectors collaborate on projects and decision making, however, is very much in its infancy stage of development.

3. Modelling of AM processes has established its utility for the exploitation of AM. Thus, modelling can be a potential area for the maturation of AM processes.

4. As there is no substantial body of knowledge exists in modelling the AM processes, suitable modelling techniques have to be developed for every specific case.

5. Need to do the thorough analysis of the data part of AM processes for modelling it. In addition to that, what type of data is expected and as well as how much detailed level of data is required?

6. Need to identify the particular AM process characteristics that are suitable for applying to executable modelling techniques.
2.4 Business Process Management and Workflow Technology

Business process management (BPM) is a field of management focused on aligning organizations with the wants and needs of clients. It is a holistic management approach that promotes business effectiveness and efficiency while striving for innovation, flexibility, and integration with technology. BPM attempts to improve processes continuously [49]. It could therefore be described as a “process optimization process”.

2.4.1 Introduction

“A business process is a collection of activities or tasks designed to produce a specific output, and the business process model is thus a specific ordering of work activities or tasks across time and place, with a beginning, an end, and clearly defined inputs and outputs (H. Smith and P. Fingar, 2003)[49].”

“A business process is the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result. Successful systems start with an understanding of the business processes of an organisation (K. Vergidis, A. Tiwari, B. Majeed, and R. Roy, 2007) [50].”

BPM Life Cycle

BPM activities can be grouped into five categories: design, modelling, execution, monitoring, and optimization [52].

![Business Process Management Life-Cycles](image)

**Figure 2.2 Business Process Management Life-Cycles [52]**

*Design:* Process design encompasses the identification of the existing process. Area of focus include representation of process flow, the actors within it, alerts, notifications,
escalations, standard operating procedures, service level agreements, and task hand over mechanisms.

*Modelling*: Modelling takes the theoretical design and introduces combination of variables (e.g. changes in rent or materials cost, which determine how the process might operate under different circumstances).

*Execution*: One of the ways to automate process is to develop an application that executes the required steps of the process; however, in practice, these applications rarely execute all the steps of the process accurately or completely.

*Monitoring*: Monitoring, encompasses the tracking of individual processes. So that information on their state can be easily seen, and statistics on the performance of one or more processes can be provided. The degree of monitoring depends on what information the business wants to evaluate and analyse and how business wants it to be monitored, in real time, near real time or ad-hoc.

*Optimisation*: Process optimisation includes retrieving process performance information from modelling or monitoring phase; identifying the potential or actual bottlenecks and the potential opportunities for cost savings or other improvements; and then, applying those enhancements in the design of the process. Overall, this creates greater business value.

### 2.4.2 Process Modelling and Execution

Process models play an important role in the development of organisational business processes. Process models are processes of the same nature that are classified together into a model. Thus a process model is a description of the process at the type level. Since the process model is at the type level, a process is an instantiation of it. The process model is roughly an anticipation of what the process will look like. The goals of the process model are to track the process to know what actually happens and takes the point of view that looks at the way a process has been performed and determines the improvements to make it perform more effectively.
**Business Process Modelling**

Among the various process models, we deal with the business process models. Recent years have seen an increasing interest in methodologies, techniques and tools to support the design of business process under the banner of business process modelling. It is the subject for many undergraduate and postgraduate courses, as well as main topic of investigation in several research centres around the world.

There are several key factors why business process modelling has become an effective and popular tool in industry today. Most important of these is the ability of modelling to integrate systems, people, data flow etc, into a logical framework, which thereafter can be used for analysis and improvement of the current process system. According to [53, 54], a business process can be seen as set of partially ordered activities intended to reach a goal, and the business process model is defined as the generic description of the class of the instance of the business process. Business process model describes how the instance to the business process is to be carried out. A conceptual BPM is independent of particular IT or organisational environment and an executable BPM is customized to a particular environment, it may be instantiated to carry out the specific instance of the business process [118].

**Benefits of Business Process Modelling** [55, 56, and 57]:

- BPM assists an enterprise to achieve strategic objectives by providing methodology and tools to develop integrated business process models.

- BPM plays an important role in the business process management discipline. With advances in technology from large platform vendors, the vision of BPM becoming fully executable (and capable of simulations and round-trip engineering) is coming closer to reality every day.

- Modeling languages like Business Process Modeling Notation (BPMN), Business Process Execution Language (BPEL), Unified Modeling Language (UML), and Web Services Choreography Description Language (WS-CDL) and other
technologies related to business process modeling include model-driven architecture and service-oriented architecture.

- Business Process Modeling has always been a key aspect of business process reengineering (BPR) and continuous improvement approaches.

- Formalize existing process and spot needed improvements.

- Facilitate automated, efficient process flow.

- Increase productivity and decrease head count.

**Executable Process Model**

A model is an abstraction, one that should accurately reflect an aspect of whatever is building. But will it work? To determine so, one should prove the model with code. Hence, a coarse grained model visualises the overall dependencies between the activities. Usually it is defined in several steps during the development process.

“Finally, a very precise model, covering all necessary aspects, can be executed directly by help of a workflow engine. Hence, such a model is called executable model (Axel Martens 2005) [58].”

And the definition in terms of Business process is

“An Executable Business Process is a business process whose tasks can be orchestrated with a software platform, using a combination of automated systems tasks (e.g. looking up customer information in a relational database, sending an order to a partner through a B2B gateway) and user tasks (e.g. users approving a timesheet, users starting a process from a form, users managing a process, etc.) (GLINTECH 2006) [59].”

Executable modelling allows model to be tested as prototypes. Executable model transforms the requirement process to focus on the user needs and not on technical design considerations. It also enables a wider review circle as more people can participate. It makes review active (testing) rather than passive (reading documents). While many
benefits can be gleaned for the analytical stage and the chances of sub-optimisation are reduced, most managers with guns pointed to their head for savings tend to go for the executable process that can be measured, tweaked and optimised for more savings.

**Executable Model Principles**

Executable model treats the human and automated tasks or activities equally; also it is primarily for capturing the flow of control within a process, and not for optimisation. Executable model purpose is to analyse the process of what it is supposed to do, also synthesise the implementation (how it does) as the explicit coordination of other processes, and also it is iterative, can apply until left only with the indivisible tasks [60]. To be effective, an executable model must be capable of interacting with the user to perform four basic functions [61]:

- Execute a process model representing a work process for a particular project.
- Check the validity of user requests to perform operations defined in the process model and invokes tools to carry out these operations.
- Suggest the next logical operation for the user to perform, based on knowledge of the current state of work in the total project.
- Record status and log historical events that are of interest when performed points are reached during execution of process model.

An executable process model must provide an environment in which a process model can appropriately balance how much the process guides the user and how much the user guides the process. The Executable process model focuses on the two modes of the control spectrum [61]. They are:

1) *User initiated and process guided*, where an initiative for selecting operations to be performed with the user, where as the executable model provides guidance by informing the user as to whether prerequisites specified in the process model for the selected operation have been met.
2) **Process initiated and user guided**, where operations to be performed are suggested by the executable model based upon the interpretation of the process model and the state of the process at that time. The user provides the guidance by deciding whether or not to execute the suggested operations.

Both in these types of spectrum the executable model must know the state of work in progress and take the initiative to guide the user to the next logical activity in the process model within the context of the state of the total project. Also the project manager and developer must be allowed to take the initiative to guide or control the executable process based on team knowledge of factors outside the process model. The executable model must exercise the protection against capricious user navigation, intentional or inadvertent, through the defined process [61].

### 2.4.3 Workflow Technology

This new world is characterised by an increasing number of business processes subject to continuous change. Organisations are challenged to bring ideas and concepts to products and services in an ever increasing pace. Companies distributed by space, time and capabilities come together to deliver products and solutions for which there is any need in the global market place. The trends for virtual corporations and increasing global networking of economies are real and will accelerate. As a result, more and more workflow processes are subject to continuous change. At the moment, there are many workflow products commercially available and many organisations are introducing workflow technology to support the business process. Workflow systems are currently the leading technology for supporting the business process. This technology manages the execution of tasks involved in a business activity, the scheduling of resources and the control of the flow associated information required by performers to execute the tasks.

“**Workflow model is a description of business process in detail that can be directly executed by workflow management systems** (Martin Vasko and Schahram Dustdar 2006).” Where as the “**workflow management system is defined as the system that completely defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic** (D.Georgakopoulos 1995) [9].”
And the one from the workflow management coalition (WFMC)

“The computerised facilitation or automation of business process, in whole or part is defined as workflow (WFMC 1995) [62].”

And the another definition for the workflow model in terms of the Business process from the WFMC is

“The automation of the business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to set of procedural rules (J, Tick 2007)[76].”

And the another definition in terms of business related workflows

“A task defines some work to be done by a person, by a software system or by both of them. Specification of workflow involves describing those aspects of its component tasks that are relevant to control and coordinate their execution, as well as relations between the tasks themselves (D.Georgakopoulos 1995) [9].”

**Benefits of Workflows** [9, 65, 69]:

- Workflow, which is often used casually for referring to a business process and specification of a process.
- It is used for automating the process.
- It is used for the software that simply supports the coordination and collaboration of people that implements a process.
- Workflow is the mechanism by which you can implement business reengineering practices.

**2.4.4 Languages, Systems, and Tools**

The different approaches of emerging workflow modelling languages are manifold. Today, there exist many notations for workflow modelling with various specialisations on different domains. There are various workflow modelling languages, among these the
important languages are business process executable language (BPEL), Business process modelling notation (BPMN), Yet another workflow language (YAWL), XML process definition language (XPDL). These languages provide a formal model for expressing executable processes that addresses all aspects of business processes. These languages cover a broad spectrum of workflow design notations.

**Business Process Execution Language (BPEL)**

BPEL is an “execution language” designed to provide a definition of web services orchestration. It defines only the executable aspects of a process, when that process is dealing exclusively with web services and XML data. BPEL arouses from a consortium of industry leading companies and therefore has strong vendor background and offers many implementations. BPEL is currently standardised by Organisation for the Advancement of Structured Information Standards (OASIS) lead by consortium of industry heavy weights such as Microsoft, IBM, SAP, Siebel, and Oracle [69]. With the popularity and advent of BPML and the growing success of BPMI.org and the open BPMS movement led by JBoss and Intalio Inc., IBM and Microsoft decided to combine these languages into a new language, BPEL4WS. In April 2003 BEA systems, IBM, Microsoft, SAP and Siebel Systems submitted BPEL4WS 1.1 to OASIS for standardisation via the Web services BPEL technical committee. BPEL is a business execution language to enrich and standardise the commonly used protocols, especially for long running business processes [78]. BPEL, in its essence, is an extension of imperative programming languages with constructs specific to the Business process management domain. BPEL process definition specifies the technical details of a workflow that offers a complex Web service build from a set of elementary Web services. The most important concepts of BPEL are variables, partnerLinkTypes, basic and structured activities, as well as handlers. Variables store process data and messages that are exchanged with Web Services [77]. PartnerLinkTypes define the mutual required port types of a message exchange by declaring which partner acts according to which role defined in a partner link. Basic activities specify the operations which are performed in a process. Structures activities are utilised for the definition of control flow. Handlers can be defined in order to respond to the occurrence of a fault, an event, or if a compensation has been triggered. BPEL is a block structured programming language, allowing recursive blocks but restricting definitions and declarations at top level.
In general BPEL is applied in two ways [69, 77]:

- The business protocol can define the abstract processes, which approaches data handling in a way that reflects the requirements of the workflow to execute.

- Executable business processes on the other side determine the nature and sequence of Web Service interactions.

This basic concept model of distinction between abstract and executable processes makes it possible to export and import public aspects embodied in business protocols as processes. BPEL does not define the graphical diagram, human oriented processes, sub processes, and many other aspects of a modern business process: it simply was never defined to catty the business process diagram from design tool to design tool.

**Benefits of BPEL** [79]

- BPEL is an excellent way of implementing trading partner integration using the Internet. BPEL provides the ability for a company to quickly adapt or create new business processes, both internally and to its trading partner community.

- Web service is a great technology without a way for business to use it. For businesses, having access to Web service-enabled applications connected via BPEL means significant increases in efficiency and effectiveness.

- A Business Process using BPEL can compose multiple Web services, effectively creating a completely new business application with its own public interface to end users (internal or external).

**Limitations of BPEL** [81]

- BPEL still performs orchestration from the perspective of a single service. This is not adequate for maturing SOA implementations requiring choreography of multiple services.
BPEL can't invoke non-web service APIs such as POJOs (Plain Old Java Objects), EJBs, etc.

Orchestration defines an executable series of actions, but it does so from the perspective of a single process; however orchestration has a limited view of what the business logic is doing within an application, but lacks a big picture view of the total system.

**Business Process Modelling Notation (BPMN)**

The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the process, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor these processes [69]. Thus, BPMN closes the gap between process design and process implementation. Another design goal of BPMN is its compatibility to XML-based workflow languages like BPEL. Especially the visualisation of processes designed with these notations is important. The nature of executable languages such as BPEL renders them less suited for direct use by humans to design, manage, and monitor the business process that are enacted by process aware information systems. The BPMN was first released in May 2004. It has been intention of the BPMN designers to develop modelling technique that supports

(a) Typical process modelling activities for both business and technical analysts
(b) The straightforward mapping to executable workflow specifications in BPEL

BPMN is conceived of as block structured programming language. Recursive block structure plays a significant role in scoping issues that are relevant for declarations, definitions and process execution. BPMN focuses on issues important in defining web services [78]:

- Activity types specifically for message interchange, event handling, compensation delay.
Attributes to support instance correlation, extraction of parts of messages, locating service instances.

Support for transactions, utilizing the block structure context, exception handling and compensation.

The intent of BPMN is to standardise the business process design notation. Because of the complex domain of business process design, BPMN covers a variety of different modelling techniques and allows creation of end-to-end business processes. The structural elements of BPMN will allow the viewer to easily differentiate between diverse sections of the diagram. The basic structure types of processes are the following [69]:

*Private (internal) business process:* - Private business processes are those internal to a specific organisation and concern the workflow definitions in general.

*Abstract (public) business process:* - Abstract business processes define the interaction between a private business process and other processes or participants.

*Collaboration (Global) business process:* - Collaboration business process describes the interactions between two or more business entities.

**Limitations of BPMN** [69, 77, 78]

- BPMN does not provide a complete technical model. Specifically, it is not possible to describe the data manipulated by the process, the data transformations, the KPI to monitor for each activity, etc.

- BPMN only specifies a notation; it does not provide a wire (interchange) format. Therefore, it is not possible to export a BPMN diagram from a tool and import it into another one in a standardised manner.

**XML Process Definition language (XPDL)**

XPDL is a format standardised by the workflow management coalition (WFMC) to interchange business process definitions between different workflow products, i.e. between different modelling tools and management suites [78]. XPDL is designed to exchange the process definition, both the graphics and the semantics of the workflow.
business process. XPDL defines the XML schema for specifying the declarative part of workflow process. XPDL is conceived of as a graph structured language with additional concepts to handle blocks. Scoping issues are relevant at the package and process levels. Process definitions cannot be nested. Routing is handled by specification of transitions between activities. The activities in a process can be thought of as the nodes of a directed graph, with the transitions being the edges. Conditions associated with the transitions determine at execution time which activity should be executed next.

XPDL is extensible so that it allows each different tool to store implementation specific information within the XPDL, and have those values preserved even when manipulated by tools that do not understand those extensions. This is the only way to provide for a "round trip" through multiple tool and still be able to return to the original tool with complete fidelity. XPDL focuses on issues relevant to the distribution of work [78]:

- Activity attribute specifies the resources required to perform an activity. This is an expression, evaluated at execution time, which determines the resource required.
- Activity attribute specifies the applications required to implement an activity.

These concepts together support the notion of a resource, in conjunction with an application, performing the activity.

XPDL is the most widely deployed process definition language and XML-based BPM standard, leveraged by broad spectrum of software applications ranging from ERP, call centre and CRM, BI and BAM, process modelling and simulation, enterprise content management, as well as of course several of the leading workflow and BPM Suites. It has become the de facto global standard for workflow and BPM, and a common requirement of RFPs, and has been the foundation for mission critical systems and infrastructure, such as the provisioning systems at Telecom Italia.

**Limitations of XPDL** [82, 83]

- XPDL will not integrate web services in to higher level architectures or business processes.
The XPDL plug-in discards Node Graphics information elements when importing. Instead, the plug-in flags the newly created diagram as 'needing layout' and the diagram are automatically laid out before it is first displayed.

**Yet Another Workflow Language (YAWL)**

YAWL is resulted from the formal specification of workflows. This approach aims at the formal specification of different kinds of dependencies in workflow languages [84]. YAWL is based on the well known workflow patterns initiative and provides comprehensive support for these patterns. YAWL is inspired by the high-level Petri nets, but extended with some additional features to meet the requirements, defined by the workflow patterns, especially the models defining the multiple instances, advanced synchronisation and cancellation [69]. The definition of workflow specification in YAWL is a set of extended workflow nets, which have the tree like structure. This language is an imperative language so that complex long lasting process can be checked before deployment. In YAWL the process performance can be improved through the use of maps and visualising work items [85].

YAWL is implemented in a system consisting of YAWL designer and the YAWL engine, realised as the server side web-module. The engine verifies and registers the tasks or activities, which will be stored in the YAWL repository. This repository manages a collection of executable workflow specifications.

The YAWL consists of YAWL services and is explained as follows [69]:

**YAWL work list handler**: This service assigns work to the users, with this users can accept the work items and signal their completion.

**YAWL web service broker**: This service acts as the interface between the YAWL engine and other web services.

**YAWL interop broker**: This service provides the possibility to interconnect different workflow engines.

**Custom YAWL service**: The custom YAWL service can be of any kind; this mechanism makes it possible for external services to communicate through a “gateway” with the YAWL engine.
YAWL supports the three main perspectives [86]:

**Control-Flow Perspective:** This perspective captures aspects related to control-flow dependencies between various tasks. The three features offered by YAWL not present in most workflow languages (1) OR-join task (2) Multiple instances of a task (3) The “remove tokens” task (i.e. cancellation of a region).

**Data Perspective:** This perspective deals with passing of information and scoping of variables, etc. YAWL is one of the few languages that completely rely on XML-based standards like XPath and XQuery.

**Resource Perspective:** This perspective deals with the resource to task allocation, delegation, etc. YAWL engine interacts with its environment by means of a collection of “YAWL services”, which are responsible for handling the operational and resource perspective of workflow specifications, as well as supporting communication between different YAWL engines.

**Benefits of YAWL** [80, 83]

- YAWL is inspired by the high-level Petri nets, but extended with some additional features to meet the requirements, defined by the workflow patterns, especially the models defining the multiple instances, advanced synchronization and cancellation.

- YAWL provides strong support for the workflow patterns, the stronger patterns support means that it is easier to specify complex workflows.

- YAWL offers comprehensive support for dynamic workflows, where workflows need to change after their deployment and for handling unexpected exceptions.
2.5 Gaps and Opportunities- Application of Workflow Technology to AM Processes

Literature reviewed in the previous section reveals that substantial amounts of work in workflow models have been carried out. However the applicability of WFM for AM processes still has a long way to go. The thorough scrutiny of the work on WFM leads to the identification of some gaps. These gaps present opportunities to carry out research in the workflow modelling for the AM processes.

1. No literature exists that tries to apply BPM techniques into AM processes, little effort has been made in applying BPM techniques for the AM processes beyond the use of basic flowcharts.

2. As different techniques or languages exist in WFM, not all of the WFM techniques can be beneficial for mapping out the AM processes.

3. Little research is available into how the analysed WFM techniques can be used in qualitative decision making process.

The above mentioned gaps generate opportunities to carry out research in the Workflow modelling area:

1. There is a need to investigate whether the use of WFM techniques can be beneficial in mapping out AM processes. However, the benefits of applying WFM techniques to the other processes prove that it improves efficiency, better process control, improved customer service, flexibility, etc.

2. Needs to focus on reviewing and identifying the existing WFM techniques and looking into whether it is opportune to apply them when modelling the AM processes. This helps from the workflow point of view that it has been applying to new broad discipline of AM, which contributes towards new knowledge.

Based on the identified gaps and opportunities (section 2.3 and 2.5), the problem for the present investigation has been formulated. The following section presents the problem formulation of the problem.
2.6 Problem Formulation

The problem considered for the present investigation is formulated in the following steps:

*Problem Definition*

There is currently very little literature documenting a systematic approach to modelling the characteristics of asset management processes. In order to achieve a workflow model for AM process commonalities and differences between different AM processes need to be identified.

Little effort has been made in applying workflow modelling techniques to the AM processes. There is a need to investigate whether the use of workflow modelling techniques can be beneficial in mapping out AM processes.

Although organizations have placed a large emphasis on collecting and integrating data, little effort has been placed on organising the data i.e. what data is utilized and also up to which decision level.

**Objectives**

1. To review and identify the workflow modelling languages and tools that will be suitable for applying to the AM processes in order to model them.

2. To identify the characteristics of asset management processes in order to model it with workflow modelling languages and tools.

3. To analyse the data that is required for the AM decision making process, and need to model it with the workflow modelling language.

4. To develop an intermediate process between the AM Split decision making process and the workflow model which needs to be developed.

5. To apply the workflow modelling language for the AM Split decision making process in order to fill the gap in modelling the AM processes.
Significant of the Outcome

The overall outcome of this research is to obtain a workflow model for the AM decision making process. This outcome will result in optimising the efficiency of the process, gain better process control, improve flexibility of the process, coordinate and collaborate between people who implement the process.
3.1 Overview

This chapter discusses about the methodology and techniques related to the modelling of the AM decision making process. The first section details the development of the asset management split decision making (AMSDM) process and its characteristics. The subsequent section covers the techniques and its significance for modelling the AMSDM process. The third section covers the overall summary for modelling the asset management decision making process.

3.2 Methodology and Approach

To ensure that AM decisions are made efficiently and on a scientific basis, a practical AM decision making process is necessary. Such a decision making process is defined as a set of linked (often interrelated) activities and the sequence of these activities that are necessary for making optimal AM decisions, normally within the context of an organisational structure and resource constraints.

**AM Decision Making Process**

Decision making process is based on two characteristics [94]:

- “Decision making is the study of identifying and choosing alternatives based on the values and preferences of the decision maker”. Making a decision implies that there are alternative choices to be considered, and in such a case we want not only to identify as many of these alternatives as possible but to choose the one that (1) has the highest probability of success or effectiveness and (2) best fits with our goals, desires, lifestyle, values, and so on.

- “Decision making is the process of sufficiently reducing uncertainty and doubt about alternatives to allow a reasonable choice to be made from among them”.

56
This definition stresses the information-gathering function of decision making. It should be noted here that uncertainty is reduced rather than eliminated. Very few decisions are made with absolute certainty because complete knowledge about all the alternatives is seldom possible. Thus, every decision involves a certain amount of risk.

In AM, some decisions need to be made over a long term, such as annually, while others within a much shorter period, such as within hours. Based on this time scale, AM decisions are classified into the following types [14]:

**AM Strategic decisions:** - These types of decisions include defining AM objectives based on business objectives, and developing long term strategic plans.

**AM Technical decisions:** - These types of decisions includes developing AM plans, based on overall strategic plans, to determining major asset preventive maintenance and upgrading activities, as well as operational authorities.

**AM Implementation decisions:** - These types of decisions includes scheduling O&M activities, workforce allocation, money expenditure and material delivery timetables based on AM plans for the short term, such as the next week or month.

**Emergent decisions:** - This type of decision is needed when unplanned events occur, e.g., a shaft cracks. These decisions often have to be made in a short time – half an hour to a day – to decide whether the failure-related assets should be shut down, and/or should be repaired immediately.

Asset management is an overall methodology and decision making framework that aims to the evidence-based justification and optimisation of investments in infrastructure assets. Although it is mostly perceived as strategic level tool, it nevertheless affects and can equally be successful in lower levels of decision making.

Asset management decision processes are the individual decisions that need to be made in every level of decision making, whether that is of a strategic, network, or project focus. Decision processes can therefore be concerned with budget allocations, network
optimisation, work’s programming, and selection of alternative implementation methods, among others. Decisions made at different levels of asset management are heterogeneous and the supporting data needs are bound to be quite different.

CIEAM is conducting research on the “Multi Criteria Asset Decision Support Tool”. In the part of this research Yong (2008) developed the AM Split Decision Making Process [96]. While AM involves making numerous different decisions. This project only considers two types of decisions:

1. Asset operational and maintenance (O&M) decisions
2. Asset capital renewal decisions

This research aims to address the following issues:

1. Development of asset management decision making process for selected decisions
2. Identification of suitable technologies and methodologies for each selected decision.

In this approach the relationship among the different AM decision types, roles and time scales is described using a multiple-scale decision-making framework. An AM decision framework is a conceptual model or process guideline on how to make AM decisions effectively through proper integration of various AM decision models and methodologies. Figure 3.1 shows that different types of decisions are not isolated. They have interactions with each other, e.g., repeated needs to make short-term corrective repairs to a particular asset may lead to a change in its long-term maintenance strategy. On the other hand, short-term decisions have to be in compliance with the long-term strategies of an organisation. An AM decision-making process has to enable decision makers to deal with these interactions effectively.
Based on the general decision making process that can cope up with AM decisions in different time-scales, the split decision making process had been proposed. The split process divides the decision making process into a basic decision making process and a number of decision information acquisition/generation processes. The former focuses on decision making activities only while the latter provides inputs for decision making. This division leads to the concept of a “split” AM decision-making process model.

The split decision making process can effectively solve different decision time scale problems. When making AM decisions based on these models, one has to implement both the basic decision-making activities and the decision-support information generation and analysis activities. In practice, however, basic decision-making activities are necessary for all AM decisions, not every AM decision needs to implement the information generation and/or analysis activities. For example, once failure modes have been identified, decision makers only need to use the results of this process for subsequent

Figure 3.1: A typical multi-scale decision making framework [96]
decision making. They do not need to repeat the decision support work in each decision making task. In addition, implementation of the information generation and/or analysis activities often takes a long time which is not suitable for those AM decisions which need to be made within a short period such as reactive decisions.

![Figure 3.2: Concept of the split decision making process [96]](image)

**Technologies and Methodologies Identification**

Selecting suitable technologies and methodologies is important for ensuring the effectiveness and efficiency of AM decision support. In this process the selection of models/methods is as follows:

1. The models/methods must be able to generate outcomes which are accurate enough for industrial needs.
2. The models/methods must be mature and have a number of successful applications.
3. The models/methods must be simple to implement using computer programmes.
4. The models/methods must be easy to understand.
5. The models/methods should complement one another

**Decision Option Selection**

Option selection from a set of decision alternatives is conducted based on a comparison of the alternatives. A number of well established technologies and methodologies have been developed for comparing and ranking decision options. The decision models/methods that are chosen from the literature based on the above criteria are [96]:

60
- **Decision Trees**: A decision tree is a decision support tool to rank a set of predefined decision options. It uses a graph to represent possible decision options and the possible outcomes of each option. The probability, consequences, resource costs, and utility of each outcome are also described in the decision tree. A decision tree is used to identify the decision option which is most likely to reach a goal.

- **The Analytic Hierarchy Process (AHP)**: The AHP is a decision making technique developed by Thomas Saaty in 1970. It can be used for selecting a decision when there are a limited number of decision options and each has a number of attributes. The basic concept in the AHP is to use a hierarchy to describe a decision problem and reduce complex decisions to a series of pair wise comparisons, and then synthesize the results.

- **Multistage Comprehensive Fuzzy Evaluation (MCFE)**: MCFE was developed based on the fuzzy set theory introduced by Zadeh in 1962. Fuzzy set theory allows modelling imprecise concepts, in linguistic form, like ‘slightly’, ‘quite’, and ‘very’ using membership values range from 0 to 1. MCFE was proposed to apply to multiple attribute evaluation problems where a ranking of alternatives have been predefined. It is also known as Multi-level Comprehensive Fuzzy Evaluation or Appraisal. MCFE provides a formalised approach to evaluating a decision candidate which involves uncertain concepts.

**Decision Optimisation**

To make optimal decisions, linear programming is adopted with consideration of random variables and constraints. Specifically the optimisation approach has developed for minimising the total expected AM cost. This step need to be developed in future, hence we not able to provide much information.

**Top priority tasks working on the process**

1. Develop a Decision Rule Matrix which summarises and accumulates AM decision makers experience and knowledge.
2. For decision models, we choose to focus on Decision Trees first. This methodology can take into account uncertainties of outcomes which are often a challenge in industry. AHP and MCFE modules can be developed later.

3. For optimisation models, still discussions going on to decide for the best approach.

To implement this generic high level AM split decision making process, it must be specified with all details of the analysis. Such a decision making process has been developed based on some selected modelling technologies.

The main aim of this research is to provide a practical decision making process for AM. This process is based on the newly-developed generic decision making process, and on selected decision models/methods, and to develop a software tool to demonstrate the feasibility of the decision process. Also mentioned, the main possible future direction in such a system would be the coordinator agents and the functionalities for the coordinator agents are as follows:

- Accept and analyse inquiries from users and/or other coordinators.
- Assign tasks to appropriate basic analysis agents and/or lower level coordinator agents.
- Accept outcomes from the basic analysis agents or lower level coordinator agents.
- Synergise the outcomes and feed back the final result to the users and/or the other coordinator agents.

Figure 3.3: Concept of the multi-agent system based split decision support system
The above mentioned functionalities of Figure 3.3, the coordinator-agents are available in the workflow models and this is the main key aspect for taking the above future direction as my research direction and proceeded in that way. As mainly, workflows are used to describe the tasks, procedural steps, organizations or people involved, required input and output information, and tools needed for each step in a process. Hence in this research we are trying to model the AM split decision making process by using the workflow modelling language and that to in particular with YAWL language because of its various benefits, which will be discussed in next Section 3.5.

3.3 Asset Management Split Decision Making (AMSDM) Process

Based on the above concept and taking into account the NAMS Group’s decision process model and Rhodes’ five-step process model, our generic split AM decision-making process model is structured as shown in Figure 3.5.

3.3.1 Description of the AMSDM Process

The first step in this process model is to identify an AM decision which needs to be made. Asset Management involves numerous decisions, from routine maintenance planning to how to respond to an unexpected failure. Different decisions need different information and analyses.

The second step is to identify the objectives and the constraints for making the decision. Accurately recognising the decision objectives and constraints is imperative because these objectives and constraints define the criteria for optimising the decision. The objectives of a specific AM decision have to be in compliance with the overall business objectives. Therefore, to identify the objectives and constraints, one often needs to conduct a number of analyses which may take a long time to complete. These analyses are not suitable for those decisions that need to be made within a short time period. Fortunately, although every decision must be based on a clear understanding of the objectives and constraints, this does not necessarily mean that the objective and constraint analyses have to be conducted during each decision making event. Instead the analysis of objectives and constraints can be completed in advance, based on the experience and knowledge of domain experts, in order to produce a set of ‘pre-packaged’ decision
options which can be applied quickly based on the current system or asset state only. To allow this, the AM decision objective and constraint identification process has been separated from the basic decision-making process in Figure 3.

The third step is to gather the health status and operational information of assets which are associated with the identified decision. This step is essential for all AM decision making. It includes identifying asset failure modes and causes, assessing assets’ current conditions and predicting the next failure times of the assets or their future reliability changes. The impact of AM decisions on improving asset health conditions also needs to be analysed. Asset health assessment and prediction is often time-consuming since it typically involves gathering and analysing historical data for a large number of assets. For the same reason as mentioned above, this long-term asset health assessment and prediction process is separated from the short-term basic decision-making process in our model.

The fourth step is to identify all potential decision options. In engineering asset management, some options are discrete (e.g., replace a component) while others are continuous (e.g., increase the frequency of inspections). After the decision options have been identified, one then needs to shortlist them against “deal-breaker” rules, i.e., eliminate those options that cannot meet the objectives and/or constraints. When there are large numbers of options or there are continuous options, obtaining a shortlist of decision options becomes difficult. Note in Figure 3.5 that there is a feedback loop from decision verification (at the end of the process) to decision option identification. If all shortlisted decision options are unsatisfied, the decision makers may come back to look at those options previously discarded. In some cases, discarded options become viable again because of changes to the decision objectives and constraints.

The fifth step is to rank the decision options based on decision criteria which are determined according to the decision objectives and constraints. In modern asset management, AM decisions often involve multiple actors, different objectives and constraints, i.e., AM decision making usually belongs to multiple criteria decision problems. As a result, ranking decision options is often difficult. To address this issue, various decision option ranking models and methodologies have been developed, such as Decision Trees, the Analytic Hierarchy Process, and so on as explained in Section
3.2.1.1. These techniques can effectively assist in AM decision ranking. However, using them correctly typically requires a sound knowledge of how they work and, in particular, an understanding of their limitations. In addition, in most cases, it also takes a significant amount of time to conduct decision option ranking analyses, so the ranking process is also separated from the basic decision-making process in our model.

The sixth, seventh and eighth steps are to optimise the decision parameters. After the fifth step, the decision options have been ranked. Then one can determine a best option based on the rankings. However, the determination of the best option does not necessarily mean that the AM decision can be finalised because AM decisions are so complex. In practice, further analysis may be needed to optimise the parameters which are associated with the selected decision option. For example, when the reliability of an asset is lower than an acceptable level, a number of maintenance activities can be applied to improve its reliability, including conducting preventive maintenance or renewing the whole asset. If a decision to renew an asset is made, then one needs to further decide on the optimal renewal time. To address these issues, our split decision-making process model has additional steps in which we need to identify data availability and then conduct optimisation analysis using an appropriate optimisation model or method based on the decision objectives and constraints which have been identified in the second step.

The ninth step is to verify the decision. This step is the last action in our split AM decision-making process model. It usually involves a number of what-if analyses to ensure that the decision chosen is robust. Once the decision has been validated, it becomes the final one. If the selected decision option is not satisfactory, the decision makers need to modify the objectives or reconsider the decision options.

The basic notations used in the process are shown in figure 3.4. The split decision making process is shown in Figure 3.5. In this figure, yellow blocks form a basic decision making process. Green double line blocks represent the information generation processes. More details regarding the information generation processes named AM decision objectives are shown in the AM decision objectives and constraints identification process which is shown in Figure 3.6. The details regarding the asset health and its condition are given in the asset health assessment and prediction process, which is shown in Figure 3.7. The decision option ranking process, relationship analysis process and simulation process
are all models/method dependent. An example of the part of the option ranking process and optimisation process are shown in Figures 3.8 and 3.9 respectively.

Figure 3.4: The notations for the Figures 3.5 to 3.9 [96]
Figure 3.5: Asset management split decision making (AMSDM) process [96]
Figure 3.6: Asset management decision objectives identification process [96]
Figure 3.7: High level asset condition assessment and prediction process [96]
Figure 3.8: Option ranking process [96]
Figure 3.9: Optimisation Process [96]
3.3.2 Analysis of the AMSDM Process

Basically when we look at the majority of decision making processes, they are all based on the several basic kinds of decisions [94]. These decisions will be compared with the AMSDM decision making process and the following justifications are made:

**Decisions Whether:** This is the yes/no, either/or decision that must be made before we proceed with the selection of an alternative. E.g. Should I buy a new house? Decisions whether are made by weighing reasons pro and con. It is important to be aware of having made a decision whether, since too often we assume that decision making begins with the identification of alternatives, assuming that the decision to choose one has already been made. Similarly in the AMSDM process, at each and every point the Decision Whether check is available to verify with the various decision makers whether to precede the flow in the direction of yes or in the direction of no.

**Decisions Which:** These decisions involve a choice of one or more options, from among a set of possibilities. The choice being based on how well each option measures up to a set of predefined criteria. Compared this kind of decision with the AMSDM process, we can notify that the task of defining various decision options and the task of selecting the best option are the various possibilities for the AM decision. And selecting the best possibility among the various possibilities shows the choice on how well each possibility measures up.

**Contingent Decisions:** These are decisions that have been made but put on hold until some condition is met. E.g. If we decided to buy a car, if we can get it for the right price; most people carry around a set of already made, contingent decisions, just waiting for the right conditions or opportunity to arise. Time, energy, price, availability, opportunity, encouragement, etc all these factors can figure in to necessary conditions that need to be met before we can act on our decision. Similarly, for the AMSDM process also, we take a decision and double check whether that decision is suitable for the present state of asset. For example, all the analysis models in our AMSDM process are to cross check twice before making a decision.
Based on all these kinds of decisions the AMSDM process has been developed. These kinds of decisions will come through, some before the execution of process and some during the execution.

### 3.3.3 Characteristics of the AMSDM Process

Based on the thorough analysis of the AMSDM process and the various AM decision making processes eight characteristics have been identified for the AMSDM process and they are explained as follows:

1. **Information:** This is the knowledge about the decision, the scope and limitations of the decision, to include the clarification of goals, the effects of its options, the probability of each option, and so forth. A major point to make here is that while substantial information is desirable, the statement that “the more information, the better” is not true (Shien Lin, Murphy, Glen D 2008). Too much information can actually reduce the quality of a decision. The justification for this statement is explained in five points:

   a. A delay in the decision occurs because of the time required to obtain and process the extra information. This delay could impair the effectiveness of the decision or solution.
   
   b. Information overload will occur. In this state, so much information is available that decision-making ability actually declines because the information in its entirety can no longer be managed or assessed appropriately.
   
   c. Selective use of information will occur. That is, the decision maker will choose from among all the information available only those facts which support a preconceived solution or position.
   
   d. Mental fatigue occurs, which results in slower work or poor quality work.
   
   e. Decision fatigue occurs, where the decision maker tires of making decisions. Often the result is fast, careless decisions or even decision paralysis--no decisions are made at all.

2. **Alternatives:** These are the possibilities one has to choose from. Alternatives or Options can be identified (that is searched for and located) or even developed (created
where they did not previously exist). Merely searching for pre-existing alternatives will result in less effective decision making.

3. **Criteria:** These are the requirements that each option or alternative must possess to a greater or lesser extent. Usually the options are rated on how well they possess each criterion. For example, option Replace ranks a 6 on the criterion, while option Maintenance ranks a 2 on the same criterion.

4. **Goals:** What is it we want to accomplish? For example, I want to buy a car, and then ask “Which should I choose?” without thinking first of what the goals are, what overall objective we want to achieve. We need to ask ourselves, “What should we do? What should we choose?” so that we can come to clarification of “what are our goals?” A component of goal identification should be included in every instance of decision analysis.

5. **Value:** Value refers to how desirable a particular outcome is, the value of the option, whether in dollars, satisfaction, or other benefit.

6. **Preferences:** These reflect the philosophy and moral hierarchy of the decision maker. We can say that personal values dictate preferences. Some people prefer certainty to risk, efficiency to aesthetics, quality to quantity, refurbish to maintenance, and so on. This, when one person chooses to ride the wildest roller coaster in the park and another chooses a mild ride, both may be making good decisions, if based on their individual preferences.

7. **Decision Quality:** This is a rating of whether a decision is good or bad. A good decision is a logical one based on the available information and reflecting the preferences of the decision maker. The important concept to grasp here is that the quality of a decision is not related to its outcome: a good decision can have either a good or bad outcome. Similarly, a bad decision can still have a good outcome.

Good decisions that result in bad outcomes should thus not be cause for guilt or recrimination. If we decide to take the scenic route based on what we know of the road (reasonably safe, not heavily travelled) and our preferences (minimal risk, prefer scenery over early arrival), then our decision is a good one, even though we might happen to get
in an accident, or have a flat tire in the middle of nowhere. It is not justified to say, "Well, this was a bad decision."

In judging the quality of a decision, in addition to the concerns of logic, use of information and options, three other considerations come into play:

A. *The decision must meet the stated objectives most thoroughly and completely.* How well does the option chosen meet the goals identified?

B. *The decision must meet the stated objectives most efficiently, with concern over cost, energy, and side effects.* Are there negative consequences to the alternative that make that choice less desirable? We sometimes overlook this consideration in our search for thrills.

C. *The decision must take into account valuable by-products or indirect advantages.* A new employee candidate may also have extra abilities not directly related to the job but valuable to the company nonetheless. These should be taken into account.

8. **Acceptance:** Those who must implement the decision or who will be affected by it must accept it both intellectually and emotionally. Acceptance is a critical factor because it occasionally conflicts with one of the quality criteria. In such cases, the best thing to do may be to choose a lesser quality solution that has greater acceptance.

For example, when cake mixes first were put on the market, manufacturers put everything into the mix—the highest quality and most efficient solution. Only water had to be added. However, the mixes didn't sell well—they weren't accepted. After investigation, the makers discovered that women didn't like the mixes because using the mixes made them feel guilty: they weren't good wives because they were taking a shortcut to making a cake. The solution was to take the egg and sometimes the milk out of the mix so that the women would have something to do to "make" the cake other than just adding water. Now they had to add egg and perhaps milk, making them feels more useful. The need to feel useful and a contributor is one of the most basic of human needs. Thus, while the new solution was less efficient in theoretical terms, it was much more acceptable. Cake mixes with the new formula became quite popular.
Thus, the inferior method may produce greater results if the inferior one has greater support. One of the most important considerations in decision making, then, is the people factor. Always consider a decision in light of the people implementation. Only decisions that are implemented, and implemented with thoroughness (and preferably enthusiasm) will work the way they are intended to.

These are the various characteristics that are identified for the AMSDM process. Based on these characteristics in next chapters, we try to improve the AMSDM process, and explain in detail how these characteristics will be applied for modelling the AMSDM process.

3.4 Association between AMSDM Process and WFM Technique

In this section, we try to explain the connection or the link between the AMSDM processes with the workflow modelling technique. As we know from the Gaps and Opportunities of Sections 2.3 and 2.5, that there is a need for modelling the AM processes. In addition to that, we use workflow modelling techniques in particular because, to provide the chance for automating the AM processes. The benefits of applying WFM to the AMSDM process are to optimise efficiency of the process by delivering the right work to the right people at the right time, to gain better process control, improves the flexibility of the process, etc. This leads to the application of workflow modelling technique for the AM process. Among the AM processes, we choose the AMSDM process as the high level AM process for modelling, which leads to the first step in modelling the AM process. Figure 3.10 depicts the steps, how the AMSDM process and the workflow languages are linked together to develop the AMSDM workflow model.

The AMSDM workflow model will be developed by following the two parallel step wise processes. Among these two parallel processes, one process is for AMSDM process and another process is for workflow languages.
AMSDM process: In this step wise procedure, the first activity AMSDM process is to suggest that, among several AM decision making processes, we selected the AMSDM process for modelling. This activity is performed in the literature review of Section 2.2 and the details regarding the AMSDM process is given in Sections 3.1 to 3.3. The next following activity to this is the Analysis of AMSDM process. This step is to thoroughly analyse the AMSDM process, i.e. identifying the various characteristics of AMSDM process by comparing with the various decision making process. This is performed in

Figure 3.10: Association between AMSDM process and YAWL workflow model
Section 3.3.1. In addition to that how these identified characteristics are suitable for modelling with the YAWL language, which will be discussed in Section 5.1.1.

The next activity after analysing is the Ameliorating the AMSDM process. This is the main important step, among the AMSDM process step wise process, this activity is one of the main contributions in achieving the AMSDM workflow model. In this step we are going to explain the gaps in the AMSDM process and how to fill these gaps. Filling these gaps will leads to an ameliorated process definition which specifies the information needed for workflow modelling. Thus it can be used directly for modelling in a workflow language. In a broad sense, it bridges between the design of a process definition by a domain expert and the deployment of the process using BPM technology. This step will need to be discussed in Chapter 4.

**Workflow Languages:** - The second parallel step wise process is for carrying on with the workflow languages. Among these steps, the first activity is to point out that, among the various business process modelling languages, we selected the workflow languages for modelling the AM process. This is because, to develop the best AM process model and that to which can be automated. This complete step was discussed in Section 2.4. The next activity is to analyse the various workflow languages. This analysis is for choosing the best workflow modelling language. Among the various workflow modelling techniques like BPMN, BPEL, XPDL, YAWL, etc, which technique will be best suitable for modelling the AM processes. For example, BPMN is graph-oriented, which means that a model captured in BPMN can have an arbitrary topology, whilst most BPEL constructs are block-structured, which means that if a segment of a BPEL model starts with a branching construct it ends with the corresponding synchronisation construct. A mapping from BPMN to BPEL such as the one proposed in [76] needs to handle the above mismatches properly and may still result in BPEL code that is hard to understand. Although BPEL provides control links that can be used to connect various constructs, it imposes a number of constraints on the usage of control links (e.g. they cannot form a loop nor cross the boundary of a loop) so that the support for a graph-based modelling is restricted.
Due to these various results from analysis, we preferred YAWL as the best suitable modelling language. The various workflow languages were explained in Section 2.4 and the YAWL workflow language in detail will be discussed in Section 3.5.

Upon the analysis, we decide the YAWL workflow language as the best suitable language for modelling the AMSDM process. Hence the next step is to learn the YAWL language. The main concepts and the principles of YAWL will be discussed in next Section 3.5.

**AMSDM Workflow Model:** - Now the last step is to combine the both parallel processes to achieve the AMSDM workflow model i.e. modelling the ameliorated AMSDM process definitions with the YAWL workflow language. This is the most important step in which we associated two separate concepts and form a new concept. The complete modelling of AMSDM process will be explicated in Chapter 5.

### 3.5 Yet Another Workflow Language (YAWL)

YAWL is a powerful modelling language based on the well-known workflow patterns initiative. This workflow patterns initiative is joint effort of Eindhoven University of Technology led by Professor Wil van der Aalst and Queensland University of Technology led by Professor Arthur ter Hofstede, started in 1999. The aim of this initiative is to provide a conceptual basis for process technology. The research upon the patterns provides the thorough examination of the various perspectives that need to be supported by a workflow language or a business process modelling language. YAWL is supported by a state of the art open source support environment, which is well established in the field of business process management [90]. The environment supports business process modelling, correctness analysis, simulation, deployment, and post-execution analysis. YAWL is inspired by the high-level Petri nets, but extended with some additional features to meet the requirements, defined by the workflow patterns, especially the models defining the multiple instances, advanced synchronisation and cancellation.

YAWL is based on the rigorous analysis of existing workflow management systems and related standards, this analysis shows that the contemporary workflow systems, related standards (e.g. XPDL, BPML, BPEL), and theoretical models such as Petri net have problems supporting essential patterns [90].
Let us first go through some basic terms or definitions that are used in YAWL.

**Business Process:** A set of interdependent activities that need to be performed by in response to a business event, to achieve a business objective.

**Workflow Application:** A software application that coordinates the tasks that compose a business process, in whole or part. Sometimes the term “workflow” is used as shorthand for “workflow application”.

**Workflow Specification:** (Also known as *Workflow Model*) A description of a business process to the level of detail required for its deployment into a workflow engine. A workflow specification defines which tasks should be performed, under which conditions and in which order, which data, documents and resources are required in performing each task, etc.

**Workflow System:** A system that can be used to develop and to run a workflow application. A workflow system usually includes a process editor to support the design of workflow models, a workflow engine to support the execution of workflow models, and at least one worklist handler.

**Workflow Engine:** The runtime component of a workflow system responsible for determining which tasks need to be performed and when, for maintaining execution logs, and for delegating the performance of tasks to software applications/services or to a worklist handler.

**Case:** (Also known as *Workflow Instance*) A specific execution of a workflow model as a result of an event. For example, an order management workflow is instantiated every time new order arrives. Each of these orders leads to a different case.

**Task:** (Also known as *Activity*) A description of a unit of work that may need to be performed as part of a workflow. Workflow models are composed of tasks. A task may be either manually carried out by a person or automatically by a software application.
**Work item:** (Also known as *Task Instance*) A particular instance of a task that needs to be performed as part of a given workflow instance.

**Worklist:** A list of work items.

**Worklist Handler:** (Also known as *Task Management Service*) A software component that manages work items issued by a workflow engine and that assigns, prioritises and presents these work items to human participants according to policies that may be configured in the workflow model.

YAWL is a new proposal of a workflow/business processing system that supports a concise and powerful workflow language and handles complex data transformations and web service integrations. As it implements the most common workflow patterns, YAWL can be used as a lingua franca for expressing the behaviour of web services. Despite its graphical structure, YAWL has a well defined formal semantics. It is a state-based language and the semantics of workflow specification is defined as a transition system.

### 3.5.1 YAWL Principles

YAWL is an original and sophisticated workflow language building on insights gained from the workflow patterns research and on concepts from Petri nets (ultimate mix of theory and practice). YAWL is precisely defined, vendor independent, powerful approach to business process management. YAWL is a very powerful, yet fundamentally simple language for process modellers to describe even the most complex business process. This language is very unique in terms of:

- **Expressive**- YAWL is based on workflow patterns. The patterns have been used to evaluate many Business process management (BPM) languages and have influenced functionality of and open source BPM systems.

- **Rigorous**- YAWL is based on Petri nets and is formally defined. Specifications have a precise definition and sophisticated design time support for analysing workflows (thus catching potentially costly mistakes before deployment),
Adaptive- YAWL offers a unique solution for evolving workflows. This allows businesses to be highly responsible to process changes resulting from today’s highly dynamic environments. In addition, YAWL provides powerful support for handling both expected and unexpected exceptions.

Extensible- The architecture of YAWL environment is service-oriented. By exploiting well-defined interfaces of the components of this environment new functionality can be added in a seamless manner. The environment is open source and can therefore be modified by third parties.

Useable- Control flow dependencies in YAWL can be specified graphically and an intuitive design time environment exists. While YAWL is a highly expressive language, its concepts are relatively straight forward.

Configurable- YAWL provides conceptual support for a repository of common model practices in a given domain that can be configured for a specific setting, such as an enterprise or a project.

Due to these principles and its benefits, we choose YAWL as the best executable workflow modelling language for modelling the AMSDM process.

3.5.2 The YAWL System

To support the YAWL language, the YAWL system had been developed, which uses state-of-the-art technology [84]. First we would like to discuss the overall architecture of this system, which is depicted in Figure 3.11. Workflow specifications are designed using the YAWL designer and deployed into the YAWL engine which, after performing all necessary verifications and task registrations, stores these specifications in the YAWL repository, which manages a collection of runable workflow specification.
Once successfully deployed, workflow specifications can be instantiated through the YAWL engine, leading to workflow instances (or *cases*). The engine handles the execution of these cases, i.e. based on the state of a case and its specification, the engine determines which events it should offer to the environment. The environment of a YAWL system is composed of so-called YAWL services. Inspired by the “web services” paradigm, end-users, applications, and organisations are all abstracted as services in YAWL. Figure 3 shows the three standard YAWL services: (1) YAWL Resource Service, with integrated worklist handler and administration tool; (2) YAWL web services invoker; and (3) YAWL worklet service, which provides dynamic flexibility and exception handling capabilities.

The YAWL worklist handler corresponds to the classical worklist handler (also named “inbox”) present in most workflow management systems. It is the component used to assign work to users of the system. Through the worklist handler users are offered and allocated work items, and can start and signal their completion. In traditional workflow systems, the worklist handler is embedded in the workflow engine. In YAWL however, it is considered to be a service completely decoupled from the engine. The YAWL web services invoker is the glue between the engine and other web services. Note that it is
unlikely that web services will be able to directly connect to the YAWL engine, since they will typically be designed for more general purposes than just interacting with a workflow engine. Similarly, it is desirable not to adapt the interface of the engine to suit specific services; otherwise, this interface will need to cater for an undetermined number of message types. Accordingly, the YAWL web services broker acts as a mediator between the YAWL engine and external web services that may be invoked by the engine to delegate tasks (e.g. delegating a payment task to an online payment service). The YAWL interoperability broker is a service designed to interconnect different workflow engines. For example, a task in one system could be subcontracted to another system where the task corresponds to a whole process.

Each service shown in Figure 3.11 conforms to the architecture of a so-called custom YAWL service, and any number of custom services can be implemented for particular interaction purposes with the YAWL engine. A custom service connects the engine with an entity in the environment. For example, a custom YAWL service could offer communication with mobile phones, printers, assembly robots, etc. Note that it is also possible that there are multiple services of the same type, e.g. multiple worklist handlers, web services brokers, and exception handling services. For example, there may exist multiple implementations of worklist handlers (e.g. customised for a specific application domain or organisation) and the same worklist handler may be instantiated multiple times (e.g., one worklist handler per geographical region) [101].

As alluded to earlier, services interact with the engine and each other via a number of interfaces, which provide methods for object and data passing via http requests and responses. All data are passed as XML; objects are marshalled into XML representations on the serving side of each interface and reconstructed back to objects on the client side. The YAWL engine provides four interfaces:

- Interface A: which provides endpoints for process definition, administration and monitoring;
- Interface B: which provides endpoints for client and invoked applications and workflow interoperability, and is used by services to connect to the engine, to start and cancel case instances, and to check work items in and out of the engine;
- Interface E: which provides access to archival data in the engine’s process logs; and
- Interface X: which allows the engine to notify custom services of certain events and checkpoints during the execution of each process instance where process exceptions either may have occurred or should be tested for.

The resource service also provides three interfaces to allow developers to implement other worklist handlers and administration tools while leveraging the full functionality of the service. Interface R provides organisational data to external (authorised) entities such as the YAWL Process Editor; Interface W provides access to the internal work queue routing functionalities; and Interface O allows organisational data to be provided from any data source. In addition, the service’s framework is fully extendible, allowing further constraints, filters and allocation strategies to be “plugged in” by developers.

Workflow specifications are managed by the YAWL repository, and workflow instances are managed by the YAWL engine. Clearly, there is also a need for administration tool that can be used to control workflow instances manually (e.g. deleting a workflow instance or a workflow specification), manually allocate resources to tasks, and provide information about the state of running workflow instances and details or aggregated data about completed instances. This is the role of the administration tools integrated into the resource service.

**Design and Runtime Environment**

The YAWL Editor provides a GUI design environment for the specification and verification of YAWL workflows. It is an independent tool that interacts with the YAWL Engine via Interface A and with the Resource Service via Interface R. In chapter 5, we show in detail how the Editor supports the modelling of AMSDM process control-flow dependencies, data passing, and resource management.

The YAWL runtime environment supports worklist handling where each work item is handled based on the three interaction points, i.e. *offered, allocated, and started* (with *suspended* a derivative of *started*), within a work item lifecycle. The user interface in the runtime environment is presented by means of Web pages and forms within a browser. In Chapter 5, we will discuss the information about the AMSDM process model work item that has to be allocated, including the process specification, the identifier of process instance, and the task that the work item belongs to, its creation time and age.
Dynamic Workflow

Workflow management systems are used to configure and control structured business processes from which well-defined workflow models and instances can be derived. However, the proprietary process definition frameworks imposed make it difficult to support: (i) dynamic evolution (i.e. modifying process definitions during execution) following unexpected or developmental change in the business processes being modelled; and (ii) deviations from the prescribed process model at runtime.

Without support for dynamic evolution, the occurrence of a process deviation requires either suspension of execution while the deviation is handled manually, or an entire process abort. However, since most processes are long and complex, neither manual intervention nor process termination is satisfactory solutions. Manual handling incurs an added penalty: the corrective actions undertaken are not added to “organisational memory”, and so natural process evolution is not incorporated into future iterations of the process [101].

The YAWL system provides support for flexibility and dynamic exception handling through the concept of worklets [102]. The worklet executed for a task is run as a separate case in the engine, so that, from an engine perspective, the worklet and its parent are two distinct, unrelated cases. The worklet service tracks the relationships, data mappings and synchronisations between cases. Any number of worklets can form the repertoire of an individual task, and any number of tasks in a particular specification can be associated with a worklet. A worklet may be a member of one or more repertoires, i.e. it may be re-used for several distinct tasks within and across process specifications.

3.5.3 Process Modelling and Execution using YAWL

YAWL supports both process modelling and execution. Each YAWL construct has both a graphical representation and an executable semantics, and thus a process model written in YAWL is directly executable. Some existing process languages such as BPMN and EPC [103] take a different approach and focus on the specification of intuitive models that can be easily understood by the various stakeholders. For process automation, these models need to be transformed to models specified in an executable language such as BPEL or YAWL. A typical example of this approach is the use of BPMN in conjunction with BPEL. However, there are obvious drawbacks to this separation of modelling and
execution, especially when both languages are based on different paradigms or when the modelling language contains potentially complex concepts and little consideration was given to their precise meaning. For example, BPMN is graph-oriented, which means that a model captured in BPMN can have an arbitrary topology, whilst most BPEL constructs are block-structured, which means that if a segment of a BPEL model starts with a branching construct it ends with the corresponding synchronisation construct. Although BPEL provides control links that can be used to connect various constructs, it imposes a number of constraints on the usage of control links (e.g. they cannot form a loop nor cross the boundary of a loop) so that the support for a graph-based modelling is restricted. A mapping from BPMN to BPEL such as the one proposed in [76] needs to handle the above mismatches properly and may still result in BPEL code that is hard to understand.

**Background in Workflow Patterns**

To gain a better understanding of the fundamental concepts underpinning business processes, the Workflow Patterns Initiative [86] was conceived in the late 1990s with the goal of identifying the core architectural constructs inherent in process technology. After almost a decade of research, more than 120 workflow patterns have been identified in the control-flow, data, and resource perspectives. The control-flow perspective captures aspects related to execution order of various tasks in a process, e.g. sequence, choice, parallelism and synchronisation. The data perspective describes how data elements are defined and utilised during the execution of a process. The resource perspective deals with the overall organisational context in which a process functions and the issue of resource to task allocation. In addition to these, the exception handling perspective deals with the various causes of exceptions and the various actions that needs to be taken as a result of exceptions occurring. In the following we outline the patterns in each of these perspectives.

*Control-flow patterns* describe structural characteristics of a business process and the manner in which the thread of execution flows through the process model. Originally 20 control-flow patterns were proposed [87], but in the latest review this has grown to 43 patterns [93].

*Data patterns* aim to capture a series of data characteristics that occur repeatedly in business processes. In total 40 patterns were defined [91], which can be divided into four
distinct groups. 1) Data visibility patterns 2) Data interaction patterns 3) Data transfer patterns 4) Data-based routing patterns.

Resource patterns aim to capture the various ways in which resources are represented and utilised in business processes. In total 43 patterns were identified [91], which can be classified into seven categories mostly based on the typical work item lifecycle (which includes states such as offered, allocated and started). These are:


Exception patterns form a classification framework for dealing with exceptions that occur during the execution of a process. In general, an exception relates to a specific work item in an instance of a process (or case) being executed. The exception handling strategies are proposed respectively at work item level (e.g. re-allocating a work item to a different resource due to the unavailability of the resource that the work item was allocated to) and at case level (e.g. removing all remaining work items in the current case). Consideration is given to what recovery action needs to be taken to remedy the effects caused by an exception occurring (e.g. to compensate for the effects of the exception).

3.5.3.1 YAWL Language Overview

YAWL provides a comprehensive reference language for describing business processes that are to be enacted as workflows. The language constructs in YAWL are informed by the various workflow patterns, hence they have a direct correspondence with the fundamental elements which are actually encountered in real-world business processes and consequently have general applicability. The YAWL language is specified in two parts. It has a complete abstract syntax which identifies the characteristics of each of the language elements and their configuration. Associated with this is an executable, semantic model — presented in the form of Coloured Petri nets — which defines the runtime semantics of each of the language constructs.

The abstract syntax for YAWL provides an overview of the main concepts that are captured in a design-time business process model. It is composed of five distinct schemas, each of which is specified on a set-theoretic basis. Figure 3.12 summarizes the
content captured by each of the individual schemas and the relationships between them [97]. Each process captured using the YAWL abstract syntax has a single instance of the YAWL specification associated with it. This defines elements that are common to all of the schemas and also captures an instance of the organisational model that describes which users are available to undertake tasks that comprise the process and the organisational context in which they operate.

Figure 3.12: Schema definitions for YAWL abstract syntax [97]

YAWL process can be made up of a series of distinct sub-processes (where each sub-process specifies the manner in which a composite task is implemented) together with the top-level process. For each of these (sub) processes, there is an instance of the YAWL net which describes the structure of the (sub) process in detail in terms of the tasks that it comprises and the sequence in which they occur. Associated with each YAWL net is a data passing model which defines the way in which data is passed between elements in the process in terms of formal parameters operating between these elements. There is also a work distribution model that defines how each task will be routed to users for execution, any constraints associated with this activity and privileges that specific users may have assigned to them. The collective group of schemas for a specific process model is termed a complete YAWL specification.
The workflow patterns have been used to evaluate a wide range of existing workflow products and standards in terms of the control-flow, data, resource, and exception handling perspectives. They have been found to be especially useful for the comparison of process languages, for tool selection, and also for the identification of specific strengths and weaknesses of individual tools and languages. Details of these evaluation results and impact the workflow patterns have made in the past few years can be found on the web site of the Workflow Patterns Initiative.

YAWL, although inspired by Petri nets, is a completely new language with its own semantics and is specifically designed for workflow specification. Initially, to overcome the limitations of Petri nets, YAWL was extended with features to facilitate patterns involving multiple instances, advanced synchronisation patterns, and cancellation patterns. Moreover, YAWL allows for hierarchical decomposition and handles arbitrarily complex data. Over time, YAWL has also been extended to support resource management, exception handling, evolving workflows, and process verification [20].

**Control-flow Perspective**

YAWL extends the class of WF-nets with multiple instance tasks, composite tasks, OR-joins, and cancellation regions. In contrast to Petri nets and WF-nets, YAWL’s visual representation allows tasks to be directly connected as this can help compress a diagram (note this can only be done when the place in-between had one input task and one output task; the removed place formally still exists). Figure 4 shows the modelling elements of YAWL. A process definition in YAWL consists of *tasks*, which are transition-like objects, and *conditions*, which are place-like objects. Each process definition starts with a unique *input condition* and a unique *output condition*.

A workflow specification in YAWL is a set of workflow nets which forms a directed rooted graph (the root is referred to as the main or roots net). There are *atomic tasks* and *composite tasks*. Atomic tasks correspond to atomic actions, i.e. actions that are either performed by a user or by a software application. Each composite task refers to a child or sub net that contains its expansion. Both types of task can also be *multiple instance* tasks and thus have multiple concurrent instances at runtime. Also, as shown in Figure 3.13,
YAWL adopts the notations of AND/XOR-splits/joins used in WF-nets. Moreover, it introduces *OR-splits* and *OR-joins* which correspond to the Multi-Choice pattern and the Synchronising Merge pattern respectively. Finally, YAWL provides a notation for *removing tokens* from a specified region upon completion of a certain task. This is denoted by associating a dashed lasso to that task that contains the conditions and tasks from which tokens need to be removed or that need to be cancelled. This region is known as a *cancellation region* and this notion provides a generalisation of the Cancel Activity and Cancel Case patterns.

![Diagram of YAWL modelling elements](image)

**Figure 3.13: Modelling elements in YAWL [85]**

In YAWL each process has a unique identifier known as a ProcessID and each process model has a unique BlockID (this is necessary as the hierarchy within a process means it may contain several distinct process models defining sub workflows in addition to the top-level model). Each task within a process is identified by a unique TaskID. In order to allow for and differentiate between concurrent executions instances, it is necessary to introduce some additional notions. First executing instance of a process is termed as a case. It has a case identifier CID which is unique for a given ProcessID. Hence the tuple (ProcessID, CID) uniquely identifies all cases. Similarly an enabled task instance is known as a work item. It has a more complex identification scheme denoted by the
fivetuple (ProcessID, CID, TaskID, Inst, TaskNr) where Inst identifies the specific
instance of the task that is being executed (thus allowing for distinct instances of a task as
may occur if it is in a loop for example) and TaskNr which allows distinct concurrent
execution instances of a multiple instance task to be differentiated. By adopting this
identification scheme, it is possible for the semantic model to cater for multiple
concurrent processes, process instances and task instances in a common environment.

The control flow model for the AMSDM process will be developed in Chapter 5.

Data-flow Perspective

Like most programming languages, data elements are stored in variables in YAWL. There are net variables for storing data that can be manipulated by any individual task in
a net, and task variables for storing data that can be used or modified only within the
context of individual execution instances of a certain task. In the case of a composite task,
its task variables are conceptually the net variables of the corresponding subnet linked to
that task. Data passing between tasks is achieved by passing values between a net and its
tasks where XQueries may be used for transformation purposes. It is not possible to
directly pass data between tasks; since each task variable is local to its task (i.e. it is not
accessible by other tasks). The variables of a composite task serve as intermediate
variables for passing data from a higher level to a lower level of a process definition.
YAWL also supports exchange of information between a process and its environment
(i.e. workflow engine users and Web services). When data is required from the
environment at run time, either a Web form will be generated requesting the data from
the user or a Web service will be invoked that can provide the required data. In addition
to the above, data elements can be defined and used for conditional routing and for the
creation of multiple instances. If a task is an OR-split or XOR-split, its branching
conditions are specified as XPath Boolean expressions over certain variable(s) associated
with the task. The data carried by the variable(s) may determine the evaluation results of
the expressions and thus determine which branch (es) will be chosen.

All these net variables, task variables and data passing for the AMSDM process will be
described in Chapter 5.
**Resource Perspective**

The third key perspective of a process, after control-flow and data, is the resource perspective. The YAWL resource perspective provides direct support for 38 of the 43 identified resource patterns – the five remaining being particular to the case-handling paradigm. The resource perspective is responsible for describing the resources who undertake a given business process and the manner in which associated work items are distributed to them and managed through to completion. For each task, a specific *interaction strategy* is specified which describes how the associated work item will be distributed to users, and what degree of autonomy they have in regard to choosing whether they will undertake it or not and when they will commence executing it. Similarly, a detailed *routing strategy* can be defined which identifies who can undertake the work item. Users can be specified by name, in terms of roles that they perform, based on capabilities that they possess, in terms of their job role and associated organisational relationships or based on the results of preceding execution history. In YAWL, a human resource is referred to as a *Participant*. Each participant may perform one or more *Roles*, hold one or more *Positions* (each of which belongs to an *Org Group*) and possess a number of *Capabilities*.

Before going into details of resource perspective, first discuss the life cycle of work items i.e. the manner in which work items are advertised and ultimately bound to specific resources for execution [93]. Figure 3.14 illustrates the lifecycle of a work item in the form of state transition diagram from the time the work item is created through to final completion or failure.

![State Transition diagram for Work distribution](image)

**Figure 3.14: State Transition diagram for Work distribution [93]**
Each node in Figure 3.14 represents a possible state of a work item. Each edge within this diagram is prefixed with an S or R indicating that the transition is initiated by the workflow system (S) or resource (R) respectively. Initially a work item comes in to existence in the created state. This indicates that the preconditions required for its enablement have been satisfied and it is capable of being executed. At this point however, the work item has not been allocated to a resource for execution. In state created the system can take one of three possible courses of action: (1) offer the work item to a single resource, (2) allocate it to a resource, or (3) offer it to multiple resources. The difference between allocating and offering to a single resource is subtle. If a work item is allocated, there is a commitment on the part of the resource to execute the work item. Note that this commitment may be imposed by the system via S: allocate or by the resource him/herself (via R: allocate s or R: allocate m). No such commitment is implied to offered work items. Depending on the system and the resource, an offered or allocated work item can be started (cf., state started). Started work items can be completed, suspended (followed by a resume) or failed.

Workflow tasks that, at runtime, are required to be performed by a participant have their resourcing requirements specified at design time concomitantly with the design of the process control-flow and data perspectives, using the YAWL process editor. Conceptually, a task has three interaction points (places in a task lifecycle where distribution decisions can be made by the system and/or participants) [101]:

- **Offer**: A task may be offered to one or more participants for execution. There is no implied compunction (from a system perspective) for the participant to accept the offer.
- **Allocate**: A task may be allocated to a single participant, so that the participant is committed (willingly or not) to performing that task. If the task was previously offered to several other participants, the offer is withdrawn from them; and
- **Start**: A task is allocated to a participant and started (enters executing state).

Correspondingly, each participant may have, at any particular time, tasks in any of three personal work queues, one for each of the interaction points (a fourth, suspended, is a derivative of the started queue). A process designer must specify that each of the three interaction points be either user- or system-initiated. More details regarding these
interaction points and the resource items are given in Chapter 5. These all items will be discussed in the resource model section of the AMSDM workflow model in Chapter 5.

With these three perspectives, the process modelling with YAWL language had been successfully completed.

3.6 Summary

In this chapter, discussed about the AM decision making process and in particular explained the methodology of the AMSDM process and gave description about this process. For this AMSDM process, several characteristics were identified. These characteristics will assist in modelling the process. The next topic discussed in this chapter was how to associate the AMSDM process with the YAWL language. For this association, a framework was developed to show how these two broad disciplines will be combined together. The explained the YAWL language and its various principles.

This chapter is the crucial one, because this chapter has got all the basic important information related to this research. In addition to that it gives all the necessary connections between the various disciplines.
4.1 Introduction

There are gaps between the original AMSDM process flowchart (in Chapter 3) and a process definition that is ready for execution. Hence a detailed process definition that is ready for execution which will be called as the “Ameliorated AMSDM Process Definition” is developed. This ameliorated AMSDM process definition is one of the main contributions in this proposed research. The first gap lies in the order of performing tasks in the process. In the current AMSDM process definition, there are missing tasks and also for some tasks their execution order needs to be altered. The term “control flow” (a BPM terminology) is referred to the execution order of the tasks in a process. To fill in this gap, identify the missing tasks and reason the order of task executions in the original AMSDM process flowchart. This leads to an improved control flow definition of the AMSDM process, which will be discussed in Section 4.3.

The second gap in the original AMSDM process definition is that it lacks the precise information about the data associated with the tasks in the process. Even though some data information is available it is mixed with the task names and it is not clear that what are the input and output data for each task. The term “data flow” (a BPM terminology) is referred to the data information associated with each task in the process. To fill in this gap, identify the missing data information. In addition to that identify the various input and output data for each task, which will be explained in Section 4.4. In this section, the data collection, which depends on the various aspects like decision making process, decision objectives, and asset conditions of the AMSDM process, will be explained in Sub-section 4.4.1. The data analysis of the AMSDM process related to the data information that is collected in Section 4.4.1 will be discussed in Sub-section 4.4.2. Finally the data modelling of the AMSDM process based on the collected and analysed information, and it will be discussed in the Section 4.4.3.

The third gap is that the information missing in the original AMSDM process definition in regards to the human resources associated with each task. The information is necessary in answering the following two questions: firstly, what are the various organisation roles
available in the AMSDM process; and secondly, how these roles are allocated to each of the tasks. The term “resource flow” is referred to the various organisation roles that need to be allocated for each task, which will be discussed in Section 4.5. In this section, the organisational structure of roles will be discussed in Section 4.5.1. The allocations mechanism of roles to the tasks will be discussed in Section 4.5.2.

Filling the above three gaps, leads to an ameliorated process definition which specifies the information needed for workflow modelling. Thus it can be used directly for modelling in a workflow language. Emphasise the significance/benefit of this ameliorated process definition: in a narrow sense, it maps from the original conceptual process definition of AMSDM to a detailed comprehensive process definition that is ready for implementation in a workflow/BPM system. In a broad sense, it bridges between the design of a process definition by a domain expert and the deployment of the process using workflow technology.

4.2 Overview of Ameliorated AMSDM Process Definitions

For the original AMSDM process diagrams presented in Chapter 3, the ameliorated AMSDM process diagrams will be presented over here. The basic notations that are used in the ameliorated processes are shown in Figure 4.1. The ameliorated AMSDM process for the original AMSDM process is shown in Figure 4.2. The original AM decision objective identification process is split up in to two processes named ameliorated AM decision objective identification process and ameliorated AM objective identification process, which are shown in Figures 4.3 and 4.4 respectively. The ameliorated Asset health assessment and prediction process and the ameliorated option selection process are shown in Figures 4.5 and 4.6 respectively. More details regarding these ameliorated process definitions will be discussed in Sections 4.3, 4.4, and 4.5.

In all these ameliorated processes, the top-down flow, which is shown with the single arrow mark, is the control flow. The left-right flow, which is shown with the blocked arrow mark, is the data flow. The data information that is showed to the left of the task is the input data and the data information that is showed to the right side of the task is the output data. The person that is showed on the top of the each task is the role who will
execute that particular task. As we explained in Section 3.3 the step wise manner of how the AMSDM process is structured. Hence here in this section we are not presenting any details about the process, only giving the details about the ameliorated figures of the AMSDM process. The explanation regarding these figures will be discussed in the subsequent sections.

![Diagram of Notations for the ameliorated AMSDM process]

**Figure 4.1: Notations for the ameliorated AMSDM process**
Identify an AM decision

Have the decision objectives been identified?

Gather Decision objectives

Have asset conditions been identified?

Gather asset conditions and operation information

Define decision options

Have the decision options been ranked?

Gather decision ranking information

Select the best option

AM decision objectives identification process

Asset health assessment and prediction process

Option ranking process

Best decision option
Do the decision parameters need to be optimised?

Have the relationship between parameters and objectives been analyzed?

Yes

No

Gather the relationship information

Identify the optimal decision parameters

Are all outcomes of the decision known?

Simulation sensitivity analysis process

No

Gather the Outcomes

Are any other options available?

Yes

Final decision

Output the decision

Best decision option

Collect decision parameters for the best option

Decision parameters

Decision maker

Decision maker

Decision user

Decision user

Asset manager

Asset management expert

Managing director

Figure 4.2: Ameliorated AM Split Decision Making (AMSDM) Process
Figure 4.3: Ameliorated AM decision objective identification process
Define the Business objectives
Define and quantify indicators for each business
Define the overall AM objectives
Define and quantify indicators for each AM objective
Audit the AM objectives

- Are the AM objectives in line with Business objectives?
  - Yes
    - END
  - No
    - Modify the attributes of AM objectives

Figure 4.4: Ameliorated AM objective identification process
Figure 4.5: Ameliorated Asset Condition Assessment and Prediction process
Identify Decision options
Review Asset health and cost information
Triggered by failure, person, Technology, cost
Asset Manager
Asset Engineer
Decision Maker/User

Can the decision be made directly with a satisfied confidence level?
Decision Maker/User
Yes: Make a decision directly
No: Look at decision rules

Can the decision be based on rules with a satisfied confidence level?
Decision Maker/User
Yes: Make a decision based on rules
No: Choose an analytical model for decision support

Do fuzzy variables have to be used?
Asset Manager/Engineer
Yes: Make a decision using multistage comprehensive evaluation
No: Do multiple criteria need to be considered?

Reviewed Asset health and cost information
Decision rule matrix
Analytical models
Decision Maker/User
Decision Maker/User
Asset Manager/User
Asset Manager/Engineer
Asset Manager/Engineer
Asset Manager/Engineer
4.3 Control flow definition of the Ameliorated AMSDM process

This section explains the changes made to the control flow definition of the original AMSDM process in Chapter 3. There are major changes and minor changes. Minor changes like renaming task names, changing the execution order. These changes are made to give better understanding about the tasks and also to make task name simple and short.

Major changes are made in order to fill the gaps in the execution order when modelling.
The first major change is in the main AMSDM process of figure 3.3. The change is that, for the task “Select the best option and check the decision parameters” is break up in to two tasks. One task is “Select the best option” and the other task is “Check or Gather the decision parameters for the best option”. This break up has been done because; it is not possible to perform both actions in one task. As these two tasks require different variables with different data types, workflow model will not support to perform two actions in one task. In addition to that decision parameters need to be gathered based on the best option that is selected. Hence we made this change in the process.

The second major change is in the sub process “AM decision objective identification process” shown in Figure 3.4. The change is that, the process itself is split up in two processes. One process is the “Ameliorated AM decision objective identification process” and the other process is the “Ameliorated AM objective identification process” shown in Figures 4.3 and 4.4 respectively. These two processes are linked by making the ameliorated AM objective identification process as the sub process to the ameliorated AM decision objective identification process. This change have been made because, there is no need to execute the newly created sub process each time, but previously, the tasks in this sub process, need to execute each and every time when the flow comes to the AM decision objective identification process.

The third major change is also in the sub process “AM decision objective identification process”. The change is that, added one new task to the process named “Modify the goals” at the decision making task “Are the AM goals inline with the AM objectives” as shown in Figure 4.3. Similarly, in the “ameliorated AM objective identification process” also added one task to the process named “Modify the AM objectives” at the decision making task “Are AM objectives inline with Business objectives” shown in Figure 4.4. These tasks are added because, when the result of the decision is negative, that means there is a need to change the information to make the result of the decision as positive. Hence, we added one more task for both the processes in order to modify the goals and its attributes in the main sub process and to modify the AM objectives and its attributes in the sub process of the main sub process.

For the “Asset condition assessment and prediction process”, there is no change in the control flow definition. Since the flow is quite simple and straight forward. Hence there is
no change in the control flow of the process shown in Figure 4.5 compared with the Figure 3.5. Similarly in the “Option selection process” the control flow definition has not been changed much. The naming of the various tasks has been changed to make them understand, apart from that the flow has not been changed compared with the Option selection process shown in Figure 3.6.

4.4 Data Flow Definition for the Ameliorated AMSDM process

Data collection, data management, and data integration are essential parts of the AM framework that are critical to its success. Timely and accurate data lead to information and form the basis for effective and efficient decision making. Besides the goal of AM is the development of decision-support systems that provide “access to quantitative data on an organisation’s resources and its facilities current and future performance” [14].

Data collection is very much dependant on the intended use of the data. It is obvious that the level of detail and depth needed for the collected data varies according to the hierarchical level of the decisions that need to be made. Although all decision-making levels are undisputedly part of the overall asset management process, data collection requirements have to specifically consider how the collected information is going to be used at the various management decision levels with the different collection methods explained in the literature review. Also as a general recommendation it is noted that the accuracy, quality and currency of the data should be decided based on the cost of data collection and the value and benefit associated with the data. “Data should only be collected if the benefits that they provide outweigh the cost of their collection and maintenance” [44]. Data collection costs can and should be minimised by collecting only the needed data and only when needed. The data collection activities and methods used should be based on and produce results that match the levels of accuracy, precision and resolution requires by the decision processes to be supported.

Once the data have been acquired it should be managed with the various formats and storage media that include paper format, electronic databases, hard disks etc. As asset data have been collected at different times, by different units, using different methods, and stored in varying formats and media, there is naturally a need for data integration.
Data integration is essential to transform the data into useful information, able to support decision making at the various management levels.

4.4.1 Data Collection for the Complete AMSDM Process

For the AMSDM process the collection of various data depends on the different types of asset and the actions that are performed upon the asset. With these two variables the data required for the process will be changing. As the general split decision making process is a high level process, the data collected here is also the high level data, i.e. not the specific detailed data with all the hierarchal levels. In addition to that not the specific data related to any particular application or industry. In this data collection, considered all the variety of major decisions that can be taken on the asset. Hence the data collection job became difficult to produce detailed data. The data which is collected, will give the necessary information about what sort of data the process consists of as well as what kind of data each task requires and that to of which type. In the “Ameliorated AMSDM process” and its sub processes shown in Figures 4.1 to 4.5 the data part is represented in parallelogram. This data diagram is connected to the task by using the double arrow symbol. The representation of data shown in Figures 4.1 to 4.5 will gives the information about what data required for each task and what is the input data and output data for each task.

4.4.1.1 Data Collection for Ameliorated AMSDM Process

The data collection for the ameliorated AMSDM process is as follows:

**Identify an AM decision**

The activity of this task is to take a decision upon the asset what to do. For taking the decision, the information needed is classified into three types namely Asset Type, Action or Activity on Asset, and Time duration. The asset type is mainly the type of the asset that need to make a decision, i.e. whether building, pump, machine, computer, aircraft, etc. The next one that need to consider is the action that need to be taken on the selected asset i.e. replace, repair, maintenance, fixing, etc. The next one that need to consider is the duration of time i.e. when should the selected action need to complete upon the asset whether in a day, a month, a year, etc. The input for this task is nothing and the output for this task is the AM decision with three variables Asset type, Action on asset, Time duration.
Have the decision objectives and constraints are identified
As this task is decision making task, the control flow of the process will be changed based upon the decision of this task. When the decision is true the flow will be proceed to the task “Decision objectives information from database or manual” else the flow will proceed to the “AM decision objective identification process” The input for this task is the AM decision which is the output of the previous task and the output is the decision whether it is true or false. The data collection for the sub process will taken place separately.

Decision objectives information from database or manual
The activity of this task is to provide the objectives for the AM decision. Actually for this task the data should be retrieved from the database depending on the type of the asset and action that is to be performed on the asset and in what time or the data that comes from the “AM decision objective identification process”. But at present, as there is no integration with the database, so the objectives will be entered manually. The input data of the task is the AM decision and the output data of the task is the various objectives of the decision. The various objectives are reliability, capacity, safety, cost, technology, profit, time, energy consumption, etc.

Have asset conditions been identified
Similar to the task one before, this task is also the decision making task. When the decision is true the flow will flow to the task “Gather asset condition and operation information” else the flow will proceed to the sub process “Asset health assessment and prediction process”. This sub process is already a developed tool, where we need to just trigger the tool, when the flow preceded to this sub process. The input of this task is the decision objectives from the previous task and the output will be the result of the decision either true or false.

Gather asset condition and operation information.
Similar to the task “Decision objectives information from database or manual” this task is also need to gather the data of asset conditions from the database, provided if the model integrates with the database. As there is no integration with database the data for the conditions of the asset will be entered manually. The input data of the task is the decision objectives and the output data is the various asset conditions and operational information.
The various asset conditions are deteriorating condition, insufficient capacity, performance failure, etc and the operational information like the energy, environment, bounded limits, etc.

Define decision options
The activity of this task is to provide the various options for the decision by leading the asset conditions and the decision objectives into consideration. For example the various options are replace the asset completely, replace some part of the asset and maintain the remaining part, repair the whole asset, refurbish, change the work load, extend the asset life, etc. The input data of this task are the asset conditions and the decision objectives and the output data of the task are the various options that need to be taken into account.

Have the decision options been ranked
As this task is the decision making task, it will control the flow of the process. When the result of the decision is true the flow will lead to the task “Gather option ranking information”, to gather the different ranks for various options defined in previous task. Else the flow will lead to the process “option ranking process”, to obtain ranking for these options. The various decision options are to be ranked in order to give the priority for the options to decide which decision options will be desirable for the process to proceed. The input data of this task are the various decision options and the output data is the result of the decision whether true or false.

Gather option ranking information
The activity of this task is to gather the different ranks of various options from previous task that are obtained either from the database or from the “Option ranking process”. The input data of this task is the various options and the output data is the ranking of the various options.

Select the best option
The activity of this task is to select the best option among the various ranked options in previous task. The input data for the task are the various ranked options from previous task and the output data is the best option that needs to be selected.
Enter or gather the decision parameters for the best option

The activity of this task is to gather the various decision parameters for the best option that is selected. The various parameters are the cost, profit, time, etc. The input data for this task is the best option which is the output for the previous task and the output data is the various parameters.

Do the decision parameters need to be optimised

As this is the decision making task, the result of the decision will be the true or false. At present, we are considering it as only false because, from the Section 3.2.2, it can be seen that discussions are going on for selecting the best approach for optimization. Hence the result of the decision will be taken as false and the flow will proceed to the task “Are all outcomes of the decision known”. The input data of the task is the decision parameters and the output data is the result of the decision i.e. false.

Gather the outcomes

The activity of this task is to gather the various outcomes of the decision, so, that will give the idea about the decision whether the decision made is the effective one or not. When the process is integrated with the database then the outcomes will be automatically retrieved from the database or from the “Simulation sensitivity analysis process”. Hence the input and output data for the task are the various outcomes of the decision.

Is the decision satisfied?

As this task is the decision making task, it will take the input data as the various outcomes of the decision and delivers the output as the result whether true or false. If the decision is satisfied the flow will proceed to output the decision else the flow will proceed to the task “Are any other options are available” to check whether any options are available or else change the objectives of the decision. This task is the main loop for the process. This loop process is repeated until the decision is satisfied.

Output the decision

The activity of this task is to provide the final output of the process, i.e. the AM decision, which satisfies all the decision making tasks. The output of this task is the best option which was selected. The decision which we gave initially at the starting of the process will go through all these steps and provide us with the best decision option.
4.4.1.2 Data Collection for AM Decision Objective Identification Sub process

The flow will proceed to this process, if the decision objectives or goals are not identified and need to identify these decision objectives or goals. The input data for this process is to trigger it when the decision objectives are not identified. The output data is to produce the set of decision objectives and its constraints. The detailed discussions on the collection of the data for each task are as follows:

**Define the goals for the specific AM activity**

The activity of this task is to define the various goals for the asset, based on the asset type. The input data for the task is the AM activity that the decision is involved. The output data are the various goals for the AM activity like increase reliability, increase capacity, reduces cost, increase profit, etc.

**Define and quantify attributes for each goal**

The activity of this task is to define the various attributes for each goal that were defined in the previous task. The input data of the task are the various goals. The output data are the various attributes that are to be defined for each goal. The identified attributes for the various goals are shown in Table 4.1.

<table>
<thead>
<tr>
<th>Goals</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>Predictability, Maintainability, Proven</td>
</tr>
<tr>
<td>Capacity</td>
<td>Length, Volume</td>
</tr>
<tr>
<td>Cost</td>
<td>Dollars, Percentage</td>
</tr>
<tr>
<td>Energy Consumption</td>
<td>Volume, Capacity, Technology</td>
</tr>
<tr>
<td>Profit</td>
<td>Dollars, Value, Percentage</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Volume, Technology, Cost</td>
</tr>
</tbody>
</table>

**Table 4.1: Goals and attributes for the specific AM activity**
Audit the Specific AM activity goals
The activity of this task is to inspect the various goals and see whether these goals are satisfied for the AM activity or not. The input data for this task are the various goals and its attributes. The output data are also the various goals and attributes, but these output goals and attributes are the satisfied goals by the audit members.

Modify the goals
The activity of this task is to modify the goals and its attributes that are not in inline with the AM objectives. The input data for this task are the goals and attributes that need to be modified. The output data are the modified goals and attributes, which are in inline with the AM objectives.

Defined decision objectives and constraints
The activity of this task is to provide the final goals and attributes for the specific AM activity. These final goals and attributes are called as the decision objectives in this task. The input data are the finalised goals and attributes which are inline with the AM objectives. The output is to link back with the main process and supply the whole data information that is available from this whole sub process to the main process as the input for the task at which this sub process will be joined to the main process.

Similarly, the “AM objective identification process” shown in Figure 4.4 which is the sub process of the “AM decision objective identification process” will follow the same steps for defining the business objectives and its attributes, and then defining the AM objectives and its attributes, and check whether these two objectives are in inline with each other. The flow will proceed through all these steps and produce the output as the various AM objectives that are in line with business objectives.

4.4.1.3 Data Collection for the Ameliorated Option Selection Sub process
The flow of execution will reach this process, if there is a need to identify options for the decision, and also when these decision options need to rank. The input data for this process will be triggering it, when need to identify the decision options and rank them. The output data will be the number of decision options with ranking. The detailed discussions on the data collection of this process are as follows:
Identify decision options

The activity of this task is to provide the various decision options to the “option selection process”, defined in main process. The various decision options are like do nothing, fix the problems, Recover the asset, Upgrade the asset, Change work load, Renew the asset, etc. The input data will be these decision options and the output is to display these decision options on the screen.

Review asset health and cost information

The activity of this task is to review the asset conditions and its health. In addition to that review the cost information as well. The asset conditions and its health will be gathered from the task of main process “Gather asset conditions and operation information”. Asset conditions and its health are predicted by the Asset specifications, Asset degradation profiles, Asset operation information. The cost information needs to provide here by considering the various decision options. The various costs are like Repair cost, Replacement cost, maintenance cost, production loss, etc.

Can the decision be made with satisfied confidence level?

As this is a decision making task, we need to make a decision up on this task. Based on the available information like decision options, asset conditions and its health, and cost information, is it possible to make a decision? If yes, then the process designer can directly provide the rank for the various decision options. This ranking is to decide which decision option is the best one.

Look at decision rules

Decision rules are a set of verbal equivalent of a graphical decision tree, which specifies class membership based on a hierarchical sequence of (contingent) decisions. These decision rules are yet to be defined in our process (Section 3.2). Hence the data for this task is yet to be supplied.

Choose an analytical model

The activity of this task is to choose an analytical model for decision support. There are several analytical models, which can support the decisions. For example Cost benefit analysis, Decision tree, AHP, Multistage comprehensive fuzzy evaluation, etc. Among
these analytical models, process designer need to select one analytical model to support
the decision. Hence the input will be these analytical models and the output will be to
select one analytical model and display on screen. Based on the model selected, the
corresponding tasks of the process will be decided.

For the remaining tasks of the process, the data collection is a bit difficult, because these
all tasks are based on the selected analytical model. Hence the data also is based on these
models. From the Section 3.2, we can observe that still research is going on to select the
best analytical model. Hence the data collection for these tasks is done through some
guidelines, i.e. we provided the option on the output screen for the asset management
expert of an organisation to provide some guidelines or prototypes on which basis to
choose these analytical models, which model to choose, and which is the best one. With
these guidelines the user can choose the suitable analytical model. More details
regarding this process will be provided in the data analysis section.

With all this information, the data collection of the Ameliorated AMSDM process has
been completed successfully. The data that is collected will be incorporated in to the
model while modelling the Ameliorated AMSDM process. The modelling of data will be
discussed in the next chapter. Now proceed to analyse the collected data, and it will be
explained in separate section. The data analysis is mainly performed on those tasks or
processes, which are not covered in this section.

4.4.2 Data Analysis for the AMSDM Process

Data analysis is a process of gathering, modelling, and transforming data with the goal of
highlighting useful information, suggesting conclusions, and support decision making
(Wikipedia). The data analysis is to be done, when the distribution of a variable is not
normal, the data may need to be transformed or categorised. Furthermore, a decision
should be made on how to handle missing data. Also in many cases, a check to see
whether the randomisation procedure has worked will be the starting point for analysing
the implementation of the design. This can be done by checking the variables. Hence for
these purposes the data analysis has to be performed.
The data analysis for the AMSDM process is as follows:

For the task “identify an AM decision”, simply the string variable is not sufficient, it should contain more information apart from the string variable. From the data analysis the string variable of the AM decision is categorised and divided into three variables namely Asset type, Activity on asset, time duration. The Asset type variable is to know about on which type of asset the decision is to be made i.e. building, pump, engine, etc. The Activity on asset variable is to know about the action that is to be performed on the selected asset type i.e. replace, repair, maintenance, etc. The third variable, Time duration is to know about the time that is taken for performing the selected action on the asset i.e. week, month, year, etc. due to this analysis on the AM decision, it gives more clarity. And also while modelling; there is a chance to use these AM decision variables individually when required. Due to this there is no need to enter the AM decision each time. The AM decision with all these information can be used for each task in the process by giving it as the input data for the task and no need to re-enter it each time.

Apart from this task, the data for remaining tasks are well analysed and collected in previous section. Hence we are not discussing the remaining tasks over here.

**Asset Health Assessment and Prediction Process**

This sub process is not covered in the previous data collection section. The reason for this is that, after deeply analysing the ameliorated AMSDM process, we came to conclusion that there is no need of modelling this sub process. This is because

1) As most of the tasks in the sub process are performed by one single role, so there will be no coordination of roles between these tasks. But workflow model is used mainly for coordination of people.

2) There is a pre existing tool for this sub process, which can be triggered directly from the main process.

3) The tasks in the process “identify failure development curves”, “Assess asset current condition”, and “Predict failure probability” are different tools to determine the asset conditions.

4) The data generated in this process is of graphs, curves, functions, etc, which can’t be captured by the YAWL language.
Due to these reasons, we decided not to model this sub process. If our workflow model triggers these tools, then that will be enough to gather the required information and transfer it to main process.

**Option Selection Process**

The flow will proceed to the sub process “Option selection process”, when there is a need for identifying the various options for the AM decision. In addition to that, this process will also be triggered when the identified decision options need to be ranked. The final outputs of the process are the various decision options that are ranked. This process is still in the formulating stage, where some of the tasks still need to be developed and some of the developed tasks need to be improved (Section 3.2). The tasks that need to be developed are “Decision rule matrix”, “Analytical models”. The application of workflow model for this sub process is only up to certain level of tasks. This is because; there will be no coordination of roles between various tasks. The tasks that are after the task “Choose and analytical model” are the steps to select the various analytical models. Depending upon the certain decisions, the suitable analytical model will be selected. These decisions need to be made by single role. As there is no coordination of roles, we thought of that no need of workflow model. However these models are either pre developed or need to develop in future. Hence, if we just trigger these models and at the end transfer the generated information back to the main process, then that will be enough to make the decision.

4.5 Initial Resource Modelling of the AMSDM Process

“A Resource is an entity that is assigned to a workflow activity and is requested at runtime to perform work in order to complete the objective of this activity” (Michael zur Muehlen 2004) [98].

“A resource model contains the definition of human and technical resources that are involved in the execution of a workflow model as workflow participants” (Michael zur Muehlen 2004) [104].
Workflow management systems coordinate tasks (or activities), resources and data according to the formal representation of the process logic. These can help realize efficiency potentials through the elimination of transport and wait times between process activities, and provide a detailed level of control over the assignment of work to process participants.

A resource is classified as either human or non-human i.e. a resource that does not correspond to an actual person – e.g. plant, equipment, system, etc. A human resource is typically a member of an organisation. An organisation is a formal grouping of resources that undertake work items pertaining to a common set of business objectives, where work item is a feature of the System back end workflow process that provides the assignment of tasks to the users of the organisations that are participating in the trade transaction. Human resources usually have a specific position within that organisation and in general, most organisational characteristics that resources possess relate to the position(s) that they occupy rather than directly to the resource themselves. A resource may have one or more associated roles. Roles serve as another grouping mechanism for human resources with similar job roles or responsibility levels e.g. Asset managers, users, decision makers, union delegates, etc. Individual resources may also possess capabilities or attributes that further clarify their suitability for various kinds of work items.

Non-human resources may be durable or consumable in nature. A durable resource is one whose capacity to undertake work is unaffected by the amount of work that it has undertaken, whereas a consumable resource is one that is consumed in the act of completing a work item. There is usually a rule of consumption or capacity with consumable resources indicating how much work they can actually undertake before being depleted and requiring further replenishment. Each resource may have a schedule and history associated with them A schedule is a list of work items that a resource is committed to undertaking at a specified future times where as a history or work log is a list of work items that a resource has completed at some time in the past.

4.5.1 Organisational Structure

The representation of organisational structures has become an increasingly important part of information system design. Beyond the traditional functions of accounting or human
resources management, collaborative application systems such as groupware or workflow management systems rely on a division between the business processes, i.e. the temporal and logical order of those activities that are necessary to process a business object, and the structure of the resources that perform these activities.

**Two Strategies for Resource Modelling**

During the modelling of resources for a workflow application two approaches can be distinguished [98, 104]:

1. **Technology-driven Approach**

The technology-driven approach to resource modelling presumes no predefined set of resources in the organisation. Instead, the structure of entity types needed in the workflow model is derived from the specification of workflow model itself. Typically, roles derived from an existing workflow model are of the form authorized to perform activity X. The use of roles instead of particular resource makes the workflow model independent of changes in the organisational population. However, changes in the workflow model can easily affect the resource model, because newly defined activities require new roles to be defined.

2. **Organisation-driven Approach**

The organisation-driven approach can be found in environments where workflow management systems are introduced into larger organizations and a formally defined organization structure of the enterprise already exists. In this case, the current organisational structure has to be depicted in the workflow management system, either by direct modelling or by referral to an external resource repository, such as a human resource module of ERP software. The advantage of this approach is the identity of the organisational structure within the workflow models and real organisations. If workflow models are to be modified by domain experts as opposed to system administrators, it is easier for them to relate to the organisational model depicted in the system.

Among these two strategies, we are using the technology-driven approach for the ameliorated AMSDM process. This approach has been chosen because; at present the resource modelling for this process is in the initial stage, and the workflow model which
we are going to develop in the next chapter is the generic process model and is not related to any specific organisation. Hence there is no specific organization structure for this process. Therefore we are using the technology-driven approach, which need not presumes any predefined set of resources in the organization.

For the AMSDM process model, the various roles have been identified for executing each task. Depending on the organisation or company these roles may differ. Every company has its own preference and may have specific structure in allocating the roles for the tasks. Hence, in this process our task is to identify the possible various roles in general. The various roles that are identified for the ameliorated AMSDM process are Asset Manager, who will look after the asset and takes certain decisions upon the asset Decision maker, who makes immediate decision upon the part of the asset. System, which makes decision without any human intervention Admin user, who uses the process and gives certain inputs to the process Analyst, who will provide or cross check the information needed for the asset. Similarly we identified some other possible roles like Share holders, Audit persons, General Manager, user, process designer, operation manager, etc which are shown in Figures 4.1 to 4.5.

4.5.2 Role-based Task Allocation

The majority of research on role-based access is directed towards the development of an efficient, flexible and secure access mechanism that abstracts from the single user and uses abstract roles as a grouping of either users or permissions to administer the organisational population that needs access to an information system. The role-based task allocations are based on the assignment concepts which are explained as follows [98,104]:

Assignment concepts for Resources

A dynamic perspective on the resources involved in workflow execution is the handling of the resource assignments at runtime. The most common mechanism for this is a hierarchical assignment through the workflow management system, which places work-items on the work lists of qualified resources. A resource can accept, reject, delegate or postpone the execution of a work item (depending on the services provided by the workflow management system).

120
With regard to the assignment of resources to tasks, three different concepts can be distinguished:

**Direct Designation**

In this case an activity or task is assigned to one or more entities on the resource model directly. At run-time, the workflow engine can directly look-up these resources in the resource repository and place the relevant work-items on their work lists. This kind of assignment is easy to handle for the workflow administrator, because s/he is concerned with a single entity type: the workflow performer. If an activity is to be made available to a group of people, all members of the group have to be assigned to the workflow activity one by one. The direct assignment concept provides no independence of workflow model and organisational model.

**Assignment by Role**

Most workflow management systems provide workflow modellers with a role entity type. Within this domain, one role entity is used as synonym for one or more resource entities. The main purpose of role model is the separation of workflow and resource model, where changes of the organisational population do not affect the workflow model directly. The use of roles instead of a direct assignment also provides means of indirect workload balancing, because all members of a qualified role are notified about the pending work-item, but only one member of this group needs to perform the activity. From a technical point of view, the workflow management system has to perform a resolution process to determine the members of a role before it can notify these resources. Therefore, an error handling procedure has to be implemented, if this resolution process returns empty set of resources.

**Assignment by Formal Expression**

The most complex form of activity assignment uses a formal expression. In this case, not only the entity types of the resource model have to be known to the workflow modeller but also the relationship between these entity types and possible functions depending on the workflow execution history. The attributes used in such a formal expression can either be dependent on the workflow instance, such as the information about the performer of the last activity, as well as independent on the workflow instance, such as the relationship of a resource to another resource. If a formal expression is used for
activity assignment an error handling similar to the one used during the assignment by role has to be implemented, in case the expression returns an empty set of resources. Formal expressions may not only relate to the relationships between entities type of the resource model but also to specific attributes of the resources.

Among the three assignment concepts, the assignment by role concept will be using for modelling the ameliorated AMSDM process definitions. This concept is chosen because; it is in the initial stage of resource modelling, where the resource model and workflow model are separated. Due to this, the organisation structure will not be affected. As our process is generic process, it will be applied to several organisations, which has individual structure. Generic process will be useful only if it can be applied to multiple organisations. Hence direct designation is not applicable to our process. Coming to formal expressions concept, this concept needs to have the details regarding the entities and functions. In the initial stages these details can’t be analysed, so the assignment by role concept is applied. The more details regarding this concept can be provided in the next chapter under the AMSDM process resource modelling Section 5.1.3.

These are the various reasons to use the role-based task allocation for modelling the resources in ameliorated AMSDM process shown in Figures 4.1 to 4.5.

4.6 Summary

In this chapter, identified several gaps existed in the original AMSDM process. These gaps were rectified and developed new process definitions for the AMSDM process which leads to ameliorated AMSDM process definitions. The first gap lies in the order of performing tasks in the process. To fill in this gap, identified missing tasks ad modified the order of task executions in the original AMSDM process flow chart. The second gap is that it lacks the precise information about the data associated with the tasks in the process. To fill in this gap, the missing data information was identified and also the various input and output data for each task. The third gap is that the information missing in regard to the resources associated with each task. To fill in this gap, identified the various possible roles associated with each task. Filling the above three gaps, leads to an ameliorated process definition which specifies the information needed for workflow
modelling. Thus it can be used directly for modelling in a workflow language. In a broad sense, it bridges between the design of a process definition by a domain expert and the deployment of the process using BPM technology. The filling of these gaps is one of the main contributions in this proposed research.
CHAPTER 5

MODELLING AND EXECUTION OF THE AMSDM PROCESS

USING YAWL

The Ameliorated AMSDM process definitions developed in Chapter 4 specifies the information needed for workflow modelling. Based on these ameliorated process definitions, the AMSDM process is modelled using the YAWL workflow modelling technique. The Section 5.1 discusses the development of the AMSDM workflow model, comprising control flow definition in Section 5.1.1, data flow definition in Section 5.1.2, resource modelling in Section 5.1.3. The subsequent Section 5.2 discusses about the execution of the AMSDM process. The final Section 5.3 gives the summary of the overall chapter.

5.1 A Workflow Model of the AMSDM Process

The benefit of applying YAWL workflow language to the AMSDM process is to develop a workflow model for a high level asset management process. Through this workflow model, one can control the flow of the process; the necessary data can be imparted into the model, and each task can be allocated to various roles. In addition to that there is a possibility to automate the process and also it can accommodate flexibility and enable both ad-hoc and evolutionary changes. Applying the existing technique (YAWL language) to the AM process (AMSDM process) is completely new area of research. Hence this is one of the main contributions in this proposed research. As mentioned in Chapter 1 that applying existing technique to the new process is a totally new concept. Hence trying to model the high level AM decision making process with the WFM technique gives main contribution towards both areas. The main goal of this AMSDM workflow model is to furnish the best AM decision as an output by satisfying all possible characteristics of the AMSDM process as explained in Section 3.3.1. This model can also justify whether the selected AM decision is good one or not. This justification will be done through several analysis sub processes existed in the AMSDM process.

The modelling of the process is based on the ameliorated AMSDM process definitions developed in Chapter 4. These process definitions are the starting point for modelling the
AMSDM process. The complete information required for modelling the process in terms of all perspectives was provided in Chapter 4. Based on this information, try to apply the YAWL language to the AMSDM process to originate the AMSDM workflow model.

For modelling a process with the YAWL language, there is a procedure to follow:

1. First step, need to model the control flow of the process, which describes the manner in which the thread of execution flows through the process model.
2. Second step, need to model the data flow of the process, which aims to capture the data characteristics that occur repeatedly in the process.
3. Third step, need to model the resource part of the process, which aims to capture the various ways resources are represented and utilized.

These three steps are the modelling steps for the process. After successfully completing the modelling of the process, the fourth and final step is to execute the process model to reveal the outcome of the model. These steps will be explained one by one in the following sub sections.

The development of the AMSDM workflow model is discussed in the following sub-sections starting with the control-flow definition of the process.

**5.1.1 AMSDM Process Control Flow Definition**

In this section, gradually refine the sequence of tasks of an ameliorated AMSDM process definition to an executable process model using YAWL. The control flow for the AMSDM workflow model is developed based on the changes and improvements that were made in the ameliorated AMSDM process of Section 4.3. First concentrate on the control flow and only informatively mention the data necessary for and resulting from the performance of each task. In addition to that, explained each task in comparing with the characteristics of the AMSDM process (see Section 3.3.1). And then, in detail, go through the definition of the data and its flow through the process in Section 5.1.2. The Control flow definition will be explained in the step wise manner by considering the high level task names, which were shown with the pink colour in Section 3.3. From this Section 3.3, only grab the high level task names, but the complete model is developed based on Section 4.3.
Here, explanations are given for the basic notations in YAWL language to model the control flow definition of the ameliorated AMSDM process. From Section 3.5, a workflow specification in YAWL is a set of extended workflow nets (EWF-nets) which form a hierarchy, i.e. a tree like structure. Tasks are either atomic tasks or composite tasks. Each composite task refers to a unique EWF-net at a lower level in the hierarchy. Atomic tasks form the leaves of the tree like structure. There is one EWF-net is without a composite task referring to it. This EWF-net is named the top level workflow and forms the root of the tree like structure. Each EWF-net consists of tasks (either composite or atomic) and conditions which can be interpreted as places. Each EWF-net has one unique input condition and one unique output condition. In contrast to Petri nets, it is possible to connect “transition-like objects” like composite and atomic tasks directly to each other without using a “place-like object” (i.e., conditions) in-between. For the semantics this construct can be interpreted as a hidden condition, i.e., an implicit condition is added for every direct connection. The split tasks and join tasks are used to trigger the outgoing flows and incoming flows respectively. Figure 5.1 shows the various symbols used in AMSDM workflow model.

![Symbols used in YAWL](image)

**Figure 5.1: Symbols used in YAWL [85]**
As mentioned, the control flow definition of the ameliorated AMSDM process is modelled in a step-wise manner to obtain the AMSDM workflow model. These various steps are as follows:

**Step 1: Identify an AM decision**

Now look into each task in detail. An ameliorated AMSDM process of Chapter 4 will be started, when an organization or industry wants to make a certain decision on the AM activity. Hence the process starts with an atomic task named “Identify an AM decision”, which includes information about the asset name, asset location, asset information, decision maker, activity of the asset, time duration, etc. These variables are captured in our model through an “Identify an AM decision task”. The output from it is currently simplified to only contain three variables *Asset name, Activity on Asset, Time duration.*

Figure 5.2 shows the task with the output variables. This task is related to the, information characteristic of Section 3.3.3. This task gives the knowledge about the decision, the scope and limitations of the decision, the clarification of goals, the probability of each option, etc. The basic information needed regarding the decision is given by this task.

![Diagram of Identify an AM decision task](image)

**Figure 5.2: “Identify an AM decision” task**

**Step 2: Identify decision objectives and constraints**

Once a decision is obtained from previous step, then the objectives for this decision need to be identified. These decision objectives need to be gathered either from the database or from the AM decision objectives identification process or to be entered manually. As from one task, splitting the flows in to two flows and in which either one of the flow will be true. The capturing of this behaviour is done through split task (i.e. flow relation coming out from single task to multiple tasks), this task is also refined to an XOR split task and is applied for “Check decision objectives task”. The XOR join is applied to the “Gather decision objectives information task”, so that any one of the two flows will be
accepted. The composite task “AM decision objective identification process” will be explained in the Appendix A. For this composite task, there is an instance of the YAWL net which describes the structure of the (sub) process in detail in terms of the tasks that it comprises and the sequence in which they occur. The result from the execution of all these tasks is the collection of various Decision objectives (Figure 5.3). This task is related to the characteristic, Goals. Here users will mention what goals they accomplish? What should they do? What goals they need to choose? We need to ask ourselves these questions first and set the goals and objectives for the decision before moving further. If we don’t have these goals and objectives as predefined set, then we must precede the flow to the sub process, “AM decision objectives identification process”. In this sub process, define all the decision goals and AM objectives and set back these goals and objectives to the main process.

Figure 5.3: Control flow for “Identify decision objectives and constraints” task.

The flow details of which task need to be selected is given by the XPath expression /AM_split-decision-making-process/Check_Decobjectives/text()='true'. Figure 5.4 shows the flow details and the XPath Expression used for controlling the flow. If the XPath expression is defined for top flows then the bottom most flow will be automatically selected, provided that if the XPath expression for the top flows is false. Similarly, the flow details for all the remaining decision making tasks will be applied.
Step 3: Identify asset conditions
Similar to the previous task, the asset conditions are gathered from the database or from the Asset health assessment and prediction process or entered manually. The asset conditions will be gathered at the task “Gather asset conditions and operation information”. All this behaviour is captured through the XOR split and joins tasks. The Asset health assessment and prediction process is already a pre existing tool. Hence we are not modelling this sub process. This can be shown in the complete control flow of the process of Figure 5.8. The results from execution of this task are the collection of asset conditions and its operation information. Figure 5.5 shows the control flow definition of ameliorated AMSDM process for asset conditions. This task is related to the characteristics, Information and Value. As in the Information characteristic, the scope and limitations of the decision, the effects of options are based on these asset conditions and operation information. Whereas the Value characteristic, refers that the outcome will be affected with these asset conditions.
**Step 4: Define decision options**

This atomic task is to define the various options for the decision. This task will be executed after gathering the necessary information about the decision objectives and asset conditions. Based on this information the decision options need to be defined. The options may be on the asset, or the activity that is performing on the asset, or the time taken to successfully complete the decision. The output from it is currently simplified to only contain Decision options. Figure 5.6 shows the control flow for the decision options by taking set of tasks in to one individual task. This is the most important task in every decision making process. This task will act as an iteration task for the complete process. At present, it is a normal atomic task, but later on when we go into details it will turn in to XOR join task. The other flow will be connected from the atomic task “Are any options available”. This task comprises the characteristics as Information, Alternatives, criteria, and Value. The information characteristic is related with this task as per the effects of various options and the probability of each option. With this information, we can get the basic idea of which option needs to be selected. The Alternatives characteristic, gives the various options that need to be defined, where as the Value characteristic, refers to how desirable each outcome will be. This also gives the clarity for the decision options that whether the decision must be in dollars, satisfaction, capacity, or other benefits. The Criteria characteristic is the requirement that each option must possess to a greater or lesser extent. Usually decision options are rated on how well they possess each criterion.

**Figure 5.6: Define decision options task**

**Step 5: Select a decision option**

Among the various decision options, we want to select the best option based on the rankings of the decision options. For this there is a process named Option ranking process to rank the options, subjected to if the options are not ranked. For ranking of options the flow is transferred through the XOR split and Join tasks. After ranking is finished, we can
know the best decision option and the decision parameters for that best option. The outputs from these tasks are decision option ranking information, best option, and decision parameters (see Figure 5.7). This task is related to the characteristics Criteria, Preferences. In the previous task we explained that on how well each criterion is possessed, based on that the ranking of each decision options will be given. Hence for this task, criterion is the major characteristic. Coming to Preferences, depending on the decision maker each preference will be different, so the selection of best option might change depending upon the preferences that he made.

Figure 5.7: Control flow for the Select a decision option task

For the remaining tasks, the control flow is similar to the previous tasks with the XOR splits and joins tasks. However the data flow and the variables used for these remaining tasks are different. As this data flow part will be explained in the next Section 5.1.2, so here we directly going to show the complete control flow of the ameliorated AMSDM process. Most of the remaining tasks are the decision making tasks for controlling the flow i.e. If the outcome of the decision condition is true, then the flow will proceed to one task otherwise the flow will proceed to another task. As we explained these sorts of tasks in before steps, hence we are directly going to explain the complete control flow of the ameliorated AMSDM process excluding the sub processes. The sub processes will be explained in the Appendix part. Figure 5.8 shows the complete control flow of the AMSDM process. This figure will give all the necessary information regarding the control flow of the AMSDM workflow model.
Figure 5.8: Complete control flow of the AMSDM Workflow Model
Consideration of Task Automation

All the atomic tasks in YAWL can have decomposition defined as manual or automated. A task with a manual decomposition is a task that is intended to be executed by a human resource, e.g. a participant in the organisational model. The task with an automated decomposition is a task that is not offered to any resource but is executed by the system. This type of task can be used to manipulate the content of simple data assignments to complex reports generation. Alternatively it may be associated with a codelet, a discrete piece of code that is executed for mapping between the input parameters and output parameters.

When modelling the control flow of the process, we need to have basic understanding regarding the decomposition that need to execute each task i.e. manual or automated. At present, in this AMSDM workflow model every task will be executed by the manual decomposition, but there are some tasks that can also be executed by the system. The decision making tasks for controlling the flow of the model with the XOR-Splits can also execute by the system. These decision-making tasks can be executed by the system, provided that if the model is having interface with the database. At present as there is no interface with the database, we are using the manual decomposition. In future, once the model has the interface with the database these decision-making tasks can be automated. Figure 5.9 depicts the tasks that can be automated.
Figure 5.9: Manual and Automated tasks in AMSDM Workflow Model
With all these steps the control flow definition for modelling the ameliorated AMSDM process has been successfully completed. Due to this control flow definition, described the structural characteristics of the ASMDM process and the manner in which the thread of execution flows through the AMSDM workflow model. Here the contribution towards this research is that to organise the flow of execution of each task.

**5.1.2 AMSDM Process Data Flow Modelling**

After successfully modelling the control flow definition for the ameliorated AMSDM process, the next step is to model the data flow for the process.

The objective of the AMSDM process is the handling of the various decisions in different organisations. The question arises whether a workflow management system is really needed. However, the complexity of the process related data (i.e. large amount of data is required to know the information regarding the decision objectives, asset conditions, and decision options, etc. In addition to that, this data needs to be handled in a structured, systematic, and a just way) justifies the use of workflow system. The data flow modelling is yet another main contribution in this research. For this data modelling, we collected some data and it was explained in Section 4.4. Based on the data that was collected and analysed for the ameliorated AMSDM process in Section 4.4, the data flow modelling is discussed to obtain the AMSDM workflow model. The data collected in Section 4.4 will be imparted into the model through various variables. The data flow modelling is explained in two sub- sections. Data definitions explained in Section 5.1.2.1 and data mapping explained in Section 5.1.2.2.

**5.1.2.1 Data Definitions**

This data definition section consists of two definitions. They are data structures and data types, which are explained as follows:

**Data Structure and Data types**

The data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently. Data structure is used in almost every program or software system. Specific data structures are essential ingredients of many efficient algorithms, and make possible the management of huge amounts of data, such as databases and
internet indexing services. Next coming to data type, the data type is defined as the data storage format that contains a specific type or range of values. When computer programs store data in variables, each variable must be assigned to specific data type. There are two different data types. They are built in data types, which are already existed in the programming language, second type are user defined data types or complex data types, which are not pre existed in the language, user will define these data types according to the requirements.

In YAWL by default, a number of simple XML Schema data types are defined for variable definition, but if we need more complex data types, we will define our own XML Schema definition to describe them. To capture the complex data of this domain in YAWL the following data type definitions are made for modelling the ameliorated AMSDM process.

- **AM Decision Type**: - This complex type, comprises of data related to the AM decision, i.e. Asset Name, Activity on Asset, Time Period. For the moment is minimized to only contain these three variables, but easily can extended to contain information such as asset model no, organisation name, asset type, asset part, asset information, asset life, etc. The code for defining the AM Decision Type in XML Schema is as shown below.

  ```xml
  <xs:complexType name="AMDecisionType">
    <xs:sequence>
      <xs:element name="AssetName" type="xs:string"/>
      <xs:element name="ActivityOnAsset" type="xs:string"/>
      <xs:element name="TimePeriod" type="DateType"/>
    </xs:sequence>
  </xs:complexType>
  ```

- **Date Type**: - This complex type comprises the Start_date and End_date of the decision, i.e. the starting date of the work on which the decision is made and the ending date on which the started work will be completed. The XML Schema definition for this complex type is as follows:
Decision objectives List Type: - comprising the decision objectives related data, i.e. reliability, energy consumption, efficiency, technology, cost, etc. these objectives are minimised to only contain the name of the objective, which is of string type. But these need to be further extended for defining the attributes of each objective. These attributes are already identified in section 4.4.1, but not implemented in the model due to the lack of user interfacing. The various decision objectives that comes under this complex type are as follows:

Asset Conditions List Type: - comprising the information related to the condition of the asset, i.e. performance failure, erosion, structural deficiency, deteriorating
condition, insufficient capacity, etc. Based on all these variables, the asset conditions are identified. The code for this complex data type is as follows:

```xml
<xs:complexType name="AssetConditionsListType">
  <xs:choice>
    <xs:element name="PerformanceFailure" type="xs:string" />
    <xs:element name="Erosion" type="xs:string" />
    <xs:element name="StructuralDeficiency" type="xs:string" />
    <xs:element name="DeterioratingCondition" type="xs:string" />
    <xs:element name="InsufficientCapacity" type="CapacityType" />
  </xs:choice>
</xs:complexType>
```

```xml
<xs:complexType name="CapacityType">
  <xs:sequence>
    <xs:element name="volume" type="xs:int" />
    <xs:element name="length" type="xs:int" />
  </xs:sequence>
</xs:complexType>
```

Operational Information Type: - Comprising the information related to asset operating condition like energy utilising, environmental conditions, limitations, etc. we identified these three operational information. There can be more operating conditions. The code for these operational information is as follows:

```xml
<xs:complexType name="operatingInformationType">
  <xs:sequence>
    <xs:element name="EnergyUtilising" type="xs:string" />
    <xs:element name="EnvironmentConditions" type="xs:string" />
    <xs:element name="Limitations" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```
- **Asset Health Type**: This complex type Comprises of information related to the asset health, i.e. asset specifications, asset degradation profiles, asset operation information, repair cost, replacement cost, etc. The code of XML Schema for this complex type is as follows:

```xml
<xs:complexType name="AssethealthType">
  <xs:sequence>
    <xs:element name="AssetSpecifications" type="xs:string" />
    <xs:element name="AssetDegradationprofiles" type="xs:string" />
    <xs:element name="AssetOperationInformation" type="xs:string" />
    <xs:element name="RepairCost" type="xs:int" />
    <xs:element name="ReplacementCost" type="xs:int" />
  </xs:sequence>
</xs:complexType>
```

- **Decision Options List Type and its Ranking**: Comprising the information related to the options for the decision, i.e. replace, repair, refurbish, maintenance, combination of both variables like repair some part and replace remaining part, etc. After defining the decision options, next need to rank these options and this ranking of decision options will be done through the complex type Ranking Options list type. The code for these two complex types is as follows:

```xml
<xs:complexType name="DecisionOptionsListType">
  <xs:sequence>
    <xs:element name="DecisionOption" type="xs:string" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RankingOptionsType">
  <xs:sequence>
    <xs:element name="DecisionOption" type="xs:string" />
    <xs:element name="Rank_Given" type="xs:int" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```
Objective Attribute List Type: - This complex data types is to define various decision goals and attributes for the sub process AM decision objective identification process. These are the goals that need to be achieved by the decision. The XML Schema code for this complex types is as follows:

These are the major complex data type schema definitions derived for the AMSDM workflow model. These data types includes main process as well as sub processes. These complex data types can be used by any process in the AMSDM workflow model, and can be used any number of times with appropriate mapping. For these complex data types, the lower hierarchical order must be definitely the normal data types like string, int, double,
etc. These complex data types are of nested types i.e. in one complex data type there may be another complex data type, but the lower hierarchical order must be one, from the recognised data types.

**Global Variables**

In YAWL language Net variables will be acting as the global variables. The purpose of these variables is to store data that can be manipulated by any individual task in a net. The tasks within that net may need to read or update these net variables. For example the data type AMDecisionType offering a structure that comprises all decision-related information. A net variable AMdecision_List of the data type AMDecisionType is defined. The content of this variable will be continuously updated and data gradually added to it as decision passes through the different steps of the process. Furthermore, the net variables Decision_options, Check_DecObjectives, AM_Activity, Decision_Objectives, etc are defined. Among these variables, some of them are output only and some of them are local. Output only is for those variables that are assigned after the process started execution. Local variables can start any time, but need to have an initial value. Figure 5.10 shows the net variable definitions in YAWL.

![Net Decomposition](image_url)

**Figure 5.10: Net variables for the AMSDM Process**
5.1.2.2 Data Mapping

After successfully defining the complex data types and net variables, it is time to define the task variables for storing data that can be used or modified only within the context of individual execution instances of a certain task. In the case of a composite task, its task variables are conceptually the net variables of the corresponding subnet linked to that task. Like nets, tasks have decompositions where we can specify variables and a label to associate with the task or net. Data passing between tasks is achieved by passing values between a net and its tasks where XQueries may be used for transformation purposes. It is not possible to directly pass data between tasks; since each task variable is local to its task (i.e. not accessible by other tasks). Task variables have several uses. One use is to transferring information between workflow users and workflow engine. A second use is for passing data between web services and/or external code and/or applications that the running workflow engine invokes and the net the task resides in. Next to the task variables, task parameters should be discussed. There will be two task parameters, namely input parameters and output parameters. Both the parameters can be assigned to any tasks to allow the passing of state between nets and their tasks, and between tasks and workflow engine users and web services.

**Input parameters** use an XQuery to massage a net variable state (across possibly several net variables) into a value that can be passed to a single selected task variable.

**Output parameters** use an XQuery to massage a task variable state (across possibly several task variables) in to a value that can be passed to single selected net variable.

Based on the data that was collected for each task in ameliorated AMSDM process of Section 4.4, the data mapping is preceded in this AMSDM workflow model.

**Step 1: Identify an AM decision**

The task variables for the first task, ‘Identify an AM decision”, is AM_Decision, which is of output variable and AMDecisionType. During the execution of the task, the variable gets the values from the user and passes them to the net. The value of the AM_Decision task variable is passed to net variable through the XQuery {/AM_Decision_List/AM_Decision/*}. Figure 5.11 shows the update parameters window for the task “identify an AM decision”, displays the task variables and the XQuery for the output parameter.
Figure 5.11: Parameter definition for identify an AM decision

Step 2: Identify decision objectives and constraints

The task variable for the ‘check decision objectives’ is the check_objectives. It is an output only variable of Boolean type. For the ‘Decision objectives information’ task, task variable is Decisionobjectives. It is an input and output variable of ObjectiveAttrListType. The value of the Decision_objectives net variable is passed to it through the XQuery in Figure 5.13, and similarly from task variable is passed to net variable through XQuery in Figure 5.12. Task variable Decisionobjectives get the value of the Decision_objectives net variable and displays it to the user. Now again the task variable receives the new value from the user and passes it to the Decision_objectives net variable by overwriting the old value.
For the “Gather decision Objectives information” the task variable is *DecisionObjectives*. It is an input only variable of *ObjectiveAttrListType*. The value of the *Decision_Objectives* net variable is passed to it through the XQuery

\[\{/AM_Split_decision_making_process/Decision_Objectives/\} \]

For the “AM decision objective identification sub process’, the task variables are *Goals, Am_Attributes, Goal_Attributes*, and two *Boolean* variables. The *Goals* is an output variable of *objectiveListType*. *Am_Attributes* and *Goal_Attributes* are output variables of
ObjectiveAttrListType. These task variables will act as net variables when we go into the execution of this sub process. The value of the Decision_Objectives net variable will be updated with the value of Goal_Attributes task variable, and it is passed through XQuery and is shown in Figure 5.14. 

\{/AM_Decision_Objective_Identification_Process/Goal_Attributes/\}

Figure 5.14: parameter definition for AM Decision objective identification process

**Step 3: Identify asset conditions**

The task variable for the “Check asset condition” is the Boolean variable with the name Check_Asset_Condition. The flow detail for this Boolean variable in order to select the order of execution of tasks is done through XPath Expression, which are shown in Figures 5.15 and 5.16
The task variables for the “Gather asset condition and operation information” are the Asset_Conditions and Operation_Information. Asset_Conditions is an output only variable of AssetConditionsListType in which the user or Asset manager will gather the conditions of an asset. Operation_information is also an output only variable of operatingInformationType, which will gather regarding the operation information of the asset. The values of the net variables Asset_conditionsList and Operating_informationList are updated with the values of the corresponding task variables. The values are passed through the XQuery expression as shown in Figure 5.17.

Step 4: Define decision options
For the task “Define decision options” the task variables are Operating_Information, Asset_Condition, and Decision_Options_List. The first two variables are the input only
variables. The values are passed in to these variables from the net variables `Asset_conditionList` and `Operating_informationList` respectively and display the list to the user. `Decision_Options_List` is the output only variable of `DecisionOptionsListType`, in which the user will define the various options for the decision, and adds empty fields of the net variable `Decision_Options`. The passing of information is shown in Figure 5.18.

**Figure 5.18: Parameters definition for Define decision options**

**Step 5: Select a decision option**

The task variable for the “Are the decision options ranked” is the Boolean variable. For the task “Gather option ranking information” the task variable `Decisionoptions_rank` is defined. `Decisionoptions_rank` is an input and output variable of `RankingOptionsListType`. It takes as input the value of the net variable `Decision_options`. During execution, the task variable is populated with the data from the user and added to the end to the net variable `Rankinggivenlist`. As `Decisionoptions_rank` is an input variable, all the items contained in the list are displayed for the user at the execution of the task. This also implies that their values are editable.
The variable definitions and the XQuery for the output parameter are shown in Figure 5.19. The XQuery for the input parameter (adding the value for Decisionoptions_rank to the items of the Rankinggivenlist) is displayed in Figure 5.20.

**Figure 5.19: Parameter definitions for Gather option ranking information**

**Figure 5.20: XQuery for input parameter for the task**
The task variables for the best option and gather the decision parameters are \textit{Ranking\_Information}, \textit{Best\_Option}, and \textit{Dec\_Parameters}. \textit{Ranking\_information} is of input only variable of RankingOptionsListType. \textit{Best\_Option} and \textit{Dec\_Parameters} are the output only variables of int and DecisionParamListType respectively. The values are passed in to these variables through XQuery.

\textbf{Step 6: Optimise decision parameters}
Different variables under these tasks are of the Boolean variables, which decide the flow of the execution of tasks in which direction to proceed. The other task variables under these tasks are \textit{Relationshipinf} and \textit{OptimaldecParam}. These two variables are of output only variable and of string type. The values will be passed in to these variables through simple XQuery expressions and update the empty net variables with these values given by user, \text{"/Gather\_the\_relationship\_information/Relationshipinf/text()"}. \text{"/identify\_the\_Optimal\_decision\_param/Optimaldecparam/text()"}. As mentioned in chapter 4 and in Section 3.2.1.2, that the optimisation models are still in infancy stage, and needs to be developed. Also the data in these steps is completely curves, functions, analysis, etc. The YAWL data perspective cannot capture these curves or functions, only it can trigger the necessary tools or tasks and allocate them to different participants. Hence we will model this step in terms of Control flow and resource flow perspective, but not the terms of data flow perspective. Here in this step, the variables defined above will be used simply to carry the flow and also to capture the supported data of YAWL language. In addition to that “Relationship analysis sub process” is still in developing stage and need to be completed in future. Hence we will be not modelling this sub process.

\textbf{Step 7: Verify the decision}
The task variable for this task is the \textit{VerifyDecision} which is of output only variable and AMDDecisionType type. This variable collects the information from the user or from the “Simulation/sensitivity analysis sub process” and updates the net variable outcomes. The information from task variable to net variable \textit{Verify} is passed through the XQuery output parameter \text{"/Verify\_the\_decision/VerifyDecision/*"}. The Simulation/sensitivity analysis sub process is another sub process which needs to be developed in future (from Section 3.2.1.2).
Step 8: Output the decision

The variable for this task is the FinalDecision which is the input only variable of AMDecisionType. The decision which we gave initially in the starting task of the process will go through all these tasks and make the necessary modifications to the decision by various analyses and outputs the right decision on verifying it. The value to the task variable is passed from the net variable Verify, and is done through the XQuery input parameter. The variable definition for this task is shown in Figure 5.21.

![Update Parameters for Atomic Task "Output the decision"](image)

**Figure 5.21: Parameter definitions for Output the decision**

With all these steps, the data flow modelling for the AMSDM workflow model successfully completed. However the data flow modelling for the sub processes will be presented in the Appendix. This data modelling is an important step in modelling the process with the YAWL language. Without it, the development of AMSDM workflow model is not possible. Hence the data modelling to obtain the AMSDM workflow model is yet another important contribution for this research. Due to this data modelling, we captured the complete data of the AMSDM process.
5.1.3 AMSDM Process Resource Modelling

After successfully completing the control flow and data flow modelling, next step is to model the resources. Resources are needed to execute work items, i.e. invocations of tasks for specific cases. A resource is an entity that is capable of doing work and can be classified as either human or non-human. The various terms used under the resource modelling were already defined in Section 4.5. In addition to that, the possible various roles for the AMSDM process also defined. Based on these roles and discussions in Section 4.5, we model the YAWL modelling way of resources for the AMSDM workflow model. Due to this resource modelling, the coordination and collaboration of people is possible. In this research, the initial stage of resources were modelled which gives contribution towards the resource modelling. The resource modelling section is explained in two sub-sections. AMSDM resource model organization data is explained in Section 5.1.3.1 and resource allocation for tasks is explained Section 5.1.3.2.

5.1.3.1 Organisational Data

The YAWL resource model will start by starting the YAWL engine. When the YAWL engine started, the PostgreSQL and the Apache Tomcat services will be started. After starting these two services open the web browser and type the following address http://localhost:8080/resourceService/faces/participantData.jsp then a window will be opened asking for username and password. As we are using the YAWL for the first time, login into the YAWL with default username and password then a window will be opened as shown in Figure 5.22. From the OrgData tool, the various Role names will be created who will be taking part in the AMSDM Process model. At this step entering the various role names is mandatory, whereas the entering the Capabilities, Positions, and Orggroups are optional.

As mentioned in Section 4.5 that, not dealing with any particular organization or company and the AMSDM workflow model is the generic process model, hence not able to mention the capabilities, positions and orggroups sections. If, consider any specific organizations then the capabilities are like in which area the person is working, what are his strengths and weaknesses, power, whether to take decisions, etc? Positions are like the current position of the person in the organisation, power, responsibilities, etc. By taking all these factors into consideration, the work item is offered to the particular role. But as not dealing with any particular organization these capabilities and positions are not defined. Hence, several role names like Asset manager, Audit Commission, Board
Person, CEO, decision maker, System, user, etc are added. These role names are added with the description of the role to give idea regarding the work that need to be performed in this process model as shown in Figure 5.22. Here provided all this information to just give sufficient knowledge for the reader.

![Roles for the AMSDM process model](image)

**Figure 5.22: Roles for the AMSDM process model**

After successfully creating all possible roles, next step is to create users or participants for each role. In YAWL, a human resource is referred to as a participant. Each participant may perform one or more Roles, hold one or more positions and possess a number of capabilities. Participants are the actual resources who will be undertaking the execution of the tasks and choose the work item. Figure 5.23 shows the way to create participants and allocate to each role. From the Users tool, the various participants for each role will be created. As we didn’t define the positions and capabilities while defining the roles, hence while creating participants also not bother about these two terms. Privileges are like where the participant is giving a chance to select the work item, reorder the work item, etc. Figure 5.23 shows an administration form for maintaining organisation data in terms of individual users. The remaining tools like work queues, admin queues will be discussed in next section.
5.1.3.2 Task-Resource Allocation Scheme

After successfully completing the creation of Roles and Participants, next step is to manage the created resources and allocating to tasks. This should be done for each task individually by selecting the Manage Resourcing wizard. This wizard consists of five steps. These five steps are based on the three key interaction points for managing the resourcing of work items spawned from a task.

- **Offer**: The point at which it is decided that a number of participants could undertake the work item.
- **Allocation**: The point at which one of the participants offered the work item is nominated to do that work item.
- **Start**: The point at which the participant allocated a work item begins working on it.
At each of these interaction points, we may choose to have the system dynamic make a
decision on resourcing at each point, or alternatively, allow a user to manually make each
decision. Among these two choices, a process designer must specify that each of the
interaction points be either user or system-initiated. If an offer is user-initiated, it is
passed to an administrator so that it can be manually offered or allocated at a later time. If
an offer is system-initiated, the designer must also provide details of a distribution set of
resources to which the offer should be made. A distribution set may consist of the union
of zero or more individual participants, zero or more roles, and zero or more dynamic
variables (which at runtime will be supplied with details of participants and/or roles to
which the task should be offered). The resultant distribution set may be further filtered by
specifying that only those participants with certain capabilities, occupying certain
positions and/or being members of certain org groups, be included.

If an allocation is user-initiated, the task is placed on the offered queue of each of the
participants in the distribution set, from which one of the participants may manually
choose to allocate the task to him/herself, at which point the task is removed from the
offered queues of all offered participants. If the allocation is system-initiated, an
allocation strategy (e.g. random choice, round robin, shortest queue) is invoked that
selects a single participant from the distribution set, and the task is placed on that
participant’s allocated queue.

Finally, if a start is user-initiated, a participant must select the task from their allocated
queue to start execution of the task. If a start is system-initiated, the task is automatically
started and placed in the participant’s started queue for action.

However in the AMSDM workflow model all the three interaction points are user-
initiated. The reason for choosing the user is that, as we have not mentioned the
capabilities and positions while mentioning the roles. These capabilities and positions
will be used here as filters for the system to select the best participant for the work item.
For user-initiated decision there will be only two steps among the five steps of manage
resourcing wizard. In user-initiated decisions, we will be forwarded directly from step 1
to step 5 by avoiding the in-between three steps. The five steps of the manage resourcing
wizard are:
Step 1- Choose that all the three interaction points will be performed by user.

Step 2- Select the various roles and the corresponding participants to whom should be informed of the existence of the work item.

Step 3- Arrange the filters based on capabilities and organisation data, so that only one participant will be suitable for the work item.

Step 4- Specify the system behaviour

When allocating a work item like Random choice, shortest queue, etc. these 2, 3 and 4 steps will be avoided. Figure 5.24 shows the step 1 of manage resourcing wizard.

Figure 5.24: Manage Resourcing wizard for the step 1 of interaction point

After completing step 1, the next step will be directly forwarded to step 5 in which establish default user runtime privileges for the task. The privileges are all predefined and should select the list boxes as either yes or no. Figure 5.25 shows the user privileges.
After successfully selecting the runtime privileges, the Manage Resourcing wizard will be completed. As mentioned that the behaviour of interaction points will be selected by the user, so during the execution, the process designer needs to select the various participants based on the roles that are suitable for the execution of the task. A designer may also specify certain constraints to apply, for example that a certain task must not be performed by the same participant who completed an earlier task in a process (Separation of Duties), or that if a participant who is a member of the distribution set of a task is the same participant who completed a particular previous task in the process, then they must also be allocated the new task (Familiar Task). At runtime, a participant, having the required privileges (or authorisations), can further affect the allocation and execution of tasks. If a task is allocated to them, he/she may: deallocate it (pass the task to an administrator for manual reallocation); delegate it (to a member of their ‘team’ – those who occupy positions that ultimately report to the participant’s position); or skip the task (complete it immediately without first starting it). If the task has been started, a participant may reallocate it (to a member of their team), and in doing so may preserve the work done within the task thus far (stateful reallocation), or to reset the task data to its original values (stateless reallocation).
With all these necessary steps, the modelling of resources is successfully completed and also gained the access to coordinate between different people and various roles. Due to this coordination, the decisions can be made promptly without any delays. Hence the resource modelling is a new arena for modelling the processes and in particular to the AM processes is a novel approach. This results in yet another contribution towards the modelling of AM processes in this research.

The successful completion of these three modelling steps leads to the AMSDM workflow model. Now the next step is to execute this workflow model. The execution of this model will results in the output. This newly developed AMSDM workflow model contributes new knowledge to the broad discipline of asset management.

5.2 AMSDM Process Execution

As discussed in the Section 3.5, workflow specifications are designed using the YAWL designer and deployed into the YAWL engine which, after performing all necessary verifications and task registrations, stores these specifications in YAWL repository, which manages a collection of “runable” workflow specifications. The name of the specification is “AM split process”. Once successfully deployed, AMSDM workflow specification can be instantiated through the YAWL engine, leading to workflow instances or cases. The engine handles the execution of these cases, i.e. based on the state of a case and its specification, the engine determines which events it should offer to the environment. The environment of a YAWL system is composed of so-called YAWL services. Inspired by the “web services” paradigm, end-users, applications, and organizations are all abstracted as services in YAWL. The YAWL worklist handler corresponds to the classical worklist handler (also named “inbox”) present in most workflow management systems. It is the component used to assign work to users of the system. Through the worklist handler users are offered and allocated work items, and can start and signal their completion. The user interface in the runtime environment is presented by means of Web pages and forms within a browser. More details regarding the YAWL system, YAWL services and YAWL architecture can be found in Chapter 3.
The AMSDM workflow model specification is saved in the format of .yawl file. This .yawl file is to be deployed and can be instantiated in to the YAWL engine leading to launch a case. In addition, administrators can upload/unload a process specification as well as launch and cancel process instances using a case management administration form shown in Figure 5.26. If any sort of errors existed in the specification, then the .yawl file will not be uploaded. After successfully launching the .yawl file the execution of the AMSDM model will be started. We can cancel the launched case at any time in middle of execution using the “Cancel Case” button. This will stop the execution of specification.

Figure 5.26: Admin form for loading and launching the Case

The user can launch the number of cases for the same specification. For example, in “AMsplitprocess” specification, launch three cases, one case is for the refurbishment of building, second case is for the renewal of road, and third case is for the maintenance of pump. These three cases are independent decisions not related to each other. In this manner if any industry wants to make a decision among their several assets with different
activities, then launching number of cases for the same specification is the best solution. Figure 5.26(a) shows the form for launching number of cases. In this figure, under the Running cases can see no of cases for the “AMsplitprocess” specification. The numbers on left side of case name are the case instances, which differentiates the same specification as the different cases. The number on right side is not important. The execution of all the three cases will start simultaneously. The user can execute all the three cases simultaneously or one after another.

![Image of YAWL workflow interface](image_url)

**Figure 5.26(a): Admin form for launching three cases**

Here, provided this special case to just give an idea that it is possible to launch three cases at a time. However, we will be launching only the single case and shows the execution of that single case.

Workflow specifications are managed by the YAWL repository, and workflow instances are managed by the YAWL engine. Clearly, there is also a need for administration tool that can be used to control workflow instances manually (e.g. deleting a workflow
instance or a workflow specification), manually allocate resources to tasks, and provide information about the state of running workflow instances and details or aggregated data about completed instances. This is the role of the administration tools integrated into the resource service. Figure 5.27 (a) shows the various administration tools which are as follows:

![Administration tools in YAWL](image)

**Figure 5.27 (a): Administration tools in YAWL**

*Org Data:* - This admin tool is used to create and maintain organisation data in terms of roles, capabilities, positions, and organisation groups.

*Users:* - The admin tool, Users is used to create and maintain organisation data in terms of individual users.

*Cases:* - Administrators can upload/unload a process specification as well as launch and cancel process instances using a case management administration form.

*Services:* - This tool is used to add various services provided by the YAWL service.

*Admin Queues:* - This tool is used to manage and monitor the tasks that need to be offer, allocate, or start for execution.

*Work Queues:* - The work queues are used for managing and monitoring the offered tasks and try to allocate or start for execution.

After successfully creating the organisational data for the AMSDM model (Section 5.1.3), the execution of the model will be started through launching the case of AMSDM workflow specification. Immediately after launching the case the model will be enabled through the admin queues starting the first task. As mentioned that the three interactions
points were user-initiated in Section 5.1.3, hence *process designer* need to offer at least one participant to execute the task. When the process designer wants to offer participants to the task, he can see the list of participants available for offering. Among those participants will select one participant and offer it to the task. Figure 5.27 shows the admin screen for managing and offering task to the participant and Figure 5.28 shows the list of participants that are available for offering the task.

Figure 5.27: Admin Screen for offering task to a Participants

Figure 5.28: List of Participants Shown

Based on the information provided in modelling the control flow, data flow, and resource flow (Section 5.1), the execution of the ASMDM workflow model will take place. In previous paragraphs, we tried to give some sufficient background knowledge on how the execution will start in YAWL workflow language. In addition to that, which tools of YAWL will take part while execution. Now we concentrate on execution of the AMSDM workflow model. From Section 5.1, the various steps for execution are as follows:

**Step 1: Identify an AM decision**

The starting task of the model “Identify an AM decision” will be enabled through the tool *Admin Queues*. At this point, the designer needs to offer to a participant from the list of participants. After successfully offered to a participant, move to the tool *Work Queue*. At
this point the offered participant needs to accept the offer. After accepting the offer the work item needs to allocate to the same or different participant and starts execution. The allocated participant can deallocate the work item. Once the participant started work item that means, the participant has taken the commitment to execute that task or work item. These are the steps to follow for all the work items.

Now the task will execute and provides the output with two string variables and one date variable named AssetName, ActivityonAsset, and Timeduration respectively. The participant needs to provide the required information for those task variables and update them to the net variable *Am decision*. Figure 5.30 shows the output screen after successfully executing the task “Identify an AM decision”. For example the participant provided the information like the Assetname as Pump and ActivityonAsset as Repair and the TimePeriod to complete the activity. We will take this information as an example and precede the execution. For every work item three states will be existed, Cancel, Save, and Complete. Cancel is to cancel the work item execution and go back again to Work Queues tool for starting the execution of work item. Save, is to save the entered data and then go back to Work Queues tool. Complete, is for proceeding to next work item.

**Figure 5.29: Admin Screen for Work Queues while Executing**

---

162
Step 2: Identify decision objectives and constraints

From Figure 5.8 the next task that needs to execute is the “Check decision objectives” which is a decision making task with the Boolean variable. The output of this task is that the participant needs to select the check box (Figure 5.31), if the decision objectives are already defined. Otherwise need not select the check box, so that, the flow will proceed to sub process “AM decision objective identification process” composite task. The execution of the sub process will be explained in the Appendix section, so at present, consider as that the decision objectives are already defined. Next work item is “Gather decision objectives”. In the data flow modelling (Section 5.1.2) part of this work item, already defined some objectives as a process designer. Now the participant needs to provide the various attributes for each objective. YAWL supports automatic form generation for each work item. Hence, no of attributes for each objective can be created.
Figure 5.31: Check box for decision making task

Figure 5.32: Output Screen after executing the task “Gather decision objectives information”
Step 3: Identify asset conditions

After successfully executing the “gather decision objectives information task”, the next work item that needs to execute is the decision making task “check asset condition”. This task is also similar to the previous decision making task “check decision objectives”. The output is that the participant must select the check box to decide the direction of the flow of execution. The next work item in the list of execution is the “Gather asset condition and operation information”. The outputs of this work item are the various asset conditions and operating information. From the Sections 4.4 and 5.1.2, defined some conditions of the asset and those conditions will be shown as the output of the work item. YAWL supports the nesting of variables and also the selection of one variable among the various variables. These functions are shown in the execution output of this work item “Gather asset condition and operation information. Fig. 5.33 shows the output screen for this work item.

Figure 5.33: Output screen for the task “Gather asset conditions”
**Step 4: Define decision options**

The next work item that needs to execute is the “Define decision options”. Based on the asset conditions and the decision objectives, these decision options will be defined. Hence, these asset conditions and decision objectives are shown in the output while defining these decision options. For example, the options are 1) Replace the pump in a month 2) Repair and maintain pump for 2 years 3) Replace some parts and maintain remaining parts of pump for 5 years. Figure 5.34 shows the output.

![Output screen for the task “Define decision options”](image)

**Figure 5.34: Output screen for the task “Define decision options”**

**Step 5: Select a decision option**

The execution of this work item will results to rank the each decision option which were defined in the previous work item. The output is to give the rank for each option. This will be displayed on the screen as the decision option followed by its given rank. Suppose if the rank of the decision option1 is 3, the decision option2 is 2, and the decision option
3 is 1, then here the best option is the decision option3 i.e. replace some parts of pump and maintain remain parts for five years. This decision option will get analysed in next two steps to decide whether it is the best option or not.

The next two steps 6 and 7 are completely the analysis steps of the best decision, where the YAWL model can only trigger these analysis processes or tasks and allocate them to different participants but cannot generate the data. From Sections 4.4 and 5.1.2, it is clearly that generating output for these work items where the data type is of curves, functions, graphs, etc will not possible. It can only carry the generated data in between the respective tools, processes, or tasks. In addition to that these sub processes still need to be developed. Hence to show the output for these tasks is not possible at present. In future, once these sub processes are developed then the data generated in these sub processes and the corresponding tasks can be shown. So, we proceed directly to the output of the last step i.e. “Output the decision”.

**Step 8: Output the decision**

The output after the execution of this work item will be the AM decision. The AM decision which gave initially in the starting of the process will be processed through all these steps and gain the necessary changes where ever required and produces the final output. The output of this work item is the best decision option and it is retrieved from step 5. The best decision option in step 5 is to replace some parts of pump and maintain remain parts for 5 years. This decision will be displayed as the output of the decision.

With all these steps, the execution of the AMSDM workflow model successfully completed. The execution will reveal the outcome of the model. The execution results in coordination of different people during runtime. Due to which it is possible to dynamically allocate the person to the task. The main contribution of this research in developing the AMSDM workflow model is that, it is a completely new concept applied to the broad discipline of asset management.
5.3 Summary

In this chapter, developed the workflow model for the AMSDM process, which was the main contribution in this research. In this work, modelled with all the three perspectives of YAWL language 1) Control-flow, which describes the manner in which the thread of execution flows through the process model 2) data flow, which aims to capture the characteristics of the data that occur repeatedly in the process and 3) Resource flow, which aims to capture the various ways resources are represented and utilised.

In the control flow definition, modelled the ameliorated AMSDM process in a step-wise procedure and explained the execution flow of the model. In the data flow modelling, captured and modelled the various data characteristics like data structure, data types, data mapping, etc of the model. In the resource flow, identified various roles and these roles are captured in the model through various ways by organizing resources data. By modelling these three perspectives, successfully completed the modelling part of the AMSDM process. Next step was to execute the model and renders the output of the model.
CHAPTER 6
CONCLUSIONS AND SCOPE OF FUTURE WORK

6.1 Introduction
An increasing amount of focus has rested on studies revolving around Asset Management (AM) particulate reinforced AM and its processes. This area has the potential for industrially viable research work. Similarly, the other area Workflow techniques also hold the promise to apply for new processes in order to optimise processes and gain benefits to the industry and academia, both. Both disciplines are gaining significant popularity in many industries thus fuelling the need for research in exploring the possible benefits of their cross-disciplinary applications. This research is thus inspired to investigate these two domains to exploit the application of workflow to modelling and execution of AM processes. Specifically, it will investigate appropriate methodologies in applying workflow techniques to AM frameworks.

Chapter 2, summarized the available literature in both the disciplines of AM and Workflow techniques. Based on this literature review identified several gaps and correspondingly generated opportunities to carry out research in both the domains. Grounded on these identified gaps and opportunities, the problem for the present investigation has been formulated.

Chapter 3, discussed the methodology and techniques expended in this research. This chapter discussed the methodology of AM decision making process and in particular about the AMSDM process. The AMSDM process had been explicated through several steps. The next challenge in this chain of inter-related events was to assess and identify the characteristics of the AMSDM process. In order to obtain a workflow model for AM processes commonalities and differences between different AM processes need to be identified. This was the fundamental step in developing a conscientious workflow model for AM processes. The next step was to review a number of contemporary workflow techniques and choose a suitable technique for application to AM decision making processes. Exploring the benefits of YAWL language, YAWL has been chosen as the suitable workflow technique for modelling the AMSDM process. The main benefit of applying WFM technique to the AM process is that to automate the AM process and also
to simply support the coordination and collaboration of people that are involved in carrying out the process. In addition to that, applying workflow models to AM processes can adapt and enable both ad-hoc and evolutionary changes.

Chapter 4 developed the ameliorated AMSDM process definitions to fill out the gaps that had existed in the original AMSDM process flowchart. Emphasise the significance/benefit of this ameliorated process definition: in a narrow sense, it maps from the original conceptual process definition of AMSDM to a detailed comprehensive process definition that is ready for implementation in a workflow/BPM system. In a broad sense, it bridges between the design of a process definition by a domain expert and the deployment of the process using workflow technology.

Chapter 5 developed the workflow model for the ameliorated AMSDM process as well as execute the AMSDM workflow model. The developed AMSDM workflow model is then deployed (semi-)automatically is a state-of-the-art WFMS demonstrating the benefits of applying the workflow technology to the domain of AM. The AMSDM workflow model was modelled with all the various perspectives of YAWL language in order to gain benefits in terms of expressiveness, rigorousness, adaptability, useability, extensibility, etc.

The entire work spectrum has been presented in two broad categories viz. ameliorating the AMSDM process definitions and the modelling and execution of AMSDM process with YAWL language. Both these categories are further subdivided into three divisions:- the Control-flow definition of the AMSDM process, the Data-flow definition of the AMSDM process, and the Resource-flow definition of the AMSDM process. In addition, identifying the characteristics of the AMSDM process and associating the AMSDM process and the YAWL language are presented.

6.2 Summary of Findings

1. Out of the various AM processes, AM decision making processes were taken as first step for modelling. It provides the benefits of processing and yet retains the simplicity and ease in making decisions.
2. The work identified various characteristics for the AMSDM process, which gave clarity in modelling the process. In addition to that, it gave some lucidity in understanding the decision making processes. This fills the objective of identifying the characteristics for the AM processes.

3. Analysis has been done in selecting the WFM technique for modelling the AMSDM process. In this analysis, YAWL language was selected as suitable one for modelling the AMSDM process due to its various application principles like expressiveness, rigorousness, extensible nature, usability, flexibility, etc, which were not seen in any other WFM technique. This fills the objective of reviewing and identifying the suitable WFM technique for AM processes.

4. This research developed the framework for associating the AMSDM process and the YAWL language. This framework provides the connection between two different fields. It also contributes the step-wise procedure in dealing with two different broad disciplines.

5. This research identified several gaps in the AMSDM process. These gaps lead to re-addressing the issues in the AMSDM process in terms of its activities and the direction of flow. The work enabled the process to become more comprehensible in terms of addressing various issues related to AMSDM process.

6. This research developed the ameliorated AMSDM process definition, which specifies the information needed for modelling the AMSDM process. Emphasise the significance/benefit of this ameliorated process definition: in a narrow sense, it maps from the original conceptual process definition of AMSDM to a detailed comprehensive process definition that is ready for implementation in a workflow/BPM system. This satisfies the objective of developing an intermediate process between the AMSDM process and the WFM technique.

7. The major set of data was collected and analysed for the AMSDM process. This data is based on the various decisions and conditions that were involved in the AM processes. It will be imported in to the AMSDM workflow model while modelling the process. This accomplishes the objective of analysing the data of AMSDM process that is required for modelling.
8. The research identified various roles for the AMSDM process by considering and analysing the various organisation data of general decision making processes. The participants of these roles are the actual persons who will be executing the AMSDM workflow model. In some cases the work may be executed by machines, departments, etc. These roles will involve the coordination and collaboration between different people in making decisions.

9. This Research developed the AMSDM workflow model using the YAWL language based on the data that was collected and the roles that were identified. This model will make more prosperous and limpidity in making decisions compared with the normal AMSDM process. In addition to that, it will support coordination between various resources or people.

10. This research contributes to the new arena of modelling and execution of the AM processes, which achieves the objective of modelling the AM process.

6.3 Scope for Future Work

The coverage of this thesis is process and technique specific. The development of AMSDM workflow model using YAWL language has been attempted for investigation. However the thesis delivers a sequential procedure form process development, analysis, data analysis, and process model. As this research contributes to two different disciplines, thus the scope of future work is bidirectional.

The future work might be 1) to develop the sub processes for AMSDM process, which leads to further improvements in the process, and 2) to apply the attempted workflow model to other related processes and further verified.

1. The present work involved development of AMSDM workflow model, which was not integrated with the database. It is possible to create the integration so that the decisions can be made directly by the system without the human involvement. This means that the decision making task will be automated.
2. The developed AMSDM workflow model was a generic model and did not relate to any specific organization or industry. It is ideal to have a further study involving specific organization.

3. In the AMSDM process there were a couple of undeveloped sub processes named “Relationship analysis process” and “Simulation/Sensitivity analysis process”. There is a need to conduct further research in developing these sub processes.

4. In the developed AMSDM workflow model, the resource modelling is an initial step and need to be carried out further by depicting all possible roles, capabilities, and positions. These three variables are related to a specific organization. Hence there is a need to improve the resource modelling perspective in the AMSDM workflow model.

5. In the AMSDM process, there is a need to conduct further research in making the process more limpid and user friendly. The areas that need to be covered in this process are 1) Decision rule matrix, which summarises and accumulates AM decision makers experience and knowledge, and 2) The Analytical decision models which need to be developed for modules AHP, Multistage comprehensive Fuzzy evaluation, and for Cost-benefit analysis.

6. For the AMSDM workflow model, there is a need to create a “User form” in YAWL language which will make the model more perceivable and promiscuous to use for the end user.
REFERENCES


90. www.yawl-system.com, YAWL Website.


APPENDIX A
YAWL Model of the AM Decision Objective Identification Sub Process

The flow will proceed through this sub process when there is a need to identify the decision objectives. As in the Chapter 5, here also we explain the control-flow, data-flow, and execution of the process. However the resource-flow is similar to the way as explained in Section 5.1.3.

Control-flow definition

Based on the modifications that were made in Section 4.3, and Figure 4.3, the control flow definition for the AM decision objective identification process has been defined.

1. Define the goals and attributes for an AM activity

The flow will execute this task when there is a need to define the various goals for the AM activity of main process. The data required by this task are the various goals. The next task is to “Define the attributes for each goal”. The data required by this task are the attributes for the goals defined in the previous task. Fig A1 shows the control flow for the first two tasks.

Figure A1: Control flow for the task “Define the attributes for each goal”

2. Check whether business objectives available or not

This task is a decision making task which leads to another sub process named “AM objectives identification process” if the condition returns false. This decision making task is to check whether any business objectives are available or not. Next to this task the control flow will execute the “Audit the AM activity goals”. This task is to inspect the
defined goals and its attributes whether they are suitable to the decision objectives or not. Figure A2 shows the control flow for the tasks up to Audit the AM activity goals.

Figure A2: Control flow for the “Audit the AM activity goals”

For the remaining tasks, the control flow is similar to the previous tasks with the XOR splits and joins tasks. However the data flow and the variables used for these remaining tasks are different. Figure A3 shows the complete control flow of the AM decision objective identification process.

Figure A3: Control flow for the “AM decision objective identification process”
Similar to the “AM decision objective identification process” its sub process “AM objective identification process” control flow is also same with slight modifications. This sub process is to define the business objectives and its attributes, as well as AM objectives and its attributes, which is similar to define the goals and its attributes. Figure A4 acquires the complete control flow of the AM objective identification process.

![Figure A4: Control flow for the “AM objective identification process”](image)

**Data flow modelling**

In data flow modelling, first want to define the complex data types. For these two processes the complex data type is as follows:

**Objective Attribute List Type**: - This complex data types is to define various decision goals and attributes for the sub process AM decision objective identification process. These defined goals are the goals that need to be achieved by the decision. The XML Schema code for this complex type is as follows:

```xml
<xs:complexType name="ObjectiveAttrListType">
  <xs:sequence>
    <xs:element name="Objective_Attr" type="ObjectiveAttrType" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```
This complex data type can be used for both the processes in defining the goals and its attributes for AM decision objective identification process, and in defining the business objectives and its attributes, AM objectives and its attributes for AM objective identification process.

The next step is to define the net variables, which are already defined in chapter 5. The task variables and the data mapping are the ones which need to be discussed. Based on the data that was collected for each task in the ameliorated AMSDM process of section 4.4, the data mapping is preceded for these processes.

**Define the goals for the AM activity**

The task variable for this task is the *Goal*, which is of output only variable and ObjectiveListType. During the execution of the task, the variable gets the values from the user and passes them to the net. The value of the *Goal* task variable is passed to net variable *Goals* through the XQuery `/Define_the_goals_for_the_AM_Activity/Goal/*`. For the next task “Define and quantify indicators for each goal” task variable is *Attributes* which is of input and output variable and ObjectiveAttrListType. The value of the *Goal_Attributes* net variable is passed to it through the XQuery as shown in Fig. A5, and similarly from task variable *Attributes* is passed to net variable *AM_Attributes* through XQuery in Figure A6.
After successfully defining the goals and its attributes, the next step is to check the business objectives which lead to the “AM objective identification process”. We discuss this sub process in the later stage, so now proceed with the next task which is to “Audit the AM activity goals”. The task variables for this task are AM_Attributes, Goal_Attributes, and Audited_goalsandAttributes in which the first two variables are input only variables and the last one is output only variable of ObjectiveAttrListType.
The passing of data from net variables to task variables and from task variable to net variable is given by XQuery expressions which are shown in Figure A7.

![Figure A7: Data passing for the task “Audit the AM activity goals”](image)

Similarly, the data passing for the remaining tasks are also same with differentiating the name of task variables. Apart from the name of the task variables the data passing is similar. Hence we will be moving to the sub process “AM objective identification process”.

**AM objective identification process**

For this composite task the net variables are the task variables of its main process. The task variables are the one which need to be defined and they are *AM_Attributes* which is of output only variable of ObjectiveAttrListType, *Business_Attributes* of BusinessAttrListType, *Business_Objective* of BusinessListType, and *AM_Objectives* of ObjectiveListType. The net variables and tasks variables and the mapping between the two processes together are shown in Figure A8.
Figure A8: Variables and XQuery expression for “AM objective identification process”

There is not much difference between these two processes in mapping the data, except the difference of the variable names. Both the task variables and net variables will be changed, but the actual mapping and XQuery expressions are similar to its main process. Hence we are not discussing the data mapping of the AM objective identification process.

Execution of the AM decision objective identification process and its sub process

As we already explained how the execution of the process will starts and creating of roles and participants, and allocating tasks to the participants, now we directly show the execution of the AM decision objective identification process.

Define the goals for the AM activity

The output of this task is that the user needs to define the various goals for the AM activity. As we mentioned earlier that, we are not dealing with any particular organization, hence the output shown is the general output. For e.g. the various goals of the AM activity are increase reliability, reduce cost, maximize profit, etc. Fig A9 shows the output of the task.
Define the attributes for each goal

The goals defined in the previous task will be input for this task and will be displayed initially at the output screen. Now the user needs to define the attributes for these goals which were displayed on the output screen. For the goal increase reliability, the attributes will be maintenance, predictability, etc. For the goal reduce cost, the attribute will be dollars, percentage, etc. Figure A10 shows the output for this task.
To these defined goals and attributes, the modifications will be made in the following tasks where ever necessary. Finally, by satisfying all the conditions in the flow of execution, the finalised goals of the AM activity will be decided.

These finalized goals will be imparted in the AMSDM workflow model through proper mapping of both main process and sub process. The data generated by the output of the sub process is transferred to the main process through an input variable `DecisionObjectives`. 

Figure A10: output for the task “Define the attributes for each goal”
The flow will proceed through this sub process, when there is a need to define the decision options as well as to rank these decision options. As explained in Appendix A, we will explain in this appendix the control flow and data flow of the option selection sub process. Based on the information provided in Chapter 4, we modelled this option selection sub process. As we mentioned in the data analysis Section 4.4.2, the option selection sub process is still in developing state. Also the workflow technique will be applicable only up to certain level of tasks because of lacking in coordination of roles between tasks. Based up on this analysis the control flow and data flow of the process will be shown.

The complete control flow definition of the option selection sub process is as shown in Figure B1. The Option selection sub process is not modelled after the task “Choose an analytical model for decision support”. This is due to the lack of coordination of roles between various tasks.

![Control flow for the “Option selection sub process”](image-url)
**Data flow Modelling**

The complex data types that will be used in this sub process are

**Asset Health Type**: - This complex type Comprises of information related to the asset health, i.e. asset specifications, asset degradation profiles, asset operation information, repair cost, replacement cost, etc. The code of XML Schema for this complex type is as follows:

```xml
<xs:complexType name="AssethealthType">
  <xs:sequence>
    <xs:element name="AssetSpecifications" type="xs:string" />
    <xs:element name="AssetDegradationprofiles" type="xs:string" />
    <xs:element name="AssetOperationInformation" type="xs:string" />
    <xs:element name="RepairCost" type="xs:int" />
    <xs:element name="ReplacementCost" type="xs:int" />
  </xs:sequence>
</xs:complexType>
```

**Decision Options List Type and its Ranking**: - Comprising the information related to the options for the decision, i.e. replace, repair, refurbish, maintenance, combination of both variables like repair some part and replace remaining part, etc. After defining the decision options, next need to rank these options and this ranking of decision options will be done through the complex type Ranking Options list type. The code for these two complex types is as follows:

```xml
<xs:complexType name="DecisionOptionsListType">
  <xs:sequence>
    <xs:element name="DecisionOption" type="xs:string" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RankingOptionsType">
  <xs:sequence>
    <xs:element name="DecisionOption" type="xs:string" />
    <xs:element name="Rank_Given" type="xs:int" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```
The task variables are the one which need to be defined. The various task variables defined in this sub process are shown in Figure B2.

Figure B2: The task variables defined in the Option selection process

The data mapping for the initial tasks are similar to the previous processes, hence we are not discussing the data passing information for these tasks. But for the task “choose an analytical model for decision support” the data mapping will be slightly changed.

The task variables for this task are *Analytical_Model* and *GuidelinesFor_AnalyticalModel* which are of output only variables and of type AnalyticalModelType and string respectively. The Guidelines to select the analytical model is passed from task variable to net variable through XQuery expression */Choose_an_analytical_model_for-
decision_support/GuidelinesFor_AnalyticalModel/text ()]. Fig B3 shows the data passing information for the task.

![Figure B3: Data passing information for the task “Choose an analytical model”](image)

Through this the data flow modelling for the process is successfully completed. The outcome of this sub process “Ranked decision options” will be imparted in to the main process to precede the flow of execution further to make an AM decision.