THE UNIVERSITY OF NEWCASTLE UPON TYNE
SCHOOL OF COMPUTING SCIENCE

# EVALUATION AND DEVELOPMENT MODELS FOR BUSINESS PROCESSES

BY
Fahad Mohammed Al-Humaidan

Ph.D. Thesis

February 2006

NEWCASTLE UNIVERSITY LIBRARY
204 26710 4
Thesis L8115

# Abstract

Most organisations are working hard to improve their performance and to achieve competitive advantage over their rivals. They may accomplish these ambitions through carrying out their business processes more effectively. Hence it is important to consider such processes and look for ways in which they can be improved.

Any organisational business process encompasses several elements that interact and collaborate with each other to achieve the required objectives. These elements can be classified into hard aspects, which deal with tangible issues related to the software system or the technology in general, and soft aspects, which deal with issues related to the human part of the business process. If the business process needs to be analysed and redesigned to improve its performance, it is important to use a suitable approach or intervention that takes into account all of these elements.

This thesis proposes an approach to investigate organisational business processes by considering both soft and hard aspects. The approach, Soft Workflow Modelling (SWfM), is developed as a result of reviewing several workflow products and models using a developed workflow perspectives framework which involves several perspectives covering the soft and hard aspects of the workflow system. The SWfM approach models the organisational business process as a workflow system by handling the various perspectives of the workflow perspectives framework. This approach combines the Soft Systems Methodology (SSM) with the Unified Modelling Language (UML), as a standard modelling language of the object-oriented paradigm. The basic framework adopted is that of SSM with the inclusion of UML diagrams and techniques to deal with the aspects that SSM cannot handle. The approach also supports SSM by providing a developed tool to assist in constructing a conceptual model which is considered as the basis to model the workflow system. A case study is developed for illustrative purposes.

# Acknowledgements

First, I would like to express my grateful to Dr. Nick Rossiter, my supervisor, for his support, guidance and constant advice throughout this work.

Secondly, I would also like to thank Dr. Steve Riddle, my co-supervisor, who has given me valuable advice and help.

Thirdly, I would like to thank all members of staff in the School of Computing Science particularly Shirley Craig for her help with the references and my friends in the School.

Fourthly, I would like to thank my small family, my wife, my son and my daughters who support me and give affection which is the energy that helps me steadily work and improve my self to finish my research. In addition my special thanks are extended to my family in Saudi Arabia especially my mother and my friends.

Finally, I would like to dedicate this thesis to my parents.

Table of Content

*Abstract* .......... *ii*
*Acknowledgements* .......... *iii*
*Table of Content* .......... *iv*
*List of Figures* .......... *vii*
*List of Tables* .......... *viii*
***Chapter 1: Introduction*** .......... ***1***
*1.1 Organisational Business Process* .......... *1*
*1.2 Workflow Technology* .......... *4*
*1.3 The Problem Definition* .......... *7*
*1.4 The Research Question* .......... *11*
*1.5 Thesis Outline* .......... *12*
*1.6 Terminology* .......... *13*
***Chapter 2: Business Process and Workflow Systems*** .......... ***16***
*2.1 Business Process* .......... *16*
*2.1.1 Business Process History* .......... *16*
*2.1.2 Business Process Framework* .......... *16*
*2.1.3 Business Process Life Cycle* .......... *19*
*2.1.4 Business Process Methodology* .......... *20*
*2.1.5 Business Process Modelling Techniques* .......... *20*
*2.2 Workflow Management Systems* .......... *23*
*2.2.1 Introduction* .......... *23*
*2.2.2 Workflow Management Systems Types* .......... *25*
*2.2.3 The Need for Standardisation* .......... *26*
*2.2.4 The Workflow Management System Architecture* .......... *28*
*2.2.5 WfMC Workflow Meta-model* .......... *29*
*2.2.6 The Workflow Reference Model Perspectives* .......... *30*
*2.2.7 Workflow models* .......... *35*
*2.2.8 Workflow modelling techniques* .......... *38*
*2.3 Chapter Summary* .......... *40*
***Chapter 3: Comparison of Workflow Models*** .......... ***41***
*3.1 Workflow Perspectives Framework* .......... *41*
*3.2 Workflow Models* .......... *43*
*3.2.1 Action Workflow* .......... *44*
*3.2.2 COSA (Computer-supported Case-handling)* .......... *46*
*3.2.3 FlowMark (IBM)* .......... *49*
*3.2.4 InConcert* .......... *51*
*3.2.5 Staffware* .......... *53*
*3.2.6 SAP Business Workflow* .......... *55*
*3.2.7 WorkParty* .......... *57*
*3.2.8 Domino* .......... *59*
*3.2.9 OfficeTalk* .......... *61*
*3.2.10 WIDE (Workflow on Intelligent and Distributed database Environment)* .......... *62*
*3.3 Evaluation of Workflow Models* .......... *65*
*3.4 The SSM and the workflow models* .......... *83*
*3.5 Chapter Summary* .......... *85*
***Chapter 4: The Soft Workflow Modelling (SWfM) Approach*** .......... ***86***
*4.1 The SWfM Framework* .......... *86*

4.1.1 Soft Systems Methodology (SSM)........87
4.1.2 The Unified Modelling Language (UML)........93
4.2 The Application of SSM to Modelling a Workflow System........96
4.2.1 The Implementation of SSM Techniques in a Workflow System........96
4.2.2 SSM Techniques and Workflow System Perspectives........98
4.3 The SWfM Approach........99
4.3.1 The Organisational Business Process Investigation........100
4.3.2 The Workflow System Modelling........103
4.4 Chapter Summary........117
**Chapter 5: The Implementation of the SWfM Approach in a Case Study........119**
5.1 The Case Study........119
5.2 The Implementation of the SWfM Approach........120
5.2.1 The Organisational Business Process Investigation........120
5.2.2 The Workflow System Modelling........126
5.3 The Effectiveness of the SWfM Approach........144
5.4 Chapter Summary........145
**Chapter 6: A Comparison of the SWfM Approach with Other Workflow Models 146**
6.1 Comparison Models........146
6.2 COSA........146
6.3 WIDE........148
6.4 The SWfM Approach........149
6.5 Comparison........151
6.5.1 The SWfM approach........151
6.5.2 The three Models compared........154
6.5.2.1 The conceptual level........154
6.5.2.2 The implementation level........154
6.6 Chapter Summary........155
**Chapter 7: Tool Support for the SWfM Approach........157**
7.1 The History of CASE Tools........157
7.2 The Classification of CASE Tools........158
7.2.1 Simple Tools........158
7.2.2 Workbenches........159
7.2.3 Integrated CASE........159
7.2.4 Open Environments........159
7.3 Workflow Definition Tools........159
7.4 Rationale for a Drawing Tool in the SWfM Approach........160
7.5 The SSM Tool........160
7.5.1 Visio Software........160
7.5.2 Tool Features........160
7.5.3 SSM Drawing Tool........161
7.5.4 An Example of Using the Tool........163
7.6 CASE Tools Integration........165
7.6.1 Visio Integration........167
7.7 Tool Support for Developing a Workflow System........170
7.8 Maintenance of the SWfM Approach Models and Descriptions........172
7.9 Advantages and Disadvantages of CASE Tools........174
7.10 Chapter Summary........175
**Chapter 8: Discussion and Conclusions........176**
8.1 Overview........176

*8.2 Contribution of the Research ... 177*
*8.3 Review of the Proposed Approach ... 179*
*8.3.1 The Role of Soft Systems Methodology ... 179*
*8.3.2 The linking of SSM and UML ... 180*
*8.3.3 The Role of Unified Modelling Language (UML) ... 181*
*8.4 Related Work ... 182*
*8.4.1 Soft and hard approaches ... 182*
*8.4.2 Soft approaches ... 184*
*8.4.3 Hard approaches ... 185*
*8.4.4 The Impact of UML version 2 (UML 2) ... 189*
*8.5 The Limitations of the SWfM Approach ... 191*
*8.6 Future work ... 192*
*8.7 Conclusion ... 194*
*References ... 195*
***Appendix A: Case Study ... 207***
***Appendix B: Tool Support ... 245***

List of Figures

*Figure 2-1: The WfMC Reference Model - adapted from the WfMC workflow reference model* ........ 29
*Figure 4-1: The SWfM framework* ........ 86
*Figure 4-2: The conceptual model notations* ........ 102
*Figure 4-3: Transition of a CM activity to a UML use case* ........ 107
*Figure 4-4: The use case diagram notations* ........ 109
*Figure 4-5: Activity diagram notations* ........ 111
*Figure 4-6: The sequence diagram notations* ........ 112
*Figure 4-7: The class diagram notations* ........ 114
*Figure 4-8: The deployment diagram notations* ........ 116
*Figure 4-9: The SWfM Approach: Stages & Steps* ........ 117
*Figure 5-1: The course management rich picture* ........ 121
*Figure 5-2: Course management conceptual model* ........ 123
*Figure 5-3: The high-level conceptual model for course management* ........ 124
*Figure 5-4: The conceptual model activities grouped into subsystems* ........ 125
*Figure 5-5: The conceptual model activities with their goals* ........ 130
*Figure 5-6: Activities to use cases mapping* ........ 132
*Figure 5-7: The use cases of the course planning subsystem* ........ 133
*Figure 5-8: Activity diagram for the 'specify the need for a course' use case* ........ 136
*Figure 5-9: Sequence diagram for the 'specify the need for a course' use case* ........ 138
*Figure 5-10: Class diagram for the 'specify the need for a course' use case* ........ 139
*Figure 5-11: Class diagram for the organisational view of the 'specify the need for a course' use case* ........ 139
*Figure 5-12: All diagrams for the 'specify the need for a course' use case* ........ 141
*Figure 5-13: A class diagram for a course planning sub-system* ........ 142
*Figure 5-14: Package diagram for the course management system and its subsystems* ........ 143
*Figure 5-15: A class diagram for the organisational view of the 'course planning' subsystem* ........ 144
*Figure 7-1: The SSM conceptual model notation* ........ 162
*Figure 7-2: A Screen shot of the Registration Subsystem* ........ 164

List of Tables

*Table 3-1: Summary of workflow model evaluation* ........ *81*
*Table 3-2: Workflow models comparison* ........ *82*
*Table 3-3: Summary of the role of SSM with workflow models* ........ *84*
*Table 4-1: A subsystem description* ........ *104*
*Table 4-2: An Activity Description* ........ *104*
*Table 4-3: The CM activity and UML use case comparison* ........ *107*
*Table 5-1: The course planning subsystem* ........ *127*
*Table 5-2: The 'specify the need for a course' activity* ........ *127*
*Table 5-3: The techniques required to represent the workflow perspectives* ........ *129*
*Table 6-1: COSA Model perspectives* ........ *148*
*Table 6-2: WIDE Model perspectives* ........ *149*
*Table 6-3: The SWfM Approach perspectives* ........ *150*
*Table 6-4: The comparison of the workflow models* ........ *154*
*Table 7-1: The effects of technique change on the other techniques of the SWfM approach* ........ *174*

# Chapter 1: Introduction

Nowadays, most organisations are facing challenging competition from other organisations and are influenced by constant changes in the surrounding environment. These may take different forms such as an increasingly dynamic economy, altered customer requirements, global competition, new technologies and the emergence of the Internet and electronic commerce. Therefore, organisations are looking for some means to help them face these challenges and to maintain their positions and percentage shares in the market. They are also looking to achieve a competitive advantage. To accomplish these, it is important for the organisations to manage and analyse their business processes and their structures so as to identify their critical success factors and to improve such factors while eliminating their deficiencies.

This thesis investigates the possibility of developing an approach for investigating and modelling the organisational business process through considering its elements, *soft* and *hard*, towards implementing it using a workflow system.

## 1.1 Organisational Business Process

Each organisation has objectives and goals that it tries to achieve through the organisational business process, which includes several parts or elements that interact with each other to accomplish the required objectives. In order to discuss the organisational business process, some definitions need to be reviewed. First, the business process can be considered in general terms as 'a set of partially ordered steps intended to reach a goal' [1] or 'a group of related tasks that together create value for a customer' [2]. In more detail, Platt [3] defines the nature of the process as 'the transformation of something from one state to another state through partially coordinated agents, with the purpose of achieving certain goals that are derived from the responsibility of the process owner'. Further, Davenport [4] describes the structure of the business process as 'a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action'. Finally, the business process can be presented in detail as 'a set of partially ordered process steps, with sets of related artefacts, human and computerised resources, organisational structures and constraints, intended to produce and maintain the requested software deliverables' [5].

From the above definitions, the business process may be said to consist of the following elements: ordered activities, human and computerised resources, constraints and business rules, a set of related artefacts and organisational structure. These elements aim to achieve the organisation goals and objectives.

The business process can also be defined from different viewpoints, such as the functional view, which deals with process activities and information flows; the behavioural view, which specifies when and how the activities are performed; the organisational view, which considers where and who will perform the activities; and the informational view, which deals with the informational entities [6].

With the advent of the Internet, the growth of electronic commerce (e-commerce) and the rise of virtual organisations, the business process extends over the organisational boundary to interact with the business processes of partner organisations which participate in shared business processes [7]. E-commerce is considered to cover all electronically mediated transactions between an organisation and any third party it deals with [8]. The UK government defines e-commerce in general terms as 'the exchange of information across electronic networks, at any stage in the supply chain, whether within an organisation, between businesses, between businesses and consumers, or between the public and private sector, whether paid or unpaid' [8]. This definition covers all types of e-commerce: internal organisation business, business-to-business (B2B), business to consumer (B2C) and e-government.

The notion of electronic business (e-business) has emerged, involving all activities of e-commerce, which is considered as a subset of it. IBM defines e-business as 'the transformation of key business processes through the use of Internet technologies' [8]. E-business deals not just with transactions but also the key business processes of an organisation which are required to be restructured to adopt the advanced technology of the Internet [8].

The relationships between partner organisations are created according to certain principles; for example:

- Goal orientation: Partner organisations agree on the expected results of the interaction.
- Privacy: Each partner organisation has its own private internal process that the other organisations cannot access.

- Flexibility: This is the ability for any partner organisation to change its private internal process without affecting the results agreed with the other organisations.
- Independence: Each partner organisation wants to stay independent of any internal changes of the others [9].

Furthermore, there is a need for a mechanism to allow partner organisations to reach an agreement on the business process specification and the data to be exchanged during process execution [10]. The agreement between partner organisations involves the goal of the interaction, the way in which information is exchanged between them and the circumstances in which both partner organisations interact to conduct business or perform a task [9]. Inter-enterprise process management involves collaboration among multiple distributed process managers [10].

To achieve the required objective of the organisation, the business process should be supported by the most suitable new technology available in the market. This allows the organisation to respond quickly to the changing environment of the business and accomplish a competitive advantage over its rivals. Before specifying the suitable information technology (IT) for supporting the business process, it is important to identify the organisational objectives and aims and to investigate, analyse and model the organisational business process.

Business process modelling is one of a number of methodologies, approaches, techniques and tools that support business process (re)design. It is crucial to consider the nature and features of business processes if modelling is to be effective [11]. The modelling approach should consider the organisation and its environment, including soft issues, which deal with people, and hard issues, which deal with technical elements. The purpose of business process modelling is to construct an abstraction that defines the constituent parts of the business process. This model is used to help analysts and users to learn about and to comprehend the coherence of the business process activities that are carried out to accomplish the business objectives. Further, the model assists in identifying the information technology required to support the business process [12]. By comparing the reference model of the software with the business process model, it is possible to configure the required software package by choosing those components and functions that are relevant to the enterprise [13].

Finally, the business process model is used to control and monitor the business process and to validate its performance [14].

## 1.2 Workflow Technology

Workflow technology is considered to be a part of IT. Workflow systems are considered the leading technology to support business processes [15]. Workflow is a notion that is used to express business process activities at a conceptual level required to comprehend, assess and restructure the business process [16]. The Workflow Management Coalition (WfMC) [17] defines workflow as 'the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participants to another for action, according to a set of procedural rules'. On the other hand, the workflow management system (WFMS) manages and controls the automation of the business process and its surrounding environment. WfMC [17] defines a workflow management system as

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

The workflow management system assists in reducing the required administrative efforts and overheads and improving the quality of service provided, by managing the activities of those processes that demand clerical responsibilities, in order to allow qualified people to focus on other activities which need intellectual effort [18].

The goal of workflow technology is to offer computer-based facilities for the execution and management of routine business processes in many domains which are predictable and repetitive in their nature [19] [20]. Workflow management systems include representations of working processes which are process models to manage the execution of their tasks, plan the required resources and control the flow of information to those performing the tasks [21] [15]. They concentrate on the automation of the routing and processing of information that is required to perform the various tasks of the business process and to make this information available to the management to assist in making decisions efficiently [22, 23]. The WFMS may include monitoring facilities that mange the whole process, trace the changes in its state and make sure that the business process achieves its objectives [22] [19].

Further, it supports the users by facilitating the coordination of the execution of their tasks so that they can concentrate on performing the work activities [21]; it may help to increase the job satisfaction of the employees by assigning tedious tasks to the workflow system [22]; it improves the quality of the process through the enforcement of a standard procedure to be followed to solve any problem in the process [19]; in all these ways, it tends to enhance the efficiency and effectiveness of the business process [22].

Organisational business processes can be executed manually or they can be depicted as workflows that can be managed and executed automatically by a workflow management system. Workflows define activities involving coordinated execution of their various tasks that are carried out by different entities. These tasks can be executed manually, by an automated system, or by a combination of both [24]. To specify a workflow, it is necessary to define the various aspects that are essential in designing it. These include the various activities or tasks that are constituents, the requirements and order of these activities, and the relationships between them. The execution of these tasks by different entities can be controlled through human coordination or by a software system, i.e. a WFMS [12].

When an organisation wants to extend and conduct its business through the Internet, the workflow management system can be used to manage the cross-organisational processes. It is important to deploy a workflow management system to manage the business processes that operate within, across or between partner organisations. A workflow system should integrate all business elements that traverse organisational boundaries and control the entire business operational flow [25]. Several mechanisms are used to manage business processes that cross organisational boundaries:

- Capacity sharing: one workflow manager manages the execution of tasks by external resources.
- Chained execution: the process is decomposed into consecutive phases and each organisation handles one phase.
- Subcontracting: a subprocess is performed by another organisation.
- Case transfer: each organisation applies the same workflow process and cases are transferred from one organisation to another.

- Loosely coupled: each organisation performs a specific part of the process [7].

The e-business application has several requirements: [26]

- The ability to produce new services to develop and deploy the business process;
- Making the business process flexible to adapt for changes in the business environment or IT;
- Process monitoring and capabilities tracking to manage the process;
- Proving the appropriate performance level as well as other issues such as scalability, reliability and high availability.

The workflow management system should support the integration between front and back-end systems and among the different back-end systems themselves [26]. This will produce several results, such as cost reduction, improving service quality and time-to-market and responding quickly to the changing business environment. Inter-organisational workflows support organisations to extend their business processes beyond the boundaries of their own organisations [26].

The development of a workflow system goes through four phases of the workflow application life cycle: analysis, development, execution and administration.

The analysis phase focuses on business process modelling and investigation of its characteristics. The purpose of modelling is to produce a specification of a process that can be used in the workflow specification. The business process model makes it easier to comprehend the structure of the process and its elements (activities and relationships between them; human and information system roles; skills).

The various parts of a process are specified explicitly in the development phase. The process activities and tasks, the relationships between them, and the required data are defined using workflow graphical tools. This phase also includes integrating the various applications that are used to perform or assist in carrying out the tasks through the deployment of available techniques such as Application Program Interfaces (APIs) and Remote Procedure Calls (RPCs).

In the execution phase, the workflow specification is converted into a representation that can be executed using the workflow management engine, which creates and manages an instance of a workflow and assigns tasks to a performer, who is a human or information system service, through the role concept.

The final phase is administration, in which the workflow performance is monitored by workflow administrators using monitoring tools. The output data from this phase can be used to identify any process bottlenecks and deadlocks, which are considered as feedback for both the analysis and development phases [12].

## 1.3 The Problem Definition

The business process is considered the backbone of an organisation, through which it can achieve its goals and objectives. It is crucial to manage the business process so as to utilise the organisation resources and capabilities efficiently [12]. In the case that an organisation interacts with other partner organisations, it is possible to extend the business process to handle this interaction and any related issues.

The management of business process is done through the use of IT to assist in supporting, monitoring and controlling the various aspects of the business process [27]. To do that it is important to investigate, model and analyse the business process and then redesign and implement it using suitable technology.

Workflow technology is used to support the business process by automating the coordination of the execution of the business process tasks, which is performed by enforcing procedural rules [24]. Before the WFMS can be used to manage the business process, it is essential to model the business process as a workflow process [24]. The WFMS uses a process model to plan and coordinate the execution of the various tasks [15]; it defines the business process as a workflow model and stores this definition in a workflow repository which can be used to create and execute workflow instances and manage the interactions between the tasks with the workflow instances during the execution or run-time [24]. Much attention has been concentrated on workflow implementation issues, while conceptual modelling issues have received little focus [28].

The Workflow Management Coalition (WfMC) has introduced a workflow reference model [29] that defines five types of interface which are used to improve interoperability in workflow technology. These interfaces are an interface 1 for the import and export of process definitions, an interface 2 for workflow client application interaction, an interface 3 for application invocation functions, an interface 4 for other workflow enactment services to support interoperability between workflow engines and an interface 5 for administration and monitoring functions.

Interface 1 of the model supports the process definition, which includes a meta-model for describing process definition, a Workflow Process Definition Language (WPDL) and Application Programming Interfaces (APIs). WfMC [30] defines the meta-model as consisting of the following aspects: workflow process definition, workflow participant specification, workflow application declaration and relevant workflow data. A workflow model involves a set of concepts that are used to define processes, their activities or tasks, the relationships between activities or tasks and the roles required to carry out the various activities and tasks [16].

Interface 4 handles the interoperability between the workflow engines by defining a standardised set of interfaces and data interchange formats between workflow system components [31]. This interface can be used to support the cross-organisational process.

In addition, the OMG Workflow Management Facility has produced the jointFlow [32] standard which transforms the WfMC standards into an object-oriented framework for CORBA infrastructure. Subsequently, two standards have been developed: the Simple Workflow Access Protocol (SWAP) to instantiate, control and monitor workflow processes, and wf-XML to define the XML data content required to communicate between workflow engines. The Business Process Management Initiative (BPMI) has also developed Inter-Enterprise Workflow Systems (IEWSs) that automate and manage cross-organisational business processes by integrating the various users, applications and systems [31]. Recent workflow projects have concentrated on integrating loosely coupled business processes, for example the eFlow [33] and WISE [34] projects.

Several approaches have been developed to deal with the cross-organisational business process as a workflow system. For example, Van der Aalst [7] has proposed an approach to model and analyse the loosely coupled workflow process using a Message Sequence Chart (MSC) to depict the interactions between the various business partners through the exchange of messages and a Workflow net (WF-net) which is a variation of the Petri nets technique to represent the internal behaviour of each business partner. Also, Punia and Saxena [35] discuss inter-organisational processes in the context of e-government, in which government services (processes) are performed by multiple government agencies. E-government means 'the use of Information and Communication technologies (ICTs) to promote more efficient and effective government services, allow greater public access to information and make government more accountable to citizens' [35]. They consider the inter-organisational structure involved in the process as a virtual enterprise (VE), which citing Tagg [36], they define as 'any alliance, temporary or permanent, between two or more legal entities, that exists for the purpose of furthering business or social objectives, without causing the participants to lose their autonomy'. The coordination in a virtual enterprises may follow any one of three models, namely the hierarchal (centralised, participative or decentralised), market or ad-hoc models [36]. They use the concept of public process to model inter-organisational processes. The partner organisations create new (public) processes to be integrated with their existing processes. These public processes are common to all the partners and interface with each organisation's internal private processes. The partner organisations interact with each other through the public processes only [35]. Moreover, Ludwig and Whittingham [9] introduce an approach, the Virtual Enterprise Co-ordinator (VEC), to manage cross-organisational business processes. This approach deals with managing the outsourcing of a subprocess to another partner organisation, which is called a 'parent process-subprocess' relationship and is based on a 'gateways' mechanism that is used to support communication between the partner organisations. These gateways, which are created at the organisational boundary of each partner organisation, are set up according to agreements that specify the common terminology, a shared view of the sub-process between the partner organisations on a case-by-case basis. Finally, Klingemann et al. [37] propose a service-oriented model of cross-organisational workflows for outsourcing activities or services. A service is 'an abstract specification of the amount of work that a resource promises to carry out with a specific quality of service' [37]. A service may span multiple activities. The interaction between organisations in this model is done through well-defined interfaces which specify the method of invoking an external service and the parameters that have to be passed.

Jablonski and Bussler [38] have introduced a reference model called the 'comprehensive workflow model'. This model includes eleven perspectives that cover most of the workflow model concepts: the functional, informational, behavioural, operational, organisational, security, causality, history, quality and autonomy perspectives.

Most of the existing workflow models can be classified and developed through the use of communication-based methodologies, activity-based methodologies or hybrid techniques combining the two types.

A communication-based methodology is based on the Conversation for Action Model of Winograd and Flores [39]. This methodology expresses an action in a workflow as communication between a customer and a performer, consisting of four phases: proposal, agreement, performance and satisfaction. This is called a workflow loop. A business process is represented by a network of connected loops in which one workflow loop invokes another to complete or make progress in the main loop [16].

An activity-based methodology represents the business process by modelling its activities or tasks and the relationships between them. Many commercial workflow management systems provide an activity-based workflow model [16]. There are also hybrid techniques which combine the techniques of communication-based and activity-based methodologies to depict the business process as a workflow model [40].

These methodologies, methods or approaches cover the perspectives that are related to the physical part of the business process in modelling for workflow systems, while neglecting or ignoring the soft part, related to the human aspects of the business process. However, several authors recommend the use of soft approaches to deal with the social perspective of the business process [11].

One of the well-known soft methodologies is Soft Systems Methodology (SSM) [41] [42], which is based on systems theory and aims to 'understand the fuzzy world of complex organisations' [43]. SSM focuses on issues that are neglected by traditional methodologies, such as considering the problem as a whole through its situation and surrounding environment. SSM recognises the importance of the human factor by including the people in the analysis and modelling of an organisation. Soft approaches assist in investigating problem situations by using different techniques to connect the situation to the organisational goals and policies. These techniques consider wider issues of the social sphere that may affect the problem situation, such as organisational structure, employee job satisfaction, usability and acceptance of the system [44].

## 1.4 The Research Question

Although soft approaches are useful and the produced models can be used to generate debate and learning about the business process [11], they cannot address all aspects of the business process.

As a result, the research described in this thesis has two important characteristics. First, it addresses the problem that most of the workflow system development methods or approaches ignore the soft issues related to the human aspects of the workflow system, including user involvement, identifying problems and objectives, various perceptions of system acceptance and use, and employee job satisfaction. Second, the research concentrates on modelling internal as opposed to cross-organisational business processes.

Therefore, the research question concerns *how to model the internal organisational business process by considering its soft and hard issues in order to develop a workflow system.*

This research proposes a new approach that utilises soft system approaches such as SSM to analyse and model the organisational business process and to handle its soft aspects, and technical or hard approaches such as Unified Modelling Language (UML) to deal with the hard aspects of the organisational business process, towards implementing it in a workflow environment.

Associated with the research question and the proposed approach, a workflow perspectives framework has been developed that involves perspectives representing both the soft and hard aspects of the organisational business process. The approach combines SSM with UML in such a way that the former is used as a framework guiding the approach to investigate, analyse and model the organisational business process in order to develop a workflow model that considers both soft and hard or technical issues of the organisational business process, while the latter deals with those aspects that SSM cannot handle. This approach is illustrated by implementing it in a course management case study. A CASE tool is also developed to facilitate the construction of a conceptual model of SSM that is considered the basis for developing a workflow model.

## 1.5 Thesis Outline

Chapter 2 discusses business process modelling, business process frameworks and workflow systems, which are considered as representations of the business process. This chapter introduces the business process framework, which represents the required features of the business process that any approach or technique should model. Also, it discusses some of the workflow models and presents a comprehensive workflow reference model, covering the various perspectives that a workflow model approach should consider: functional, behavioural, informational, operational, organisational, causal, etc.

Chapter 3 reviews some of the workflow models and compares them by using some of the perspectives of the workflow reference model of Chapter 2.

Chapter 4 introduces the approach taken in this thesis, which utilises SSM and UML to investigate and model the organisational business process model towards implementing it using a workflow system.

In Chapter 5, a case study dealing with course management in a university is used to illustrate and test the proposed approach by implementing it.

Chapter 6 compares the approach of the thesis, using conceptual and implementation levels, with the optimum or leading workflow model from Chapter 3 that covers most of the workflow model perspectives. The comparison is based on the workflow perspectives framework of Chapter 3.

A CASE tool is introduced in Chapter 7. It is used to develop the conceptual SSM model, which is the first part of the approach. It involves several features that assist the developer in constructing a conceptual model and validate its correctness and consistency.

The conclusion of this study is given in Chapter 8, where the research approach is reviewed and the achievements, contributions and problems are summarised. In addition, conclusions are drawn from the experience gained during the work and the significance of the findings is examined. Considerations for future work are also discussed.

## 1.6 Terminology

In the IT field there are some terms that can have several interpretations depending on the context; there is no general agreement on the definitions of these terms. In this section, some of them will be defined and reviewed in the context of this thesis. Some terms from workflow technology are also introduced and defined.

The first term is 'methodology', which is defined in the Cambridge Dictionary [45] as 'a system of ways of doing, teaching or studying something'. In the IT field a methodology is 'a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems' [46]. Avison and Fitzgerald [43] offer a general definition of methodology as

> 'a recommended means to achieve the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy that supports, justifies and makes coherent such a recommendation for a particular context. The recommended means usually includes the identification of phases, procedures, tasks, rules, techniques, guidelines, documentation and tools. They might also include recommendations concerning the management and organization of the approach and the identification and training of the participants'.

On the other hand, the method is considered as one scenario of the methodology that may contain instructions for adjusting the methodology to fit a specific situation. The Cambridge Dictionary defines 'method' as 'a particular way of doing something', while Flynn interprets it as 'an integrated set of procedures and techniques which, when applied in a certain sequence, result in the specification or generation of an information system' [44]. Another view of method takes into account its structure or composition, recognising that a method consists of phases which include several stages, each producing one or more phase deliverables [47]. Flynn [44] uses the terms 'methodology' and 'method' interchangeably.

Further, an approach is often considered to be the same thing as a method, but the term is not as tightly defined [44]. The Cambridge Dictionary [45] defines 'approach' as 'a way of considering something'.

A technique may be considered as part of a methodology or a method; it is 'a way of doing a particular activity in the information systems development process'

[43]. Flynn [44] defines a technique as 'a fairly precisely defined set of steps that address a particular (usually small) area of the life-cycle'.

Therefore, many practitioners in the IT field use the terms 'methodology', 'method' and 'approach' interchangeably to refer to part or all of the phases, stages or steps of the system development life cycle.

The intention is to conform to these terminologies throughout the thesis but on occasion particularly though in dealing with other people works it will be difficult to find the exact corresponding definition or meaning.

In addition, the workflow technology domain involves several concepts that need to be presented and described; they are as follows.

The WfMC [17] defines 'workflow' as 'the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules'. A workflow instance is 'a process combined with an organisation (actors or groups), and assigns the tasks in the process to actors in the organisation' [28] while a workflow model 'combines a process model (an abstraction of a business process) and an organisation model (an abstraction of an organisation)' [28].

A workflow management system is

> 'a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications' [17].

Process definition can be considered as

> 'the representation of a business process in a form which supports automated manipulation, such as modelling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc' [17].

This is similar to the process model, which is 'an abstraction of business processes [emphasising] the coordination of tasks by highlighting their interdependence' [28]. Finally, 'process instance' is defined as 'the representation of a single enactment of a process including its associated data. Each instance represents a separate thread of execution of the process which may be controlled independently and will have its own internal state and externally visible identity, which may be used as a handle.' [17].

# Chapter 2: Business Process and Workflow Systems

In this chapter, the concepts of business process and workflow technology are reviewed to set the scene for reviewing some workflow models and developing a new approach to modelling a business process as a workflow system.

## 2.1 Business Process

This section reviews the various aspects of the business process. It covers the business process history, its framework and life cycle and finally the different modelling methodologies and techniques of it.

### 2.1.1 Business Process History

The study of business processes goes back to 1911 when Fredrick Taylor conducted his systematic study of work procedures to improve efficiency and effectiveness [48]. Efforts to improve business processes have continued through business process redesign and design for manufacturing. In the early 1990s the term 'Business Process Reengineering' (BPR) was introduced by Davenport & Short [49] and Hammer [50]. However, following criticisms of BPR principles and the failure of several projects and organisations, interest has moved to 'Business Process Management' (BPM) which manages and controls the business process to improve business performance [51].

### 2.1.2 Business Process Framework

To study business processes, it is necessary to understand their nature by specifying their various elements and parts [11]. In this section a business process framework is developed that includes the required features and characteristics of any method or technique of studying the business process.

To develop this framework it is important first to review the definitions of 'business process' to determine its various elements. Hammer and Champy [50] describe a process as 'a set of partially ordered activities intended to reach a goal', while Hammer [2] defines the business process in general terms as 'a group of related tasks that together create value for a customer'. Jacobson et al. [52] view a business process as 'the set of internal activities performed to serve a customer'; for Hammer and Champy [50], it is 'a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes'. Davenport [4] uses the phrase 'a structured set of activities designed to produce a specified output for a particular customer or market'. These definitions focus on how work is done within an organisation, rather than its products.

Some definitions are more mechanistic; for example, Platt [3] defines the business process as 'the transformation of something from one state to another state through partially coordinated agents, with the purpose of achieving certain goals that are derived from the responsibility of the process owner'; and Davenport [4] refers to 'a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action'. Such definitions consider the business process as an input-transformation-output process and ignore the human side.

Medina-Mora et al. [39] classify the business processes into three categories: material processes, in which human activities are performed in the physical world; information processes, whose activities deal with the flow of information; and business processes, concerned with processing information. Alternatively, Ould [53] recognises three types of business process depending on value chain concepts: core processes that include the essential activities and deal with external customers, support processes which support the core process internally, and management processes which manage both the core and support processes. Ould [54] also specifies the key characteristics of the business process:

- It consists of purposeful activities;
- It is performed collaboratively;
- It is a cross-functional process;
- It starts with external agents or customers.

Loucopoulos [55] adds the following features of the business process:

- It has products and customers;
- It aims to achieve a defined goal that is considered as a value for the customers;
- It consists of activities that conjointly attain the process goal;

- It is a horizontal form that crosses functional/organisational boundaries to achieve the business goals.

On the other hand, Melao and Pidd [11] argue that any business process is multifaceted, such that each view of it is based on a set of assumptions about the nature of organisational life. They describe four alternative perspectives on business processes as deterministic mechanisms, complex dynamic systems, interacting feedback loops or social constructs. Each perspective focuses on some elements of a business process and ignores others; however; its nature emerges when these perspectives are considered together. Finally, Curtis et al. [56] consider several modelling approaches and propose a conceptual framework for modelling the software engineering process. This framework presents four views of the business process: functional (activities), behavioural (order of activities), organisational (organisation structure and actors) and informational (entities and their structure and relationships).

To conclude the discussion of definitions, Lindsay et al. [57] argue that most of the definitions and models deal with a mechanistic perspective. However, BPM adds the human aspect, so that a business process is defined in terms of individuals cooperating to achieve a goal [57]. Lindsay et al. [57] indicate that the methods focus on the internal structure of the process and organisation and there is a need for a holistic approach which current models do not consider.

For the purpose of this research, the definition of business process should include all the aspects that cover both 'soft' and 'hard' issues. Therefore, the following definition will be used throughout the research. The business process is '*a set of related activities that are carried out through agents such as humans, machines or software applications to achieve a goal in a defined environment*'.

The constituent elements of the business process are as follows:

- business process context
- business process activities
- business process hierarchy
- events
- process inputs and outputs
- resources (human/machine)

- process information
- process rules and policies
- social and human elements such as
- process users and stakeholders and their involvement
- various views of the process
- process quality
- employee job satisfaction
- acceptance and use of the system
- environmental restrictions.

These can be classified into the various perspectives of the business process, such as functional, behavioural, informational, operational and organisational perspectives [6]. It is possible to add other perspectives to cover all the aspects of the business process, such as soft, which involves the social and human aspects of the business process, causality and quality, which will be discussed in the workflow reference model perspectives in Section 2.2.6.

### 2.1.3 Business Process Life Cycle

The business process goes through four phases from its initial creation to its continuous improvement [51]: the build phase, drive phase, monitor phase and optimize phase.

- The build phase involves creating the business process and defining all the aspects that are necessary to run the process. This is done by designing, simulating, verifying and deploying the business process. It is important to validate and verify the designed business process before it is put into operation, to ensure that it will operate to achieve the required objectives.
- The drive phase ensures that the execution of the business process is consistent with its definition at the building phase. It also deals with the exceptions that arise during execution.
- In the monitor phase, the business process performance is assessed against the defined criteria of the build phase. Data from the assessment is used to improve the business process.
- The final phase, optimize, involves the ability to continually improve the business process during execution to deal with exceptions or unexpected

circumstances by providing additional resources or altering the business process, for example by skipping non-urgent tasks [51].

### 2.1.4 Business Process Methodology

The business process field is not stable; great effort is needed to organise it and set up standards in several areas to promote its growth. Examination of the business process literature shows that there is no universal methodology that facilitates the development of an organisational business process. The aim of such a methodology is to support the development of the business process and cover its lifecycle by investigating and analysing the business process and maintaining it through its day-to-day operation.

Most existing methods, methodologies or approaches deal with specific aspects of the business process. Kettinger et al. [58] survey 25 methodologies in the context of Business Process Reengineering (BPR), while Aguilar-Saven [14] discusses some of the methodologies that can be used in business process modelling.

Van der Aalst et al. [59] set out four phases of business process management: diagnosis, design, configuration and enactment. The diagnosis phase identifies problems and deficiencies to be rectified in the design of the business process. The third phase implements the designed business process by configuring the required software that supports its functionality. Finally, the enactment phase commences when the business process is executed by using this software.

### 2.1.5 Business Process Modelling Techniques

An organisation achieves its objectives by carrying out business processes, so the business process model can contribute to an understanding, analysis and integration of the activities of an organisation [14]. The business process is considered as a complex structure that involves several elements that should be included in the business process model. Business process modelling is a technique to analyse and model the business processes by transforming knowledge about the business systems into models that depict the processes performed by the organisation [60]. Patel and Hlupic [61] argue that it is not enough to represent the business process in static models that depict the structure of the process with the flow of information between processes; they also suggest the use of dynamic models that present aspects of the business process such as resources and the movement of people and objects, as a possible solution to overcome the problem of inability to predict the result of business process change projects. They use flowcharting as a static modelling technique and the Process Charter package for the dynamic representation. Further, Hommes and Reijswoud [62] support the idea that a business modelling technique should be used to illustrate the dynamic aspects of the process as well as the static features upon which these are based. For their part, Giaglis et al. [63] consider the business process as a complex socio-technical system which requires models to depict its technical and political specifications. Technical specifications require a formal, quantitative, stochastic, documented, adaptable, reusable and objective-driven model. On the other hand, the model should show several alternatives, be a medium of communication among the different users and be easy to use to specify the socio-political requirements.

There is a plethora of methods, techniques and tools for business process modelling. Kettinger et al. [58] review 25 methodologies, 72 techniques and 102 tools in the context of BPR. They aim to establish a framework for a generic BPR methodology and specify the appropriate techniques and tools for each step. This assists in developing a conceptual framework to choose the most suitable technique based on the project characteristics.

In addition, Melao and Pidd [11] present four views of the business process by describing the essential characteristics of each one, the techniques that are used to depict its elements and its difficulties and problems. The first view is the deterministic machine, in which the business process is a group of activities or tasks that are carried out by humans or machines which convert inputs into outputs to achieve certain objectives. Process flow charting, IDEF0 and IDEF3, which are descendants of data flow diagrams (DFDs) with a functional and structured approach, and role activity diagrams (RADs) are used to model this view. The next view is the complex dynamic system that emphasises the interaction and dynamic behaviour of the elements of the business process, both internal and external, to accomplish certain objectives. The techniques required to present the dynamic view are simulation and high-level Petri nets. The interacting feedback loop view deals with the flows of resources from outside by regulating them through policies which are explicit statements of action to be taken to achieve the required result. These actions are based on information. Two types of dynamic system model are used to present this view. Qualitative models, such as causal loops and flow diagrams, are used to depict the structural features of the process, while in quantitative models, the diagrams are transformed into a set of equations that are represented by simulation. Finally, the business process is considered from a social point of view, where the emphasis is on the people with different values, beliefs, expectation and agendas who carry out its activities and tasks. Soft models from Soft Systems Methodology (SSM) are developed to generate debate and learning about how the process should be performed.

Giaglis [64] introduces a framework for classification and review of business process and information system modelling techniques which is based on three evaluation variables: breadth (concerning the modelling goals, such as human understanding and communication, process improvement, process management, process development and process execution), depth (concerned with the functional, behavioural, organisational and informational perspectives) and fit (meaning the projects into which the technique can be fitted). The business process modelling techniques are flowcharting, Integrated Definition for Functional Modelling (IDEF), Petri nets, simulation, knowledge-based techniques and Role Activity Diagramming (RAD), while business information modelling techniques include Data Flow Diagramming (DFD), Entity Relationship Diagramming (ERD), State Transition Diagramming (STD), IDEF techniques (IDEF1x) and Unified Modelling Language (UML).

Finally, Aguilar-Saven [14] reviews several business process modelling techniques and methodologies and establishes a framework to classify them based on two criteria: the purpose of the technique and model change permissiveness (passive and active). The purposes of business process modelling are to learn about the process, to make decisions on the process and to develop business process software. The business process modelling methodologies include Structured Systems Analysis and Design Method (SSADM), Soft Systems Methodology (SSM), Graph with Result and Activities Interrelated (GRAI) methodology and Simulation. On the other side, the business process modelling techniques involves flow charts, DFDs, RADs, Role Interaction Diagrams (RIDs), Gantt charts, IDEF, coloured Petri-nets (CPNs), object-oriented methods and workflow techniques.

Among the above, Melao and Pidd's classification framework is considered superior for several reasons:

- The framework is a general one.
- It is perspective-based.
- It covers almost all aspects of the business process.

These methodologies and techniques are used for modelling the business process and are considered an integral part of the implementation of many information technology (IT) systems; for example, workflow management systems (WFMSs) [11]. Business process modelling also facilities the development of software systems to support the business process [14].

This research is concerned with analysing and modelling the business process as a workflow model. In the following section the workflow paradigm will be reviewed and some workflow models will be discussed.

## 2.2 Workflow Management Systems

### 2.2.1 Introduction

Workflow technology has recently been regarded as one of the main types of IT. It has been estimated that almost $1 billion will be spent on workflow automation and technology to improve customer services. The number of companies using workflow was predicted to rise to 5.8 million in 1998 [65]. Also, the number of commercial workflow systems is increasing [66]. There are almost 250 vendors who provide workflow management systems, including Staffware, COSA Workflow, FlowMark and InConcert.

The increased demand for workflow systems is due to the roles that they play in improving the way that large organizations operate [65]. Workflow systems assist in 'the everyday operation of enterprise and work environments' [66]. In addition, workflow technology assists in defining the business process explicitly, to react to environment change, to manage and control the business process execution and to carry out the various business process activities by using integrated applications from various platforms [67]. Moreover, the use of workflow technology in electronic commerce as a back-end service-provider will shorten the business process time, assign resources efficiently and enhance the enterprise performance [68]. The management of the business process through a workflow system will greatly cut down the administrative costs and enhance service quality [69].

Workflow technology has its basis in a collection of applications for the office such as office automation, image processing, document management, e-mail, groupware applications, databases and computer support. It is a complex system that covers various areas of computer science and management science, such as database management, client-server computing, heterogeneous distributed computing, graphical user interfaces, document management and business practices and re-engineering [67]. In the 1990s, workflow technology was considered a part of BPR, used to improve the business process by automating its activities and tasks, but it had a deficiency as the focus was on the technology, ignoring the human aspects of the business process. However, the need to model and monitor the business processes of many organisations increases the interest in BPM and the use of WFMSs to perform this role [69].

The Workflow Management Coalition (WfMC) [29] is a non-profit organization that was founded in 1993 by a number of vendors to develop common technology and standards for workflow systems. It has published several documents on workflow technology terms and standards to facilitate interoperability among the various workflow systems from different vendors. It defines workflow as 'the computerised facilitation or automation of a business process, in whole or part'. Workflow systems are associated with the business process through assessing, analysing, modelling and implementing its parts and elements. On the other hand, the WFMS manages and controls the automation of the business process and its surrounding environment. WfMC defines a workflow management system as 'a 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'.

WFMSs also can assist in the specification, decomposition, coordination, scheduling, execution and monitoring of business activities, and in documenting and reflecting upon them. Business processes are modelled as workflows (activities) that are executed by a set of agents, which are hardware/software systems or human beings, with the ability to execute a finite set of tasks in an application domain [70]. In addition, Alonso et al. [66] define a WFMS as 'the set of tools used to design and define workflow processes, the environment in which these processes are executed, and the set of interfaces to the users and applications involved in the workflow process'.

Workflow systems are defined as 'systems that help organizations to specify, execute, monitor, and coordinate the flow of the work items within a distributed office environment' [65]. A workflow system consists of two components:

The workflow modelling module, which enables administrators and analysts to define procedures and activities, analyze and simulate them, and assign them to people. There is no formal model for this component, so it is called the 'specification' or 'build-time' module.

The workflow execution module contains the execution interface as seen by end users and the execution environment, which assists in coordinating and performing the procedures and activities. This component is called the 'enactment' or 'run-time' module [65].

WFMSs provide support in three functional areas:

- the build-time functions to specify and model the workflow process and its activities in a formal, computer processable definition, called a process model;
- the run-time control functions to manage and control the execution of the process and to order the sequence of its activities, which is done by a WFMS control software 'engine';
- the run-time interactions with the users and application tools to perform the different activities or tasks [23].

### 2.2.2 Workflow Management Systems Types

WFMSs have evolved through several generations. The first generation is characterised by defining the business process through hardcoding its specifications in the application programs; such systems are used in particular application areas such as image or document management. Second generation WFMSs use a script language to present the process description and can be combined with limited third party tools, but are considered as stand-alone applications. Third generation systems use the interface mechanism to interchange the process definition between the graphical tools that model the process and the workflow engine, which interprets and executes the process specification. There are several WFMSs that implement features of the second and third generations. Finally, the expected fourth generation of workflow systems may become part of the middleware and will provide management services among other services. There will be a trend to standardise the several interfaces and to use a standard process definition language [71].

The products can also be classified according to their support for the various business processes into the following types:

i. Administrative WFMSs

The business processes are defined in detail in the form of steps and their execution is managed and controlled by a set of rules.

ii. Ad hoc systems

Similar to the administrative workflow systems, but dealing with exceptions and unique situations.

iii. Collaborative WFMSs

These focus on the communications among the users in an iterative way to carry out the required steps of the process.

iv. Production WFMSs

These are used to implement the crucial business processes of the enterprise.

In addition to this classification, Alonso et al. [66] introduce one which depends on the underlying technology of the workflow systems; it includes e-mail centric systems (associated with collaborative or ad hoc systems), document centric systems (based on routing documents or forms from one participant to another in some administrative workflow systems) and process centric systems (based on databases, communication mechanisms and a range of interfaces to interact with other applications).

### 2.2.3 The Need for Standardisation

One of the aims of the WfMC is to promote interoperability between the WFMSs by using standards. Standardising of workflow technology is required for various reasons. First is the need to integrate several business processes that are implemented by workflow systems which are also required to be integrated in turn.

Second is the possibility of integrating the best of the market products to construct a workflow system to implement a business process and its activities [72]. WfMC [73] defines several levels of interoperability between workflow products. Each level specifies a set of interfaces and requirements, as follows:

- Level 1 is 'no interoperability'; there is no communication between the workflow systems.
- Level 2 is 'coexistence'; there is no standard for workflow system interoperability but the workflow products have the ability to share the same hardware and software infrastructure.
- Level 3 is 'unique gateways'; the workflow systems use the gateway mechanism to exchange work among them.
- Level 3A is 'common gateway Application Programming Interface' (API); the workflow products use gateways that share a standard API.
- Level 4 is 'limited common API Subset'; the workflow systems are interoperated directly by using a standard API that implements the workflow system's core set of function cells.
- Level 5 is 'complete workflow API'; workflow products are interoperated through a single open API that supports access to all workflow management system operations excluding the specific domain functionality of a workflow product.
- Level 6 is 'shared definition format'; all workflow systems support a shared format of the process description at run time.
- Level 7 is 'protocol compatibility'; communications among the workflow products to exchange information must be standardised by using various mechanisms that provide interoperation.
- Level 8 is 'common look and feel utilities'; the workflow systems use the same user interfaces, in addition to the previous levels of interoperation.

These levels measure the degree of interoperability or conformity between the various workflow products. This will assist in classifying workflow products and determining the level of integration between them.

### 2.2.4 The Workflow Management System Architecture

The WfMC [29] has developed an interface-based reference model for workflow system architecture which supports interoperability between the various workflow products and consists of the major components and interfaces. The main component is the workflow enactment service, which provides the run-time environment in which the process definition is interpreted by one or more workflow management engines, an instance of a business process is initiated and interacts with the various resources to carry out its activities. This component also interacts with other interfaces covering the following areas (Figure 2-1):

i. The import and export of process definitions

   Interface One standardises how different systems exchange process model and organisational information. It is possible to use specialist tools to model and simulate the business process, then use the model to transfer process elements to a workflow definition [69].

ii. Interaction with the client applications

   Interface Two specifies the required functions to allow users to interact with the WFMS. The workflow engine interacts with the client application through this interface that contains the worklist mechanism, which is a queue of work items that are assigned to the user by the workflow engine.

iii. The invocation of software tools or applications

   Interface Three defines the standards to invoke applications or tools for performing an activity.

iv. Interoperability between different workflow management systems

   Interface Four specifies the interaction between the various workflow engines to execute the process and exchange its information by passing the work items among them.

v. Administration and monitoring functions

   Interface Five deals with the control and management of the process through the use of an audit trail facility.

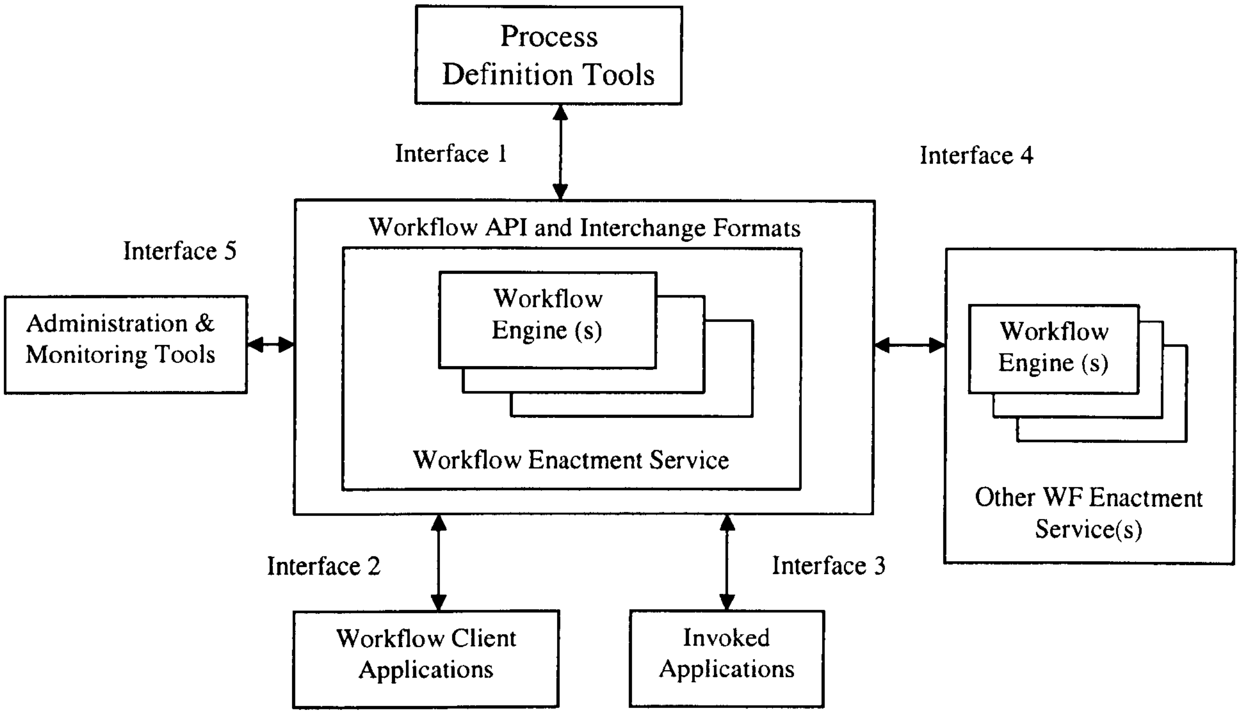



**Figure 2-1: The WfMC Reference Model - adapted from the WfMC workflow reference model**

### 2.2.5 WfMC Workflow Meta-model

Interface 1 of the above WfMC reference model [74] deals with the business process definition by supporting the import and export of the process definition. It includes a common meta-model for the process definition, a workflow process definition language (WPDL) and APIs for the manipulation of the process definition data.

The workflow meta-model is used to define the minimum set of objects, their relationships and attributes within the process definition. The model includes the following top-level entities:

i. Workflow process definition, consisting of:

   a. Workflow process activity

   b. Transition information

   c. Workflow application declaration

   d. Relevant workflow data

   e. Workflow participant specification (organisational model specification)

ii. Workflow participants specification (organisational model specification)

iii. Workflow application declaration

iv. Workflow relevant data

This model is used as a basis for developing any workflow model approach to promote the interchange between the various workflow models.

The research in this thesis concentrates on the process modelling stage. The business process will be modelled in a workflow model form by using one of the various languages that are used to define the process parts and aspects. The workflow model is 'an acyclic directed graph in which nodes represent steps of execution and edges represent the flow of control and data among the different steps' [66].

One of the important areas of workflow technology is the meta-languages that can be used as an integration platform for exchanging the process models and a standard for the selection of an application specific modelling language [75]. Zur Muehlen and Becher [75] compare five meta-languages using the origin and the target domain as well as their capabilities to support the various aspects of the workflow process models such as process, resource, data and invoked applications perspectives and the target of the these meta-languages. These are Workflow Process Definition Language (WPDL), Process Interchange Framework (PIF), Process Specification Language (PSL), Generalised Process Structure Grammars (GPSG) and Unified Modelling Language (UML). They conclude that each of these languages has a different origin and a different target domain. The WPDL, PIF and UML handle all the workflow perspectives, PSL lacks modelling data perspective and GPSG does not deal with resource and invoked applications perspectives. Finally, the WPDL, PIF and PSL are considered as interchange languages while GPSG and UML are modelling languages.

### 2.2.6 The Workflow Reference Model Perspectives

Jablonski and Bussler [38] introduced the comprehensive workflow model, which can be used as a reference model for other workflow vendors to validate the completeness of their models regarding the modelling of the workflow system. It covers almost all of the required aspects that the workflow system should have and implement. This reference model will be used throughout this thesis. A soft perspective on the business process, with several perspectives from the workflow reference model, will be chosen as a framework to compare several commercial and prototypical workflow models in Chapter Three. The framework will also be used in Chapters Four and Five to model the workflow system using SSM and UML. The workflow reference model consists of eleven perspectives, the first five of which are considered as core perspectives, while the others are optional. They are as follows:

**A – Core perspectives**

i. The functional perspective,

which is concerned with the composition of the business process into activities.

ii. The informational perspective,

which deals with the various types of data that are required to execute the different activities of the business process. There are three types of information in the workflow system: workflow control data, which is internal data that maintains the process, activity or other internal status information; workflow relevant data that is used to define the transition conditions between activities and can be accessible by workflow applications for processing; and workflow application data, related to the processing case but inaccessible by the WFMS.

iii. The behavioural perspective,

whose aim is to specify the execution order (control flow) for the activities of the business process. There are three approaches to defining the control flow of the workflows: completely specified, unspecified and specified at run-time 'on the fly'.

At run time, data flow and control flow specifications have to be considered when the execution order of activities is specified.

This perspective includes the control flow constructs that are used to specify the execution order of the workflows. These control flow constructs, such as sequence, parallel branching, conditional branching, join and loops, are used to construct a complex control flow. A WFMS should offer a standard set of control flow constructs that can be used to build more powerful, application-oriented control flow constructs.

iv. The operational perspective,

where activities are implemented by application programs. A workflow modeller has to focus on modelling the right functionality instead of dealing with the technical issues relating to application program integration. Application programs are not referenced directly in the workflow operation specification. An abstraction called 'workflow application' is used to model application invocations. A workflow application specifies the interface to an application program and hides implementation details.

v. The organisational perspective,

which is concerned with specifying who is responsible for performing the different tasks or activities. It consists of an organisation structure that defines organisational objects, which describe the different elements of an organisation and organisational relationships; and an organisation population, which instantiates the organisation structure by specifying the users and their roles and groups.

The activities are assigned to the different agents through synchronisation rules that contain information about the agents required to perform them, who are selected according to criteria such as their skills, their experience or the history of execution of the activity. So for each activity of a business process, an organisational policy is specified, determining eligible agents.

**B – Other perspectives**

vi. The security perspective,

which is concerned with who has the right to access an object. Access is controlled for reasons of security and organisational policy, being allowed when organisational and security issues are both properly satisfied.

vii. The causality perspective,

which handles the business policies, enterprise strategies and business rules that regulate the definition of the workflow type and control the execution of the activities, workflows and tasks. It stores the interrelationships between the business rules and the related perspectives that are affected by them. The dependencies between the workflows and their causalities can be maintained through the WFMS. This will facilitate the integrity of the workflow system.

viii. The history perspective,

which creates an audit trail for each workflow instance. The audit trail contains information about the workflow execution and is used in different contexts: in a system context that deals with computation and recovery, and in an application area context that deals with workflow analysis, revision and reengineering.

ix. The integrity and failure recovery perspective

There are two kinds of failure: semantic failures, which are application area dependent, and system failures, which occur because of technical problems that are caused by erroneous program code, a power failure, a hardware failure, a base service software failure or a network failure.

Semantic failures are of two types: expected and unexpected. They have to be detected and corrected. Failure detection can be implemented in various ways such as using constraints or testing return values. The recovery actions for constraint failures include termination of execution, waiting for constraint fulfilment, starting of a repairing workflow, and semantic rollback. The other type of semantic failure includes exceptions that are unintended and unexpected, such as a division by zero. The recovery procedure is to stop the affected workflow instance and assign it to some agent to rectify the failure.

System failures should be overcome and the system kept in a consistent state by using different means such as transactional mechanisms or distributed transactions.

Semantic and system failures may cause each other in reality.

x. The quality perspective,

which includes several elements related to quality. The workflow specification should be defined by incorporating variables to present these elements. The execution data that are associated with these variables are stored in a log file, then collected and analysed to validate the quality of the workflow specification and to find execution bottlenecks, resource intensive workflow steps and long-running workflows.

xi. The autonomy perspective,

which has three aspects: mobility, distribution and execution threads.

1. Mobility

This deals with off-line execution of a workflow by downloading work and executing it later on. When the user reconnects to the network, the executed work is fed back to the workflow management system. It is important to consider some issues to guarantee consistent processing, such as data and data flow, synchronisation (preventing other agents from executing the task), deadline and availability of workflow application to perform the operation.

2. Distribution

This deals with the ability to extend a workflow type across the boundary between logical domains. There are several problems with distributed workflow execution, such as status monitoring. Distributed workflow execution uses a distributed transaction mechanism to deal with failures. The transactional control should be divided between the logical domains.

3. Execution Threads

Workflow instances run independently of each other as long as they are top-level instances. Subworkflows run within the scope of their superworkflows. So whenever a workflow starts a subworkflow, this is automatically executed within the scope of its superworkflow, which means that a superworkflow waits until its subworkflows are finished.

After reviewing the comprehensive workflow reference model, seven perspectives are chosen in addition to a soft perspective to form a framework to compare several workflow models and to develop an approach to model the workflow by using SSM and UML. These perspectives include the functional, behavioural, informational, (operational) organisational, which are considered as essential and which any workflow system should have mechanisms to handle [76-81] in addition to soft, causality and quality perspectives. The other perspectives are less important and are used to complement the main functionality of the workflow system. They are considered as advanced features that can be incorporated into the workflow system to deal with specific issues.

### 2.2.7 Workflow models

There are several commercial and prototypical workflow models that can be classified as activity-based or communication-based. Activity-based models involve the following concepts:

- A *workflow* consists of a set of tasks.
- A *task* is part of a workflow that can be carried out by a human, a software program or both.
- *Manipulated objects* include documents, data records and images.
- A *role* is a human skill or a software process that carries out a task.
- An *agent* is a human or software system to fill a role and carry out a task.

Communication-based workflow models depict the action in a workflow as a workflow loop that consists of four phases:

- Preparation phase: a customer requests an action to be carried out by a performer.
- Negotiation phase: both customer and performer agree on the action to be carried out and specify the satisfaction terms.
- Performance phase: the action is carried out by the performer.
- Acceptance phase: the customer is satisfied or not with the action.

A business process is represented by a network of connected loops in which a workflow loop invokes another loop to complete or make progress in the main loop [16].

In the following some activity-based workflow models such as FlowMark, WIDE, COSA, InConcert, SAP, WorkParty, Staffware and the Action workflow model which is considered as communication-based, as opposed to prototypical workflow systems such as Domino and OfficeTalk are reviewed.

FlowMark was developed by IBM Corporation. Its metamodel, which is activity-based [82], involves the following syntactic elements: activity, control connector, transition condition, container, data connector, exit condition, synchronisation condition and task. There are graphical representations for activities, control connectors, transition conditions, containers, data connectors, exit conditions and synchronisation conditions. FlowMark uses a set of modelling constructs to model the organisation structure: levels, people, roles, organisations with people-roles, and people-organisation relationships. FlowMark assigns tasks to human agents through the role concepts and the historical information of the workflow. It also supports automatic agents by designating program or process activities as automatic [83].

Another workflow model is WIDE (Workflow on Intelligent and Distributed database Environment) which is an ESPRIT project of the European Commission [84]. Its metamodel is also activity-based. Each task is defined by the following components: precondition, action, postcondition, role constraint and exception handler. It includes several connectors such as total fork, total join, conditional fork and partial join. WIDE supports integration with external databases and provides a strong transaction support.

COSA ("COmputerunterstutzte SAchbearbeitung", computer-supported case-handling) is a workflow management system from Germany [85] which has several modelling elements: cases, workflows, activities and processes. The elements COSA uses to model organisational structure include users, groups, group types, group membership, active substitutions, substitution plans and distribution [86].

InConcert is the workflow management system of XSoft. It uses a set of modelling elements: job (workflow), task, role, document, event and action. InConcert uses a role concept to assign tasks to users or groups of users or to application programs [87, 88].

SAP is part of the R/3 system from the German company SAPAG. It has a set of modelling elements such as workflow, step (work item), task, object, application and organisation. R/3 has a module (HR-ORG) which is used to build an organisation structure. This module models users, positions, workplaces, groups and their relationships to tasks and resources. The role concept is used to assign tasks to users [38].

WorkParty is a workflow management system of Siemens Nixdorf Informations Systeme AG [89]. WorkParty is based on the idea of folders which migrate through an office. The modelling elements are tasks, workflow, variables and references to folder-external data. An organisation structure is not part of the WorkParty workflow model, but it is available through a separate tool, Organisation Resource Management (ORM). WorkParty accesses ORM to find the users responsible for a task. ORM has the modelling elements such as employee, function, position, authority and organisational unit. The position concept is used to assign tasks to users. A position represents the workplace for an employee; it has a set of assigned tasks for which the employee occupying it is responsible [90].

Staffware is a form-based workflow management system of Staffware Plc [91]. The process is defined by using the concepts of procedure and form. The procedure represents the organisation rules or policies to deal with the forms. A Staffware procedure consists of steps (Document for Response and Action). Each step includes an addressee, a document, an action and optionally a deadline. The addressee is a role in an organisation structure with certain skills and authority. The action part specifies the next action of the document when the current addressee (user) finishes with it, usually the process of another step or stop. The addressee will complete part of the document and indicate that he has completed the step. Then the document is circulated among the addressees to finish their steps. The Staffware system keeps information about the various users, their roles and their current workloads that are used to assign work items to roles. The addressee has a notepad, personal database, and a mailbox for receiving work items and emails.

Another system type is Action workflow, developed by ActionTech [39]. Its meta-model is communication-based. A task is considered as a closed loop with four phases: proposal, agreement, performance and satisfaction. There are three types of relationships between task loops: a loop may be a part of another loop, trigger another loop or resolve (make decisions to choose another action in) another loop. Action has three connectors: conditional, splitter and rendezvous to decompose the trigger relationship. It includes the concepts of roles and identities to model the assignment of tasks [92].

The research prototypes of the workflow systems are Domino and OfficeTalk.

Domino is a research effort of the Gesellschaft fur Mathematik und Datenverarbeitung (GMD) in Germany [93]. It uses a set of modelling elements such as procedure, role, action, form, a need relationship and a produce relationship. Domino offers several elements to model an organisation, such as organisational units, projects, jobs and people. Also, there are many relationships to support the organisation structure; for example, belongs to, holds, member of, and supervisor of. Domino uses roles to assign tasks. Role language expressions are predicates over the modelled organisation structure [94, 95].

OfficeTalk is a research effort conducted in Xerox PARC (Palo Alto Research Centre) [96]. It defines an office as a set of procedures. It has the following modelling elements: activity, procedure, task, repository, actor, role, goal and relationships between them. Organisation structure is not defined in OfficeTalk, but a role concept is used to assign activities. Roles are related to activities by a performer relationship. An actor has to play this role to be able to work on an activity [97-99].

### 2.2.8 Workflow modelling techniques

Workflow systems can be modelled using various techniques, which include Petri nets, UML, workflow graphs and workflow evolution.

- Petri nets is a popular technique to model workflow systems. Its advantages include these: Petri nets are graphical; they have formal semantics; they can express most routing constructs; there are several techniques to analyse Petri nets and they are vendor-independent [100]. A Petri net can be used to model the routing of cases (workflow instances). It consists of transitions to model tasks and places and arcs to depict causal dependencies. A place can be considered as a condition, such as pre- and/or post-condition for a task [76]. There are variants of Petri nets that are used to model workflow, including Workflow Nets, Information Control Nets, INCOME/WF, FunSoft nets. Each version of Petri nets deals with a specific aspect of the workflow system. The Object Coordination Net (OCoN), a combination of object-oriented and Petri nets, is used to model the system behaviour with UML structure diagrams. It is used for workflows that are implemented in complex distributed environments and B2B systems involving more than one organisation [76]. Workflow Nets (wf-net) are a version of Petri nets used to model workflow process definitions. A workflow net has one source place and one sink place, because each case (workflow instance) in wf-net is created when it enters the WFMS and deleted once it is finished. All the nodes of a workflow should be on some path from source to sink. A wf-net deals only with the behavioural perspective. The other workflow perspectives can be added by using high-level Petri nets [76].
- UML activity diagrams can also be used to model workflow systems. They have graphical features which can express most routing constructs in a

vendor-independent manner. However, the syntax and semantics of the activity diagram do not have a formal basis [101]. Therefore Eshuis and Wieringa [102] define two semantics for the activity diagram for workflow modelling. The first is a high-level semantics based on the Statemate semantics of statecharts, while the second is low-level, intersecting both the behaviour of a workflow system and the informal OMG semantics of UML state machines. Van der Aalst et al. [76] investigate the strengths and weakness of the Unified Modelling Language (UML) as the standard of object-oriented analysis and design and other Petri net approaches to advanced workflow modelling, which covers essential workflow perspectives, the flexibility and analysis issues of workflow management. The flexibility aspects include constrained flexibility, instance change and instance migration, while the analysis issues involve validation of workflow requirements, verification of the logical correctness of workflow process definition and performance analysis of potential bottlenecks. Several UML diagrams are used to model the various workflow perspectives such as use case, sequence, collaboration, statechart and activity diagrams.

- Workflow Graphs were developed during the WASA project [103]. The workflow graph is based on a directed graph that consists of nodes to depict workflows and edges to model constraints between them. Workflows can be atomic or complex. An atomic workflow does not have an internal structure, while a complex workflow includes a set of workflows which can be atomic or complex. Each workflow has relationships with its descendant workflows. The behavioural perspective is represented by control flow constraints and a start condition. A workflow graph does not support loops, replacing them by recursion. Roles and agents are used to model the organisational perspective. The informational perspective is depicted through class definitions, objects and parameters. Data dependencies between workflows are defined by data flow through mapping between input and output parameters of workflows [76].

- Workflow Evolution was developed by Casati and colleagues [104]. It involves the use of the workflow language that was developed in the WIDE project [105], including the following concepts: tasks, which are described by a name, a textual specification, a set of information details and a set of roles; a hierarchy of tasks or sub-processes; and business transactions. Tasks and sub-process are joined by fork and join connectors that are marked with transition conditions. Business transactions are used to model complex processes [76].

## 2.3 Chapter Summary

This chapter has reviewed the business process as an important aspect of an organisation and the use of workflow systems in modelling and implementation of the business process. Several conceptual models of WFMSs and workflow techniques were discussed. There was discussion of the literature and background of business processes in general and workflow systems specifically as a way to model the business process.

In the following chapter, several WFMS models are compared in the context of the framework of workflow perspectives. This will assist in identifying the capabilities of the models to cover the various perspectives, which will facilitate the development of a new approach to model all the perspectives of the workflow framework.

# Chapter 3: Comparison of Workflow Models

The workflow models that were discussed in Chapter Two will be reviewed in detail in this chapter and compared with each other. The comparison will be based on a framework consisting of some chosen perspectives from the workflow reference model that was mentioned in Chapter Two, Section 2.2.6, with the addition of a soft perspective to cover the soft aspects of the business process as defined in Chapter Two, Section 2.1.2. The aim of this comparison is to discover the capabilities of these models to cover these perspectives and identify their deficiencies. The results of the comparison will help to develop a new approach in which Soft Systems Methodology (SSM) will be used with other modelling techniques such as Unified Modelling Language (UML) to model a business process as a workflow system in accordance with the perspectives of the framework.

## 3.1 Workflow Perspectives Framework

The workflow perspectives framework is considered as the representation of the various aspects of the business process that should be covered by the workflow model or system; thus, the suggested workflow perspectives framework consists of the soft perspective with several perspectives of the workflow reference model of Jablonski and Bussler [38]. Although these perspectives do not cover the whole workflow domain, the most important perspectives for modelling workflow systems are addressed. The perspectives of the workflow framework are as follows:

**The soft perspective** deals with the human aspects of the business process. It involves soft aspects such as the users' involvement, different points of view, the acceptance of the resultant system in use, and employee job satisfaction. For the workflow products it is possible to include the involvement of the users in the modelling of the business process and the available facilities that assist the user in carrying out the assigned work items. This supports the acceptance of the system and uses it to perform the users' tasks in a way that promotes job satisfaction. The soft perspective also involves handling the exceptions that arise during the execution of the business process.

**The functional perspective** is concerned with the activities that have to be carried out during a workflow execution. Therefore, it defines what has to be done.

This perspective deals with the decomposition of activities into constituent units which represent the hierarchical structure. The business process can be decomposed into workflows or activities that are then decomposed into tasks [76]. The business process is thus defined in a hierarchical manner: process -> workflows (activities) -> tasks.

**The informational perspective** handles the workflow data that can be classified into control data and production data. The control data is related to the functionality of the workflow management system (WFMS), such as the parameters that are used for routing. The production data consists of information objects such as documents, forms and tables that are required for processing the activity or task [76]. There are also production data that are relevant for the WFMS. These data are managed by a document management system or database management system (DBMS) [38]. The data contain a set of parameters, such as *in*, *out* and *inout* parameters, along with local variables, which are used to store intermediate results. All parameters and local variables have type features. There are two types of data: the elementary data types such as *integer*, *real* and *string*, and composite types including *record* and *array*. Further, they have scope.

**The behavioural perspective** deals with the order in which the activities or workflows are carried out, which means the control flow of the workflow system. It is concerned with the routing of activities or workflows during the workflow process definition execution [76]. There are several control flow constructs, the basic ones being: sequence, conditional branching, parallel branching, join and loop.

**The operational perspective** deals with the implementation of operations that are carried out by the resources (agents) and applications. These operations are used to process the application data [76]. This perspective specifies the application programs to be used in workflow operations. The application programs are not referenced directly, however: an abstraction is used to model application invocations, called workflow applications, by specifying the interface to an application program without including implementation details. A workflow application includes parameters, constraints and properties. There are several workflow applications available to implement a workflow operation, the choice of application being based on the required properties [38].

**The organisational perspective** covers the organisational structure and its population of agents that carry out the various activities and tasks. The organisational structure is represented by using *role*, *group* and other concepts that are associated with responsibility and availability of agents. The activities or tasks are assigned to the various agents through role resolution [76]. This perspective includes organisational modelling elements, assignment strategies (organisation policies), synchronisation rules to define who actually carries out the task from the eligible users, and a notification mechanism (e-mail or worklist).

**The causality perspective** handles the business policies, enterprise strategies and legal business rules that regulate the definition of the workflow type and control the execution of the workflows, activities and tasks. This perspective stores the interrelationship between the business rules and the related perspectives that are affected by them. The dependencies between the workflows and their causalities can be maintained through the WFMS, thus facilitating the integrity of the workflow system [38].

**The quality perspective** includes several elements that are related to quality. The workflow specification should be defined by incorporating variables to present these elements. The execution data that are associated with these variables are stored in a log file. These data are collected and analysed to validate the quality of the workflow specification and to find execution bottleneck, resource intensive workflow steps and long-running workflows [38].

## 3.2 Workflow Models

In this section, models of some WFMSs are reviewed in detail, using the above workflow perspectives framework. These include both commercial and research or prototype WFMSs, chosen for the following reasons:

- These WFMSs are well known. They are used in several application areas; for example, industry, office environments and the financial sector.
- There is relevant documentation available. Although some of them have just one or two technical papers or reports describing their use, others are covered by several books and reports.
- They cover most aspects of workflow technology.

The description of these models is not intended to be comprehensive. It is based on the available information at the time of the research and as far as possible on the conceptual level of the WFMSs. These models include Action, COSA, FlowMark, InConcert, Staffware, SAP, WorkParty, Domino, OfficeTalk and WIDE.

### 3.2.1 Action Workflow

This product of Action Technologies focuses on coordination. It is a database-based system in which databases are used to store information about the specifications of workflow and business process and the histories and states of the workflow and business process during the execution time. The Action Workflow Enterprise Series provides several tools to define, control and manage a business process.

In the soft perspective, the Action workflow system allows users (performers) to negotiate the work to be done in the negotiation phase; if they reach an agreement with the customer, they are then committed to performing the work in the performance phase. The workflow technology provides user support by:

- Notifying users about actions that are required to be carried out;
- Providing users with tools and information to perform a task;
- Managing the process through reminders, alerts, follow-ups, etc;
- Giving users an overview of the locations of tasks in the overall process, both dynamically and by retaining records of workflow history and providing access to them.

Finally, the work is evaluated by the customer, based on a condition of satisfaction in the acceptance phase.

The process in the functional perspective is modelled as a workflow loop. The workflow may consist of several loops that are required to accomplish the work. The workflow loop is characterised by four phases, two participants and a cycle time. The phases are *preparation* or *proposal*, *negotiation*, *performance*, and *acceptance* or *satisfaction*. The participants include customers and performers. The Business Process Maps (BPM) technique is used to model the business process as workflow loops.

In the informational perspective, two types of data are supported: global data for a business process and local data for a workflow loop. The global data can be accessed by all participants in a business process. On the other hand, the local data can be accessed only by the customer or performer of the specific workflow loop. Action workflow supports document-based workflow by providing Action DocRoute, which includes the capability to integrate document management and imaging applications.

The workflow loops that are connected to model a comprehensive workflow or a business process can be connected in a sequential manner, by conditional branching or in parallel in the behavioural perspective. Each workflow loop in a business process can be started several times. Also, it is possible to replace part of a workflow loop by another complete loop. Applications are integrated by using the Standard Transaction Format (STF) mechanism. The application integration gateway is used by applications and the manager tool for communication, which is based on the STF exchange format. Action has a specification language, which is used by applications to call workflow services.

In the operational perspective, the workflow is executed by users or computer-based applications, in which events are scheduled to run automatic actions in workflows to start new workflows, change their state or run applications. Applications are again integrated by using the STF mechanism. Action includes a specification language which is used by applications to call workflow services.

The organisational perspective is presented through the use of the concepts of *role* and *identity*. Humans are depicted as identities that are grouped depending on their organisational function into roles. Action involves a client library that is used to build a user interface for the management of work, but it does not offer a default user interface. On the other hand, the concepts are not very expressive, so that complex rules of responsibility cannot be modelled at all. This reduces the ability to model the departmental settings.

The causality perspective is not supported in the Action workflow system.

Action supports the quality perspective by providing a tool that is used to analyse the business processes by calculating the cycle time and process values, based on workflow phases. It includes reporting facilities and what-if analysis functions to generate statistics of performance and workflow status. Further, a manager tool is used to execute and manage the workflow loops [38, 39, 92, 106].

### 3.2.2 COSA (Computer-supported Case-handling)

COSA was developed by the German company, Software-Ley, in 1990. It is a relational database-based application with a client-server-based structure. It supports the distributed execution of workflows between several COSA systems and 'serverless' computing, in which a user can carry out an assigned task while he is disconnected from the network. COSA provides several graphical tools to define the process and organisation, and to manage and control the execution of a workflow. There is also a language to define cases, workflows, processes, data and organisation structures. Further, a group language is used to assign users to activities. COSA provides a set of Application Programming Interfaces (APIs) that include all functions exported by the COSA server.

COSA supports the soft perspective by providing several facilities to users to carry out their tasks. Workflow users can perform various actions on workflow-related tasks, such as skipping, postponing, resubmitting, rerouting to other users or process managers and reserving. COSA also provides the ability to manually reroute work items to another workflow user. COSA allows a resource to reserve a work item that is displayed on a shared or global worklist for later execution by a user. It provides a facility that allows a user to reject a work item that has been allocated to him and placed in its work queue. When this occurs, the work item will be reallocated to a different resource. COSA supports an enhanced notion of delegation in that it redirects all work items corresponding to a specific task to a specified user. COSA allows users to control the allocation and execution time of work items offered to them, but they are not able to influence the timing or manner in which work items are offered to them. COSA manages the workload across a process through the redistribution function. It directly supports the ability of a user to skip work items allocated to him with the process client application. With COSA Workflow, it is possible to link external work instructions to the process, so that the correct work and help instructions for a specific activity can be automatically displayed when the activity is performed. COSA Workflow includes a personal calendar function in the Worklist Handler, with which users can indicate that they will be absent during a specified period. During this period, COSA Workflow sends tasks to substitutes. Therefore, activities are carried out in time. If a user is absent without notice, another substitute function is available to redirect his tasks to a qualified user. In COSA

Workflow, users and their managers can intervene in the process in many ways. Depending on the authorization level of the user and/or group, the following is possible with cases: performing steps in a different order, performing ad-hoc operations, skipping, repeating or undoing steps, letting a different user perform steps, refraining from performing steps for the time being (postponing, rejecting and reserving), ending processes prematurely, and letting one user perform several steps. Users themselves can create new processes and groups of activities on the fly by means of the 'drag & drop' functionality, without needing a profound knowledge of IT, through the COSA Activity Manager (CAM), which has a repository or library loaded with business rules and standard activities. Finally, COSA handles exceptions by providing the definition of external 'triggers' or events that may be used to start a sub-process. All events and sub-processes must be defined at design time.

In the functional perspective, COSA version 4.1 divides the process modelling into technical and business oriented phases. During technical modelling, function modules are defined for single activities, which are called task definitions. These are adapted to the real working situations by the departments of an organisation which focus on the business issues of the modelling. The process is defined through the use of these concepts: cases, workflows, activities and tasks. The process is specified using a workflow that consists of activities, of which there are several types: *start* and *end* activities determine the first and last activities in a workflow; a *normal* activity is used to specify a task for a user; finally, a *call* activity invokes other workflows, which are called sub-workflows. An activity has several attributes that can be associated with it to define its status, such as priority, a flag to indicate that a priority can be changed, and a time limit for execution. The related workflows are grouped into a case. Furthermore, COSA involves two types of subprocess: a synchronous subprocess, during whose execution the main process stops, to continue when the subprocess has been completed; and an asynchronous subprocess, which can be executed in parallel with the main process.

The data in the informational perspective are restricted and can be represented by using files and unstructured variables. The flow of structured data between activities cannot be specified and they have to be stored either in files or in unstructured variables. Files are defined by a name and a type at definition time. They are circulated between tasks through a referenced name. Variables can be classified into global variables, which are used by all workflows of a case, and local variables, which apply only within workflows.

The Petri nets technique and condition functions are used to model the control flow in the behavioural perspective. It supports all the basic workflow routing constructs; for example, sequence, condition branching, parallel branching, join and loop.

In the operational perspective, activities are carried out by users who are notified by memos or applications through the use of predefined automatic users. Load balancing and scalability are supported through the installation of several servers. In addition, COSA version 4.1 provides a COSA-NOW function, which carries out urgent tasks without delay. Operations can be carried out either on the client, on the COSA server to improve the system performance, or on both simultaneously. The integration of an application program into a WFMS is done through operating-system-based invocation. A Remote Procedure Call (RPC) can be implemented indirectly by starting a program that calls a function. COSA also inquires whether application programs are executed on the COSA server or on the node of the user.

The organisational perspective is described using elements including users, teams, groups, functions, roles, group types, group membership, active substitutions, substitution plans and distribution. A user is defined by a user name, a password and other attributes that specify his responsibility and authority. COSA provides some built-in users such as automatic users, to perform the automatic execution of workflows. A task is assigned to a user through a group language expression. The groups concept is also used to present the organisation hierarchy. A group can consist of other groups (subgroups) and has a manager and several members. Groups can arrange the users of an organisation according to different criteria, so that one group could be functional and another one structural. The substitution plan is used to specify possible substitution in the case that a user is not available for a specific duration time. A distribution is used to define a set of users that workflows can be assigned to. COSA version 4.1 provides an automatic allocation mechanism to assign a task in case there is no match to a user, by passing the task to the supervisor or a substitute selected by COSA.

The causality perspective is supported through the use of the concept of a *case*, into which related workflows are grouped. Thus, the case is indicated when a user starts a workflow, because the data is attached to the case, not to workflows. Also, the business rules and standard activities are stored in a repository or library of the COSA Activity Manager (CAM) that is used for case management and ad-hoc workflow.

COSA provides several tools to manage and control the workflow execution in the quality perspective. A history access tool is used to view the history of a workflow and a statistics tool is also used to collect information about a set of workflows. Further, there is a form-based administration tool with reporting facilities. COSA includes a simulator tool for portraying and analysing the business process model regarding the logical correctness and performance of the process [38, 85, 86, 106-110].

### 3.2.3 FlowMark (IBM)

FlowMark is a database-centred WFMS. All data on the workflow are stored in an object-oriented database. FlowMark has a client-server based structure. It supports the distributed execution of workflows by setting up several FlowMark servers. It is possible to deal with several databases at the same time, so the user can decide which database to work on.

In the soft perspective, during the build-time it is important that processes are defined in quality terms before prototype testing with users can be started. A process can be modelled manually in FlowMark by opening the process list, copying a process template and starting the instance or using the FlowMark API program. This program can be accessed by users via an icon on their desktop, or invoked by one of application programs when a process needs to be started as the result of some business event. FlowMark allows users to deal with the various states of the process instances, for example creating a new process and deleting a completed or finished process. It is important to consider the users' environment and exceptions during the process design phase. During the execution time, the users select the next item they wish to work on from the list, and FlowMark invokes the appropriate application program to assist in carrying out that task. When the task is completed, the server is notified, and the next activity or activities are scheduled to be done. These then appear on the work lists of the users responsible for performing them. Further, FlowMark provides a "filtering at the server" facility in which the user can specify to the server which items in his work list or process list he wishes to see, so that only those items are loaded from the server to the user's workstation. This technique can significantly reduce both response time and server resources for *sign on* and *list refresh*. There are exception conditions that result from special cases or errors made during the work done in these activities. FlowMark handles exception cases by defining an exception scenario to deal with them. If a special condition or exception case arises, it is possible to use an API call to start another process that will deal with the exception.

In the functional perspective, the process is described as a collection of activities that must be performed. It is possible to set up a duration time for the process. An activity within a process can be either a program activity that is executed by a user and has a program associated with it, or a process activity that has associated sub-processes. There are also other modelling constructs; for example, *block* and *bundle*. The former is used to group activities within a process and the latter to start a number of activities of the same type.

In the informational perspective, the data structure is a list of variables which can be predefined by FlowMark for internal use and cannot be deleted from the list, and new data structures that are defined by the process modeller to represent the application area requirements. These data structures are used to model the input parameters, stored in an input data container, and output parameters, stored in an output data container. FlowMark supports neither *inout* parameters nor the distinction between mandatory and optional parameters.

FlowMark supports the sequential execution of activities through control connectors in the behavioural perspective. There are also *enter*, *exit* and *transition* conditions that are used to define arbitrary control flow. The block concept is used to define loops by combining groups of activities within a process. However, it is difficult to judge whether the specification is semantically correct or contradictory.

Activities are assigned to users for execution in the operational perspective through the assignment rules. Assignment depends on several criteria, such as the position of a user within the organisational structure and historical information on the workflow. The *role* concept could be used to assign activities to users. The *program* concept represents the computer-based application that is used to carry out an assigned task. Programs must be registered within FlowMark. Three execution environments for application programs are supported: OS/2, AIX (Advanced Interactive eXecutive) and Windows. It is necessary to specify the application program name and the environment settings. A FlowMark workflow modelling script language is used to specify the application program invocation. There is no direct support for a CORBA or RPC call.

The organisational perspective in FlowMark is specified using a set of modelling constructs such as levels, people, roles, organisations, and relationships among these; for example, people-role and people-organisation relationships. *People* depicts employees in an organisation who carry out tasks. A person is identified by a unique identifier and has a password and additional attributes such as name, phone number, substitution and authority. The term *level* is used to categorise people based on some feature such as educational status. The concept of *role* is used to define organisational roles within a company. A role has a name and members associated with it; these are persons who are assigned to it. *Organisation* is used to describe the structure of an organisation, which can be presented in a hierarchical form that consists of sub-organisations. An organisation has a person who is assigned as manager and other persons as organisation members. A task is assigned to a human agent through the role concept, the historical information of the workflow and assignment rules. A worklist mechanism is used to display assigned tasks to users.

The causality perspective is not supported in FlowMark.

FlowMark provides a process animation tool to support the quality perspective by running the process before putting it into operation [38, 82, 83, 111].

### 3.2.4 InConcert

This is a client-server based workflow management system from XSoft. One important characteristic of InConcert is that everything in it is modelled as an object that can be extended by adding further attributes.

In the soft perspective, users make use of activity graphs to model business processes in the InConcert workflow system. During the execution of a particular job, InConcert allows run-time extension of objects through the InConcert API, which provides a high degree of customization. For example, the process management interface is used for commencing new processes, allocating work, monitoring progress, and removing completed processes. Users utilise the InConcert task organizer, which is a GUI application, to exhibit the tasks assigned to them. These can be organized using a variety of criteria, including priority, deadline, authorisation or role. The user may choose a task to carry out the action to fulfil it. Further, the documents required to perform a task are associated with it, which decreases the burden on the user to track down the necessary information to carry out a task. The process manager can also check the progress or history of events in a process, or of tasks that are overdue, using GUI applications and the reporting capabilities of the underlying relational DBMS. InConcert provides a graphical user interface to allow the process manager to modify a process during execution by adding or removing tasks or dependencies and reassigning roles. This facility is used to react to unanticipated events and exceptions.

The process in the functional perspective is depicted as a job which consists of tasks. A job template, which is built during the modelling of a job and contains all the details needed to run it, is used to initiate a job during the execution time. Documents are related to jobs as well as to tasks. If a document is related to a job, it is automatically part of every task in that job. A task is a piece of work that can be assigned to a user. It includes all information that is required to process it, such as one or more documents and references to application programs. There is also a special form of task, the subtask, which represents a job as a task.

The data in the informational perspective are represented in a document which is an abstract data object that has content, attributes and links to other documents. A document contains attributes that include the required information.

The behavioural perspective in InConcert is modelled though the use of control flow constructs such as *sequence* and *parallel mechanisms.* Conditional branching and case branching are presented through parallel activities with different performance conditions.

In the operational perspective, a task can be carried out by a user who has a defined role or through an application process, which is a routine task that has an automated agent to pass the task to the application program. Applications are integrated by writing a piece of code.

InConcert does not provide concepts or constructs to model an organisation's structure. It uses the *role* concept to assign a task to a user, a group of users or an application program. Users are notified of their assigned tasks in work-to-do lists.

The causality perspective is not supported by InConcert.

In the quality perspective, InConcert includes an audit log that is used to generate reports about the performance of the workflow execution [38, 87, 88, 112-114].

### 3.2.5 Staffware

This system was developed by Staffware Plc. and is used to manage business processes. The first release of Staffware was in 1987. It includes several tools that are used to define, analyse, control and manage the business process.

In the soft perspective, Staffware Global allows users to participate fully in automated processes by accessing the Staffware Workflow server (reference). Staffware users are able to build processes that are adaptive and very responsive to business needs, without knowing what events will happen (Dynamic Process Orchestration-Leading the way in business agility). Staffware allows users to control the allocation and execution time of work items offered to them, but they are not able to influence the manner or timing by which work items are offered to them. Thus, users can carry out any work item that is routed to them and they have the ability to manually redirect queued work items to a specific user. Further, Staffware allows the workflow system to specify the default ordering of work items in a user's work queue, but a user can alter the sequence and properties of work items displayed in his/her queue. This provides users with the ability to choose the next work item that they want to carry out from their work queues. Staffware supports direct allocation of work items to users. The Staffware Process Monitor (SPM) assists business analysts and users by allowing them to manage their business processes. Using a graphical interface, the SPM provides performance metrics and detailed status reports on entire business processes. Also, the Staffware Work-In-Progress Manager (SWIP) provides reports and audits on specific case data and queue status. Staffware's BPM toolkit enables users and technicians to adapt the business process when the business requirements change. Further, Staffware provides a facility to build user interfaces for connecting all of the Staffware process functionality. The Staffware Process Client provides users with Staffware Business Process Management functionality via user interface components supplied by Staffware. These facilities involve Work Queue Management, Forms, Case Start, Audit Trail, and integration to external applications. The Process Client can be configured to suit the preferences and working demands of the individual user. Also, Staffware Process Objects (including Staffware Server Objects) can be used to create the appropriate user interface for different types of user. Staffware provides the work queue manager interface and the forms interface. A queue summary view allows the user to display a graphical summary of workload across queues; there are work item searching and forwarding facilities; and audit trail filtering has been introduced. Participation and redirection have been added, work queue sorting and filtering introduced, and case prediction is supported when operating against a server that has this facility installed. The Staffware Process Relationship Manager (SPRM) supports process automation by providing a flexible environment for managing all enterprise relationships with customer, employees, partners and suppliers (Staffware Process Suite Version 2). Finally, Staffware provides constructs called *event nodes*, from which a separate exception handling path or sequence can be activated when an exception occurs. The process may be either suspended or made to wait until a timeout occurs. If a work item cannot be processed, it is routed to a 'default exception queue' where it may be manually purged or re-submitted.

The process in the functional perspective is defined as a collection of procedures that are modelled as a series of linked steps. There are three types of step: automatic steps that are executed by applications, normal steps, which are carried out by users, and event steps, triggered by external events.

In the informational perspective, Staffware provides the capability to define and use forms within the business process. It supports a flat object hierarchy to reduce the required time to reach desired data and also enhances data grouping.

Staffware supports several routing elements in the behavioural perspective, including *routing*, *branching*, *conditions* and *parallelism*. It includes the following routing constructs: *start*, *step*, *wait*, *condition* and *stop*. *Step* depicts an activity which has an OR-join/AND-split semantic, while the *wait* step is used to synchronise the flow. The *condition* element models conditional routing such as XOR-split. Finally, arbitrary loops are supported.

In the operational perspective, the process steps can be carried out manually by human users or automatically by invoking applications. Staffware provides Enterprise Applications Integration (EAI) and steps-server-based integrations to integrate applications from various technologies. It also includes an EAI Software Development Kit (SDK) to assist developers to build their own new plug-ins.

Staffware provides a tool called the Staffware Process Administrator (SPA) to support the organisational perspective by establishing and managing the users of the system. The tool defines several levels of users, such as work groups, supervisors and individual users with their attributes and their group memberships. The tool can import this information from an external directory service using Lightweight Directory Access Protocol (LDAP) technology. The *role* concept is used to assign steps to an addressee. A role identifies certain skills and authority for an employee. A task is assigned to a user through the Staffware Process Orchestrator (SPO). Staffware process clients provide a user with Staffware business process management functionality through a user interface. These functionalities involve work queue management, forms, case start, audit trail and integration to external applications. Users can configure the process client to suit their preferences and their working demands.

Staffware process suite includes a tool to model, analyse, test and manage the business rules.

Staffware provides several tools and facilities to support the quality perspective. These tools include audit trails to manage the information of the executed activities, a Staffware Work-In-Progress Manager (SWIP) to provide lists of work that is either in progress or outstanding, and a Staffware Process Monitor (SPM) tool to analyse the collected performance information to improve the process model. Further, Staffware provides the following facilities: a test facility to validate new versions of a procedure alongside live older copies, and a prediction facility to infer what work will exist in the future [106, 108-110, 115, 116].

### 3.2.6 SAP Business Workflow

SAP was developed by the German company SAPAG. It is part of Real-time System version 3 (R/3) which is software that connects the whole organisation by a logical transmission of data.

In the soft perspective, the R/3 system is implemented through a framework methodology that involves four steps: organisation and conceptual design, detailed design and implementation, preparation for production, and productive operation. Users are encouraged to be involved in analysing the requirements. It is important to provide efficient training of users so that they can perform their assigned tasks. The system manager will plan and supervise the distributed systems to enhance performance and availability. SAP provides several features to assist users to carry out their tasks. A business workflow feature is used to automate the procedural steps in a business process. It helps users to perform assigned tasks and track work that has been done. There is also a user interface which involves all the required information to complete an assigned task, and help information associated with each task. Moreover, transaction codes facilitate rapid access to the required screen by experienced users, while the ABAP/4 (Advanced Business Application Programming) query facility can be used to assist users in developing appropriate reports for their needs. Finally, the session manager feature can be used to develop a user menu for specific duties. Users can inherit capabilities based on position and authorisation to manipulate the various aspects of the application domain. SAP R/3 provides pre-defined branches which, when an exception occurs, allows an administrator to manually choose one of a set of possible branches.

In the functional perspective, the process is modelled as a workflow consisting of steps, of which there are five types. First, there is an *activity*, which could be a task, an external application, a transaction or a manual action. The next type is a *user decision*, based on a user's manual action. The third type is a *wait step*, which is used to synchronise the parallel branches. A *subworkflow step* is used to model parts of workflows that can be reused in other different workflows. Finally, a *condition step* is used to model the branching of workflow to follow different paths based on certain facts.

Data are not modelled explicitly, because SAP is part of the R/3 system that provides objects. Therefore, the data are obtained from the underlying system or from containers that hold workflow internal data or include references to objects in the informational perspective. Data flow between containers of several workflows is defined by source/sink definitions. Documents are managed by a document management system.

In the behavioural perspective, SAP supports the sequential and parallel execution of steps and includes two kinds of conditional constructs: *if-then-else* and *case*. The creation, synchronisation and termination of workflows are controlled through events.

Steps are carried out either by users or automatically through a computer-based application in the operational perspective. Applications are integrated into SAP through standard invocation methods such as Object Linking and Embedding (OLE) and Object Data Base Connectivity (ODBC).

The organisational perspective is supported in SAP through the provision of a Human Resource Organisation (HR-ORG) module which is used to build an organisation structure. This module models users, positions, workplaces, groups and their relationships to tasks and capabilities. The *role* concept is used to assign steps to users. A work item manager assigns work items to users, manages deadlines and writes history information. The users use the worklist client to check their assigned work items.

The causality perspective is not supported by SAP.

The quality perspective is supported through reporting tools that are used to retrieve information for analysis of the workflow and to provide statistical functions for evaluations [38, 110, 117, 118].

### 3.2.7 WorkParty

This is a product of Siemens Nixdorf Informations Systeme AG. It is a folder-based system and follows the client-server paradigm.

In the soft perspective, WorkParty provides users with a graphical modelling method for designing workflow models and pre-defined process components which are elementary control flow elements. It also offers a substitution relationship between employees and positions. Thus it is possible for an employee to replace another in case of absence. Further, there is a free choice construct that includes tasks and subworkflows, in which the sequence of this execution is defined by the user at run time. During execution, users can accept a task which is executed with or without the application program. In the case of there being no application program, WorkParty provides a description to the users to assist them in carrying out a task. Furthermore, WorkParty includes a desktop client facility which allows users to determine the next task to be performed. Finally, WorkParty supports ad-hoc activities through the insertion of unlimited numbers of new activities at a specific point to deal with exceptions.

In the functional perspective, the process is described as a workflow that consists of subworkflows or tasks. A workflow is stored in a folder which includes documents and routing information. There are three levels of priority that can be assigned to a folder to indicate the importance of work: low, normal and high. It is also possible to set a date to show the validity of the used folder.

The informational perspective is modelled through a folder that contains all the data required by a workflow. There are three types of data: workflow variables, document type and reference to documents. Parameters of tasks are described as *in* and *out* pins that are used to exchange actual values between a workflow and a task.

WorkParty provides several control flow constructs to support the behavioural perspective; for example, *sequence*, *loop* (*while* loop and *repeat until* loop), *alternative*, *free choice*, *fork*, *parallel branching* and *dead end*. 'Free choice' means that all tasks or subworkflows that are associated with this control flow construct have to be executed; however, the sequence of their execution is specified by users at run time. The *dead end* construct is used to abort workflow execution abnormally.

A task is performed by a user with or without the use of an application program in the operational perspective. An address expression which is attached to each task is used to route a task to an employee, who carries it out. The address expression, which consists of five fields, namely employee, function, position, authority and organisational unit, is evaluated during the run-time to specify an employee. These fields can have no values.

The organisational perspective is defined using a separate tool called Organisation Resource Management (ORM). The modelling constructs of *employee*, *function*, *position*, *authority* and *organisational unit* are used to define an organisation's structure. Users are called *employees* and are specified using a first and last name, a phone number, an address and other attributes. Along with the performing employees, there are also: a workflow initiator, who starts a workflow, an employee responsible for a workflow, who manages the execution of the workflow, and a workflow administrator, who makes workflows available for performers. The *position* concept models the workplace of an employee. Positions are categorised by organisational units, which are classified into temporary and unlimited ones. Organisational units can include other organisational units as members to model a hierarchy of organisational units. An *authority* is a combination of authorisations for resource assignments and responsibilities by task assignments. Finally, the term *function* is used to define organisational functions; for example, manager or sales clerk. A function groups positions that have the same authorities. A graphical tool is used to display all folders to an employee who is responsible for execution of the tasks related to them. WorkParty uses the concept of *position* to assign tasks to users. A position has a set of tasks assigned to it, for which the employee occupying this position is responsible.

The causality perspective is supported by the use of a folder that contains workflows and the associated documents. This will facilitate the ability to combine all dependent workflows in one folder to present their dependencies.

WorkParty supports the quality perspective by providing a protocol viewer to view workflow history and run-time environment to manage and control the folders during the execution time [38, 89, 90, 119].

### 3.2.8 Domino

Domino is a research effort of the German company Gesellschaft fur Mathematik und Datenverarbeitung (GMD) that has been running since 1983. Its structure is client-server based.

In the soft perspective, Domino provides a user interface that includes several facilities to assist users to switch to different applications and databases, and most of its elements can be customised. Users can automate work through customisation. Agent and action mechanisms are used to accomplish automation. Actions are integrated into the GUI and activated by users. Agents, which run on the server, can be triggered by certain events. Further workflow applications guide users during the execution of certain routine tasks. This can decrease overheads and mistakes, thus expediting processes. Automatic search capabilities and full-text indexing support are also used to assist users to search for and retrieve significant documents. Due to the widespread use of the system, it is possible for users to obtain training and support.

The process is defined in the functional perspective as a *procedure* which is a net of related actions. An *action*, which describes a task that a user has to carry out, is associated with two forms, one including its input data and the other involving its output data. The internal representation of procedures is modelled using Petri nets in which the places model forms and transitions model actions.

Data are modelled using a *form*, which is a data structure that is moved between users. The fields of a form can be defined by a workflow modeller. Domino does not support parameters and local variables.

The behavioural perspective is not supported in Domino. The execution order of the actions is specified depending on the availability of data (forms). When a form is available, the related action is ready for execution. When the work has been performed, the user sends the form to the next action, depending on the procedural structure.

A task can be carried out by a user who may use an application program, or an automated user, which is represented by the *office role* in the operational perspective.

The organisational perspective is modelled by using several modelling constructs such as organisational units, projects, jobs and people, and several relationships; for example, *belongs to*, *holds*, *member of* and *supervisor of*. A role is used to assign tasks. Domino provides *role language expressions*, which are predicates over the modelled organisation structure that are used to assign the roles to employees during execution time. There are two types of role: roles within a procedure, that is, associated with actions; and roles within an organisation. Each role in the process has an expression in the role language that determines one or more users during the execution time. Also, there are two predefined roles: the initiator role, which indicates the user who commenced the procedure, and the office role, which represents automated users. Domino provides a user interface to manage the procedure. A user interface involves two parts. A main window shows an overview of procedures that indicate the status of each procedure from a user perspective, and other windows are used to manipulate the data of a single form, to track procedures, to start new procedures, to enter data or to select enclosures.

The causality perspective is not supported in Domino.

In the quality perspective, correctness rules for the structural and behavioural aspects of a procedure are defined in Domino [38, 93, 95, 120].

### 3.2.9 OfficeTalk

This is a research effort conducted in Xerox PARC (Palo Alto Research Centre) in the late 1970s and early 1980s. OfficeTalk is implemented as a three-component architecture which consists of a database system to store all office information, a user interface, and a modelling system to feed information into OfficeTalk.

In the soft perspective, OfficeTalk provides facilities such as distributed schedulers, dispatchers, office observer workstations, alerters, a data dictionary synthesizer, change agents, and on-line office modelling, simulation and design facilities. These facilities allow users to manage office information by performing tasks such as text editing, form manipulation, filing, copying, analyzing, transmitting information and assist in coordinating and controlling the flow of forms between users. Users have the ability to select a task if they want to carry out its associated activities. Whenever an activity is completed, the performer of that activity must check for all next activities and schedule them. OfficeTalk-D incorporates an alerting facility which allows the user to define exceptional conditions. Whenever an exceptional condition arises, the system will invoke a user-specified procedure associated with that exception. Alerting can occur with items that assume exceptional values, or upon time constraints that have been exceeded. Further authorized users may use a facility provided by the system which allows them not to follow the default sequence of activities as it is described in the forms flow graph. This *exception facility* provides interface flexibility, which is necessary for an efficient office operation.

OfficeTalk defines an office as a set of office *procedures* that are specified by an Information Control Net (ICN) in the functional perspective. The procedure consists of several *activities* and *repositories*. It can be nested to define several levels of detail. An activity is a piece of work that is carried out by an office worker. A *task* is an individual transaction that is executed through many activities. Finally, it is possible to define *goals* by relating them to activities. This goal will be achieved by executing the associated activity. An activity can be connected to several goals at the same time.

The informational perspective is modelled using a *repository*, which is a container of information that is represented as data, such as forms, files and folders. Each activity is linked to a repository to fetch information from or to store results in.

The procedure structure connects all existing activities and specifies their orders in the behavioural perspective. There are several control constructs; for example, *sequence, fork, join* and *choice and decision*. It is possible to add timing and probability information to the activities of the procedure. The timing information defines the duration of activities.

In the operational perspective, an activity can be an interactive activity, which is performed by a user, or an automatic one, performed automatically through an application. Actions inside an activity are implemented by a script language. Also, applications are integrated into OfficeTalk through the use of a script language.

OfficeTalk does not include modelling constructs to depict an organisational perspective. It uses the *actor* concept to represent a user, who is associated to *roles* by a *player* relationship. A role is associated to an activity through a *performer* relationship. An actor can play several roles. There is also a *report* relationship to connect an actor to another actor to whom he reports. OfficeTalk uses the *role* concept to assign activities. Only one agent is allowed to access an activity. Activities are displayed to a user through a worklist mechanism. The notification is done by a scheduler that looks up a database for ready activities to be executed and subsequently notifies the appropriate user.

The causality perspective is not supported by OfficeTalk.

There are several tools in OfficeTalk to support the quality perspective. A simulation tool is used to simulate real tasks in order to compute the work load, waiting times and queue lengths. This information is used to find out the throughput, turn-around time and bottlenecks. Further, an observer tool manages the various tasks within an office procedure. Finally, an alerter tool, which follows an action-event scheme, is used to execute an action when an event has occurred, to deal with exceptions [38, 97, 121].

### 3.2.10 WIDE (Workflow on Intelligent and Distributed database Environment)

WIDE originated with an ESPRIT (European Strategic Programme for Research and Development in Information Technology) project in 1995. The WIDE project aims to provide the existing WFMSs with advanced technology support through active database support and advanced transaction management in a distributed environment. WIDE has a client-server based architecture.

In the soft perspective, the users provide a set of requirements for the application domain. Further, the users who execute the tasks have full freedom in the way the tasks are executed. An important issue is the commitment of every user to carry out all the tasks assigned to the roles that the users are associated with. There are two ways to assign tasks to users: through the *push* or *pull* models. The pull model gives the user the ability to browse the shared desk and pull the desired task. WIDE provides default exception handlers for all exceptional events defined in the system. The user can define new exception handlers to customize how an exception is to be managed. The default exception handlers are also available when defining new handlers. This means that new handlers can be built as extensions of the default exception handlers, by performing some actions before raising again the same exception to the upper level. Thus, the user has a powerful and flexible mechanism to control the flow in case of problems.

The process is defined through a workflow schema that consists of *supertasks*, *tasks* and *multitasks* in the functional perspective. A task is 'an atomic unit of work in a workflow process model' [84], while a supertask is 'a set of related tasks that can be considered as one unit of work at a higher level of abstraction and be assigned to one responsible agent', and a multitask is a construct that is used to 'allow the specification of the characteristics of similar tasks/supertasks to be executed in parallel' [84]. An instance of the workflow schema is called a *case*.

There are four types of information in WIDE: information that is defined in workflow schema, information that is based on shared databases, documentation elements that are exchanged through the WFMS, and temporal information for specifying time and temporal conditions. First, variables in the workflow schema are visible to the cases using them. Other predefined variables can also be associated with every case and every task for describing their attributes. Then, the persistent data of the workflow schema that is shared by all workflow agents is defined externally, using a database. Third, the documentation elements can be documents, forms, compound documents or dossiers (folders). A *document* is a set of information whose elements are not interpreted by the WIDE WFMS, while a *form* is a set of fields whose contents are interpreted by the WIDE WFMS. A compound document is a set of any fixed combination of documents, forms and other compound documents. A dossier (folder) involves a set of any variable combination of documents, forms, compound documents and other dossiers. Finally, temporal information consists of three concepts: *temporal instant* for specifying a time point, *temporal interval* for describing a time period and *duration* for defining the elapsed time. Data can be classified into global and local data. Global data are shared by all workflows; they are persistent and are stored in databases or files. On the other hand, local data are shared by activities within the same workflow. WIDE also provides an *entity data* type to present external data and their operations and a *form data* type, which corresponds to a record type, to depict the required information.

In the behavioural perspective, WIDE defines several symbols to connect the various tasks and connectors using arrows. These connectors include the following constructs to define the process flow: start and stop symbols, forks or splits [total (AND-split), conditional (OR-split), conditional with mutual exclusion (only one condition can be true)], join [total (AND-join), partial (k-AND-join, some k before this) and cycle (OR-join)] and a wait connector, which is used in a sequence with a condition associated with it.

The work tasks can be carried out by authorised agents that could be human users, software processes for executing automatic tasks, or a combination of users and computers in the operational perspective. The *role* concept is used to assign a task to an agent. The *authorised* relationship is used to connect agents of the organisation with one or more roles that are defined in the process model. Tasks are routed to agents through the domain mechanism, in which an organisation is considered as a number of workflow servers in the network that involves several users. Each agent has a specific name within a specific domain. It is possible to delegate the execution of a task to another agent. There is no information about the application's integration. However, an external event manager deals with the events that are raised by external applications used in the workflow context.

In the organisational perspective, the organisation is described as a collection of agents to represent individuals and other complex structures of the organisation. There are other concepts; for example, an *actor*, which is an individual processing entity that could be human or an automated agent, a *group*, which is a set of actors depending on common organisational aspects, and an *organisation function*, which represents a specific function in the organisation. Three types of agents are created: a case executor, who starts the case, a case responsible, who has a responsibility over the case, and a task executer, who carries out the task. The organisation is divided into several domains, which are areas that have several users connected to them. Several relationships are included in WIDE, such as a *deputy* relationship, to allow one staff member to substitute for another, an *accountability* relationship to model the relationships between positions in a position hierarchy, and an *inclusion* relationship to represent the relationship between teams in a team hierarchy. The assignment of tasks to agents is either made by a scheduler engine of the WFMS to the best actor according to the work assignment policy (automatic assignment), or performed directly by a user (manual assignment). The push/pull model is used to assign a task to a user. In the push model, each user has his own representation of the system, together with the list of tasks that he has to carry out. A task is assigned directly to the user's desk. In the pull model, the system has a desk for some roles that are shared by several agents. The workflow engine inserts tasks into that desk. Then a user can browse the shared desk and pull the desired task. Applications interact with the WFMS through an External Information Systems (EIS) interface. A wrapper module is used to subsume the EIS and the external applications in an object-oriented manner.

The supertask concept, which involves related tasks, supports the causality perspective.

WIDE supports the quality perspective by providing a workflow monitoring support tool which is a graphical tool to monitor the progress of a case. It also provides statistical information on workflow execution for analysing and defining bottlenecks [84, 122-124].

## 3.3 Evaluation of Workflow Models

In this section the workflow models are evaluated with respect to the workflow perspectives framework by comparing them and emphasising their capabilities to cover the various perspectives.

Table 3-1 summaries the evaluation of the workflow models using the workflow perspectives framework.

| Perspective / Workflow Model | Soft perspective | Functional perspective | Informational perspective | Behavioural perspective | Operational perspective | Organisational perspective | Causality perspective | Quality perspective |
|---|---|---|---|---|---|---|---|---|
| 1) Action | Allows users to negotiate the work to be done. If they reach agreement with the customer to perform the work, they are supported by workflow technology. Finished work is evaluated by the customer based on a condition of satisfaction. | Action uses a different method to define the business process through the workflow loop. | Data is modelled through global and local data and documents. | Most of the control flow constructs are supported. | A workflow is carried out by a user or a computer-based application. Standard Transaction Format is used to integrate applications. Workflow services are called using an action specification language. | Roles and identities are used to model an organisation's structure. A user is notified through a user interface. Complex rules of responsibility cannot be modelled. | This perspective is not supported. | Action provides an analyst tool, report facilities, *what if* analysis functions and a manager tool. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2) COSA | Provides several facilities to users to carry out their tasks. It also supports the ability to manually reroute work items to another workflow user. It allows a user to reject an allocated work item from the work queue. It supports delegation in that it redirects all work items corresponding to a specific task to a specified user. | COSA provides all the elements to model a business process (case, workflow, activity and task). | Files and unstructured variables that can be global or local are supported. | Petri nets technique and condition functions are used to depict the control flow. | A task is performed by a user or an application. Further, the application program is integrated through operating system invocation, and remote procedure call can be implemented indirectly. | COSA involves expressive concepts to model an organisation structure. A group language is used to assign tasks to users, who are notified through memos. | A case concept is used to group related workflows. COSA Activity Manager manages a case and ad-hoc workflow. | There are several tools to maintain quality, such as a history access tool, a statistics tool, a form-based administration tool and reporting facilities, and a simulator tool. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | It allows users to control the allocation and execution time of work items offered to them. It manages the workload across a process through the redistribution function. It links external work instructions to an activity. It provides a substitution feature. It handles exceptions by providing the definition of | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | external 'triggers' or events. | | | | | | | |
| 3) FlowMark | A process can be modelled in FlowMark manually or using the FlowMark API program. It allows the users to deal with the various states of the process instances. It provides a "filtering at the server" facility. It handles exception cases by defining an | FlowMark involves most of the elements to model a business process. | It provides predefined variables and supports the defining of new data structures. | Parallel and join constructs are not supported. | An activity is assigned to a user through assignment rules. A user can use an application to carry out an activity. An application program is invoked by using a script language. | It involves several concepts to model an organisation structure. Several criteria are used to assign a task to a user. A worklist mechanism is used to display assigned tasks. | This perspective is not supported. | It includes an animation tool to analyse the process. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | exception scenario to deal with them. | | | | | | | |
| 4) InConcert | Users model business processes using activity graphs. It allows run-time extension of objects through the InConcert API, which provides a high degree of customization. Users utilise the task organizer GUI application to exhibit the assigned tasks. The documents | It uses a new concept, a job, with tasks to model the business process. | A document is used as a media for the required data. | It supports only sequence and parallel. | A task is performed by a user, an application process or an application program. The applications are integrated through a script language. | There is no modelling construct for an organisation structure. It uses a role concept to assign tasks. Users are notified through work-to-do lists. | This perspective is not supported. | An audit log is used to generate reports about the workflow performance. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | required to perform a task are associated with it, which decreases the burden on the user. The process manager can check the progress or history of events in a process or of tasks that are overdue using GUI applications and the reporting capabilities of the DBMS. It provides a graphical user interface to allow | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | the process manager to modify a process during execution. | | | | | | | |
| 5) Staffware | It allows users to participate fully in automated processes. Users are able to build processes that are adaptive and very responsive to business needs. It allows users to control the allocation and execution time of work items offered to them. Users have the ability to | It introduces new concepts to model a business process, such as procedures and steps. | It supports forms, a flat object hierarchy and data grouping. | It provides all the basic constructs of the control flow. | A step is performed by a user or an application. Enterprise Applications Integration steps-server-based integration is used to integrate applications from various technologies and the EAI software | Several constructs are used to model an organisation structure. The *role* concept is used to assign steps to addressees through a user interface. | It involves a tool to handle the business rules. | Has several tools and facilities to support quality aspects: audit trail; Work-In-Progress Manager; Process Monitor; test and prediction facilities. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | manually redirect queued work items to a specific user. Users can alter the sequence and properties of work items displayed in their work queue. It provides several facilities to manage, control, asses the performance of the business processes and to adapt them when business requirements change. It | | | | development kit is used to develop new plug-ins. | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | provides event nodes, from which a separate exception handling path or sequence can be activated when an exception occurs. | | | | | | | |
| 6) SAP | R/3 system is implemented through a framework methodology that involves four steps. Users are encouraged to be involved in analysing the requirements. It provides several | Workflow and step concepts are used to model a business process. | Data are moved between workflows through the use of containers and documents. | Only sequence and parallel constructs are supported. | A step is carried out by a user or a computer-based application. Applications are integrated through standard invocation methods such as OLE and | The HR-ORG module is used to model an organisation structure. A work item manager assigns steps to users who use the worklist client to check the | This perspective is not supported. | Reporting tools are used to manage the quality. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | features to assist users to carry out their tasks, such as a business workflow feature, a user interface which involves all the required information to complete an assigned task, help information associated with each task, transaction codes facilitator, an ABAP/4 query facility, and a session manger. It provides pre- | | | | ODBC. | assigned steps. | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | defined branches to handle exceptions. | | | | | | | |
| 7) WorkParty | It provides the users with a graphical modelling method for designing workflow models and pre-defined process components. It also offers a substitution relationship between employees and positions. There is a free choice construct that | Three concepts, workflow, subworkflows and tasks, are used to model a business process. | A folder is used as a container of the required data. Variables and parameters are supported. | It supports most of the basic control flow constructs. | A task is carried out through a user who can use an application program. An address expression is used to route a task to a user. | An organisation structure is not part of WorkParty, which uses a separate tool: organisation resource management. A position concept is used to assign a task to a user, who deals with folders through a graphical tool. | A folder involves dependent workflows and associated documents supporting the causality perspective. | A protocol viewer is used to manage and control folders during execution time. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | includes tasks and subworkflows. The sequence of this execution is defined by users at run time. It includes a desktop client facility which allows users to determine the next task to be performed. Users can accept a task which is executed with or without the application program. It supports ad-hoc | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | activities. | | | | | | | |
| 8) Domino | It provides a user interface that includes several facilities to assist users to deal with different applications and databases. Users automate work though customisation. Automatic search capabilities and full-text indexing support are used for searching and retrieving significant documents. It is possible for the | A process is defined in terms of procedures, actions and tasks. | A form is used to present data. | It does not provide control flow constructs and the routing of the form is determined during the execution time by a user. | A task is performed by a user or an automated user, which is an office role. | Several constructs are used to model an organisation structure. Tasks are assigned using the *role* concept. A process is managed and controlled through a user interface with two window types. | This perspective is not supported. | Correctness rules are defined to validate the structural and behavioural aspects of a procedure. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | users to obtain training and support. | | | | | | | |
| 9) OfficeTalk | It provides several facilities to manage office information. Users can select a task to perform an activity upon it and schedule the next activities. OfficeTalk includes an altering facility to define exceptional conditions. | An office is defined as a set of office procedures, activities and repositories. | A repository is considered as a container holding the required data in its various formats: forms, files and folders. | All the basic control flow constructs are supported, except *loop*. | An activity can be interactive (carried out by a user) or performed automatically. A script language is used to integrate the applications. | It does not provide modelling constructs for an organisation structure. The *role* concept is used to assign activities. Only one user is allowed to access an activity. Activities are displayed to users through a worklist mechanism. A | This perspective is not supported | Several tools are used to manage quality, such as a simulation tool, an observer tool and an alerter tool. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | scheduler notifies a user to execute the assigned activities. | | |
| 10) WIDE | The users provide a set of requirements for the application domain. Users have full freedom on the way the tasks should be executed. WIDE provides default exception handlers for all exceptional events defined in the system. | A workflow schema is used to define a business process that involves supertasks, tasks and multitasks. | Four types of data are supported: workflow schema information, shared database information, documentation elements and temporal information. | Most basic control flow constructs are supported. | A task is carried out through an authorised agent that could be a human user, a software process or a combination of a user and a computer application. A wrapper module is used to integrate applications | It includes several constructs to model an organisation structure. A task is assigned to an agent through a scheduler engine according to the assignment policy for automatic assignment, or directly to a human user. The | The *supertask* concept combines related tasks. | WIDE provides a workflow monitoring support tool and statistical information on the workflow execution for analysis. |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | into WIDE. | push/pull model is used to assign a task to a user. | | |

**Table 3-1: Summary of workflow model evaluation**

To compare the various workflow models quantitatively, points may be awarded according to the following schema:

- 4 points: all the features of the workflow perspective are represented;
- 3 points: more than half of the features of the workflow perspective are represented;
- 2 points: at least half the features of the workflow perspective are represented;
- 1 point: less than half of the features of the workflow perspective are represented;
- 0 points: the perspective is not supported by the workflow model.

The various workflow models were assessed using the above schema, giving the results shown in Table 3-2.

| Workflow perspectives / Workflow model | Soft perspective | Functional perspective | Informational perspective | Behavioural perspective | Operational perspective | Organis-ational perspective | Causality perspective | Quality perspec-tive | Total |
|---|---|---|---|---|---|---|---|---|---|
| Action | 3 | 3 | 3 | 3 | 3 | 1 | 0 | 3 | 19 |
| COSA | 4 | 4 | 3 | 4 | 4 | 4 | 3 | 4 | 30 |
| FlowMark | 2 | 3 | 3 | 2 | 3 | 4 | 0 | 2 | 19 |
| InConcert | 3 | 4 | 2 | 2 | 3 | 2 | 0 | 1 | 17 |
| Staffware | 4 | 4 | 3 | 4 | 4 | 3 | 2 | 4 | 28 |
| SAP | 3 | 4 | 3 | 2 | 4 | 4 | 2 | 3 | 25 |
| WorkParty | 3 | 4 | 4 | 3 | 2 | 3 | 2 | 2 | 23 |
| Domino | 2 | 4 | 2 | 0 | 2 | 4 | 0 | 1 | 15 |
| Office Talk | 3 | 4 | 3 | 3 | 3 | 2 | 0 | 3 | 21 |
| WIDE | 3 | 4 | 4 | 3 | 3 | 3 | 2 | 3 | 24 |

**Table 3-2: Workflow models comparison**

A 'total' column is added to the table to show the scores for each workflow model and so provide a rough means of comparing them. There is no workflow model that covers all the various workflow perspectives. However, the COSA model covers most of the workflow perspectives, although it still needs some improvements in the informational and causality perspectives. Further, there are some workflow models that are not far behind COSA; these are Staffware, SAP, WIDE and WorkParty.

As already mentioned, these workflow models include both commercial workflow systems and research or prototype ones. COSA leads all the workflow models. On the other hand, WIDE as a research workflow model leads the other two research workflow models, Domino and OfficeTalk.

## 3.4 The SSM and the workflow models

The reviewed workflow models have used various approaches to model the various perspectives of the workflow model. Most of them do not consider the soft aspects in the beginning but they provide some facilities to handle them in the implementation stage. On the other hand this research introduces a new approach, SWfM, to improve the workflow modelling through involving the soft aspects of the organisational business process. The SWfM approach deals with the organisational business process and its soft aspects in the organisational business process investigation stage through using the Soft Systems Methodology (SSM) which includes some techniques to handle these aspects. Soft Systems Methodology (SSM) helps to study and investigate the organisational business process through encouraging the people in the organisation to be involved in the study and considering the various points of view. These promote the involved people to accept and use the resultant system to carry out their tasks and achieve the job satisfaction of their employees.

Soft Systems Methodology (SSM) handles some of the workflow model perspectives. These perspectives involve the soft, functional, behavioural, organisational and quality perspectives which will be called the SSM basic perspectives. Each perspective is handled by one or more of the SSM techniques. This can be reviewed in Section 4.2.2 in Chapter 4.

In the following the various workflow models are reviewed to define the perspectives that need to be improved through using the SSM techniques.

| Workflow models | SSM role |
|---|---|
| 1) Action | All the SSM basic perspectives need to be improved through using the required SSM technique. |
| 2) COSA | COSA covers all the SSM basic perspectives but it needs to consider the soft aspects from the beginning not in the implementation stage only. |
| 3) FlowMark | All the SSM basic perspectives need to be improved through using the various SSM techniques. |
| 4) InConcert | All the SSM basic perspectives except the functional one need to be improved through applying the required SSM techniques. |
| 5) Staffware | Only the organisational perspective needs to be improved through using the required SSM techniques. |
| 6) SAP | The soft, behaviour and quality perspectives need to be improved through using the required SSM techniques. |
| 7) WorkParty | All the SSM basic perspectives need to be improved through applying the required SSM techniques. |
| 8) Domino | The soft, behaviour and quality perspectives need to be improved through using the required SSM techniques. |
| 9) OfficeTalk | All the SSM basic perspectives except the functional one need to be improved through applying the required SSM techniques. |
| 10) WIDE | All the SSM basic perspectives except the functional one need to be improved through applying the required SSM techniques. |

**Table 3-3: Summary of the role of SSM with workflow models**

Business improvement requires defining some aspects that are related to the business process such as defining an organisation's strategic goal or purpose, defining the focus and the outcome of the business process, studying the business process, establishing a control mechanism of the business process and defining a standard for the business process [50]. All these aspects can be handled through using the Soft Systems Methodology (SSM) with its various techniques. First, the organisation is investigated and modelled through using the rich picture technique which depicts the organisation and its surrounding environment. Then a new vision of the organisation can be obtained through the involvement of the organisation stakeholders. This vision can be formed in the root definition which can be modelled by the conceptual model that represents the required activities to achieve the vision of the root definition. These activities can be compared with the current business process if it is available to identify the missing activities. Also these activities include some activities to monitor and control the other activities. Finally, the new or improved business process can be implemented using any technology such as workflow technology to improve the productivity of an organisation.

## 3.5 Chapter Summary

In this chapter, a workflow perspective framework which involves a soft perspective and a subset of the workflow reference model of Chapter Two has been developed. Several workflow models from both the commercial and research areas were reviewed in detail and compared using the workflow perspectives framework to find out their capabilities to deal with and cover these perspectives. According to the comparison results, COSA leads all the workflow models, while WIDE leads the research or prototype models.

The general aim of this chapter was to discover the usefulness and practicability of the framework and to measure the capabilities of the workflow models to cover these perspectives. This framework will be used in Chapter Four as a basis to investigate the capabilities of Soft Systems Methodology (SSM) to handle these perspectives by introducing a new approach based on SSM with Unified Modelling Language (UML), which will attempt to cover the various perspectives of the framework and provide suitable techniques to deal with them.

# Chapter 4: The Soft Workflow Modelling (SWfM) Approach

This chapter introduces the thesis approach of Soft Workflow Modelling (SWfM), based on using Soft Systems Methodology (SSM) to study the organisational business process and model it as a workflow system to support the business process. First, the SWfM framework and its elements, SSM and Unified Modelling Language (UML), are presented. Then the steps, techniques and notations of the SWfM approach are covered in detail.

## 4.1 The SWfM Framework

The framework of the SWfM approach includes the use of SSM to investigate the organisational business process, to address the soft perspective and to verify the methodology's capability to model the organisational business process as a workflow system. Other methods or techniques such as UML may be integrated into SSM to handle issues that cannot otherwise be covered. Figure 4-1 outlines the SWfM framework:

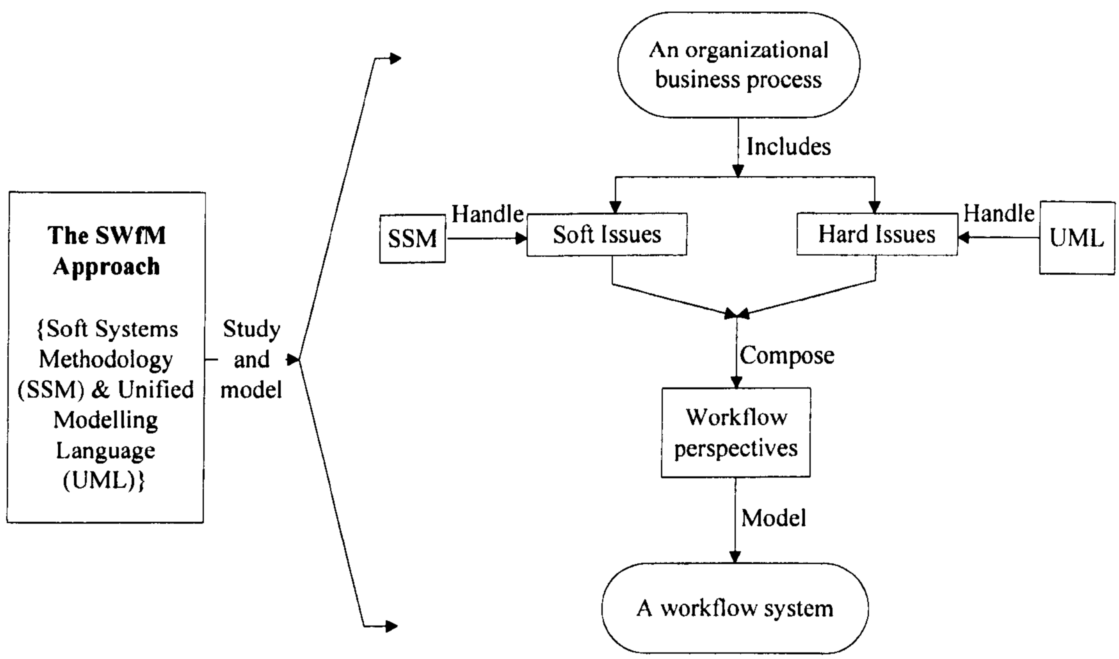



**Figure 4-1: The SWfM framework**

The above diagram shows that an organisational business process consists of two parts. The soft issues deal with human aspects of the process and some of them are represented as a soft perspective that can be handled by using SSM. On the other hand, the hard issues include the technical or tangible elements of the process, such as activities, tasks and data, which can be handled by the use of UML. These two parts can be classified into several workflow perspectives which will be addressed and modelled using the SWfM approach.

The following sections introduce SSM and UML in some detail.

### 4.1.1 Soft Systems Methodology (SSM)

SSM was developed by Checkland at Lancaster University as an output of the 'action research' programme of the Systems Engineering Department. SSM's structured thinking is based on system ideas. SSM 'concentrates on understanding problem situations, rather than developing solutions' [43]. It is used to improve the expression of ill-structured problems that have many perceptions. SSM is considered 'as an organised use of systems ideas in a methodology for learning one's way to purposeful action to improve a problem situation' [42]. It has been used in several problem areas and domains, ranging from health care information systems to enhancing crop rotation [125]. One of the recognisable areas is in information system development. There have been many attempts to use SSM in designing information systems, first by Checkland and Griffin [126], then Wilson [127, 128]. After that, Galliers [129] and Le Fever and Pattison [130] tried to implement SSM in different applications such as hospital information systems. Since then, several attempts have been made to integrate SSM with other techniques and methods to overcome the deficiencies of conventional methods, which tend to overlook the soft issues while concentrating on the hard aspects of information systems, such as data, processes and technology [131]. Stowell [132] tries to link the consensus primary task model with the data flow diagram (DFD), expanding the consensus model into a detailed data specification [133], while Prior [134] proposes a way to form a DFD from the consensus primary task model by first developing a conceptual or logical DFD which is similar to the consensus primary task model and then transforming this into a working or physical DFD for the system [133]. In addition, Sawyer [135] suggests two stages to convert from a conceptual model to a DFD. The first is to convert the primary task model to a desirable and feasible real world model by detailing and expanding the primary task model to a level where the transformation is clear. Then the real world model is turned into a data-based model, which may involve some changes in activities and data flows. The result is a model which represents the transforming activity and data flows that can be used to develop a logical DFD. Sawyer also suggests the use of the Structured System Analysis and Design Method

(SSADM) to design the system [133]. There is a difference between the approaches of Prior and Sawyer: the data model of Prior is conceptual, while Sawyer's is a real world model.

On the other hand, Miles [136, 137] distinguishes between two approaches to link SSM to structured system design (SSD) methods. First, SSM can be used as a front end to an SSD method through the link between the conceptual model and the DFD, and then the design method can be used to implement the system. In the other approach, SSM is used as a framework that controls the whole system design; hence, the SSD method is embedded within SSM. Savage and Mingers [138] investigate the possibility of linking SSM with Jackson System Development (JSD) methodology. Further, Avison and Wood-Harper [139, 140] develop the Multiview approach to combine SSM with other information system development tools. Also, COMPACT, which was developed by the government's Central Computer and Telecommunications Agency, links SSM with SSADM. Finally, Atkinson [141] introduced a new methodology, the Soft Information Systems and Technologies Methodology (SISTeM), which incorporates some ideas and techniques from SSM.

Therefore, SSM is considered as a general methodology whose implementation differs from one problem to another. SSM is 'described as a seven-stage sequence, and could, indeed be used in this way even though sophisticated users approached the use of the methodology more flexibly' [42]. There are two modes of use of SSM. Mode 1 uses it as a seven-stage methodology by implementing its stages in sequence. The stages are as follow:

xii. Problem situation is considered as problematic.

xiii. Problem situation is expressed by using the rich picture technique.

xiv. Root definitions of relevant purposeful activity systems are identified. A root definition specifies what a system is from a particular worldview. It is an intellectual act and not a description of the real world. A root definition has several elements that are used to validate if the root definition is well structured. These elements are presented by the mnemonic CATWOE (C: Customer, A: Actor, T: Transformation process, W: World-view, O: Owner of the problem, E: Environment constraints that are outside the control of the system's decision-taking process). In any situation, there may be many views or perceptions of a problem. The analyst should combine these perceptions into two types of root definition: the primary task root definition relates to the organisation's primary task, while issues-based root definitions relate to issues within the situation.

xv. Conceptual models of the systems named in the root definitions are constructed. A conceptual model is a human activity system model, which consists of logically connected activities that are carried out to satisfy the root definition.

xvi. Conceptual models are compared with the real world to generate recommendations concerning changes to improve the problem situation.

xvii. Proposals are introduced for changes which are systemically desirable and culturally feasible.

xviii. Actions are carried out to improve the problem situation.

Mode 2 is the use of the methodology in a thinking mode; SSM is used as a framework or set of guidelines. This mode has two strands of enquiry, logic-driven and cultural, which help to specify recommendations for change that will improve the problem situation.

The logical strand includes developing models of human activity systems and comparing them with the real world to identify recommendations for change. On the other hand, the cultural strand considers three elements of the problem situation: the intervention itself, the social part of the situation, and its political aspect. An interaction is made between these two strands to bring improvements in the problem situation.

SSM in this research is used to deal with the soft issues, some of which are represented as the soft perspective of a workflow system. In the following paragraphs, the SSM techniques are discussed to show how they are used to tackle these soft issues.

**i. Analyses One, Two and Three**

Analyses One, Two and Three are techniques of SSM used to investigate the intervention and the social and cultural aspects of the problem situation. Analysis one is 'an examination of the intervention itself' [142]. It is used to define the problem owners for further investigation of the problem situation and to specify relevant systems [142]. Analysis One supports the users' involvement in exploring the problem situation and constructing the relevant root definitions that are related to the situation.

Analyses Two and Three are concerned with the social and political analysis of the problem situation. Analysis Two concerns the various organisational roles, their responsibilities and the criteria by which their performances can be judged (roles/norms/values). Analysis Three handles the political part of the problem situation by examining the distribution of power and the right to carry out or stop any action. It is important to investigate how to acquire power in an organisation. This is done as the 'recent history of the organisation or group can be questioned and/or illustrated in these terms (what to have to possess to be powerful in a group or an organisation), all with the aim of finding out as deeply as possible how this particular culture "works", what change might be feasible and what difficulties would attend that change' [142].

Thus, Analyses Two and Three handle the soft issues that are related to defining the organisational structure and the employee's values.

**ii. Rich Picture**

One of the SSM techniques is the 'rich picture', which is used to present the problem situation in pictorial form. It is a useful technique as 'the start point of exploratory discussion with people in a problem situation' [142]. SSM supports pictures because, owing to the complexity of human affairs, pictures are a better medium than linear prose for expressing relationships [142].

The rich picture technique is used in the SWfM approach to deal with the following aspects:

- presenting the problem situation and the associated complexity, different perceptions and views of the people involved and the business issues.
- depicting the organisational structure, goals and policies of the situation and defining the conflicts among them, if any.

### iii. Root Definition

The second of the SSM elements is the root definition, which is used to provide 'a clear definition of the purposeful activity to be modelled' [142]. Thus, it is an important component from which the conceptual model is developed to present the required system. The root definition helps to identify a relevant system that incorporates a specific world view. The root definition part can deal with several aspects of the soft issues:

- showing the various views of the people who are involved in the problem situation.
- presenting the embodied business issues of the problem situation.

### iv. Model Building

The conceptual models or purposeful activity models are considered as intellectual devices in SSM. Their role is 'to help structure an exploration of the problem situation being addressed' [142]. These models are not representations of the real situation; they are purposeful activity systems which are built from declared world-views. They are used to initiate debate about the real situation and the desirable changes to it [142]. The conceptual model 'provides a common basis for discussion with the client and also establishes a framework for further analysis' [143]. The building of a conceptual model covers various aspects of the soft issues, as follows:

- It supports the users' involvement in building the conceptual model and encourages debate regarding the required change to improve the situation, which helps to promote job satisfaction and encourage the acceptance of the system and its use. Moreover, the users' involvement may have several advantages, such as improving the approach and modifying it according to their uses, increasing the understanding level of the approach and the acceptance of the resulting system, and encouraging participation in decision-making [144].
- In depicting those activities that may define the organisation's policies and goals, it may help to define the organisational structure of roles and responsibilities.
- It models the various views of the people that are incorporated in the root definitions which are relevant to the problem situation.

The problem of building models in SSM is that it requires experience of the real-world arrangements. Checkland states that 'the craft skill is to build a model using a background of real world knowledge' [142].

A number of techniques can be used to support the soft issues of the problem situation.

### v. Measures of Performance

The aim of using the system models is to cope with the complexity of the situation; that is, the complexity of interacting and overlapping relationships. System ideas deal with relationships, and so systems models seem a practical choice [142].

The modelled system should include a mechanism to help the system to adapt to changes in its environment. For that reason, the models of purposeful activity consist of a set of linked activities (an operational system) together with another set of activities which monitor the operational system and take control action to correct any deficiency. Therefore, it is essential to specify the criteria by which the performance of the system will be evaluated. The monitoring and control sub-system consists of 'a monitoring activity contingent upon the definition of the criteria by which system performance will be judged, and an activity rendered as "take control action" which is contingent upon the monitoring' [142].

The performance measures focus on three issues: 'checking that the output is produced; checking whether minimum resources are used to obtain it; and checking, at a higher level, that this transformation is worth doing because it makes a contribution to some high level or longer-term aim' [142]. This is a definition of the 3Es, which means efficacy (E1), efficiency (E2) and effectiveness (E3) [142].

These performance measures are used to ensure the quality of the system, to make sure that it achieves its required functionality and contains a mechanism that helps it to adapt to the changing environment.

Dobbin and Bustard [125] mention some features of the SSM that are not clearly handled in the hard paradigm. These features involve investigation of the problem situation, focusing on the system behaviour, focusing on changes to the problem situation, handling the different perspectives or viewpoints of the problem situation and resolving potential conflicts, emphasising the achievement of a desirable system, and focusing on the monitoring and control of the system.

### 4.1.2 The Unified Modelling Language (UML)

UML is a 'general-purpose visual modelling language that is used to specify, visualise, and document the artefacts of a software system' [145]. UML captures information about both the static structure and the dynamic behaviour of a system. The static structure specifies the different classes in a system and their relationships, while 'dynamic behaviour' describes the states of objects over time and the communications among them to accomplish the required functionality of the system. Several types of diagram are used to model the various aspect of the system, such as use case diagrams, class diagrams, interaction diagrams (sequence diagrams, collaboration diagrams, activity diagrams and state-chart diagrams) and implementation diagrams (deployment diagrams and component diagrams) [145].

The purpose of using UML is to model the hard aspects of the workflow system owing to the advantages and benefits that can be obtained from using the object-oriented paradigm with its standard modelling language. The object-oriented paradigm offers several features that make it the dominant paradigm today. These features include the following aspects:

- The object-oriented paradigm is considered a natural approach that simulates the real world through the use of the terms 'class' and 'object'. This improves communication and interaction among developers and users.
- The object-oriented approach manages and handles complexity through the use of abstract concepts such as *inheritance* and *polymorphism.*
- There are seamless transitions among the various system development stages or phases due to the use of the *class* and *object* concepts.
- In the object-oriented world, the developed system can be decomposed into sub-systems that can cooperate with each other to carry out the system functionality through their interfaces and the use of message-passing mechanisms. This enables the object-oriented system to be constructed in a modular way, in which each sub-system is implemented independently.
- The development of information systems from reusable classes and components makes them easier to build, quicker, cheaper and more robust. This

is done by developing a library of object classes that can be reused in constructing the various applications required by the organisation.

- The *inheritance* concept makes the developed information systems flexible and adaptable to changes, while reducing maintenance time.
- The object technology assists in developing complex applications such as graphical user interfaces, distributed systems and workflow systems which other approaches fail to address, through the use of information hiding or encapsulation and components which provide services to other parts of the system independently of the sequence of the control flow [43, 146-148].

Also, UML offers several other characteristics:

- UML is considered as the standard modelling language of the object-oriented paradigm that assists in ending the diversity and confusing war of different notations in the object-oriented approach.
- UML subsumes several notations, diagrams, rules and semantics that can be used to model the structure and logic of an application and provide multiple views of the system.
- Although UML is a complex modelling language, it is used as a communication medium among developers and users to discuss the structure of an application independently of any development process or programming language.
- UML can be extended to cover any application domain through the use of its extensible mechanisms.

The use of UML as a modelling technique has several advantages:

- assisting in understanding any application system and communicating these ideas to other developers and users;
- facilitating the development of alternatives;
- reducing the risk and cost of developing a system; and
- using the UML diagrams as a documentation of the developed system [43, 146-148].

The remainder of this section reviews the diagrams used to model the various perspectives of the workflow reference model as representations of the organisational business process. A detailed account of the use of these diagrams is given in Section 4.3.2 below.

- **The use case diagram** is used to model the various use cases of the organisational business process. It is recommended to classify the use cases depending on their range into *business use cases* that describe the interaction of the actor with the system and *system use cases* that specify the interaction with the software [149]. These features make the use case technique suitable for documenting the business process or part of it.
- **The UML activity diagram** is a useful tool that has several mechanisms and features to model the different perspectives of the workflow system. The activity diagram is used to model the functional, informational, behavioural and organisational perspectives.
- **The sequence diagram** is used to capture the dynamic aspect of each use case by modelling the collaboration among the various objects to carry out the functionality of the use case main scenario or the alternative ones of the business process that will be developed as a workflow system. The diagram is useful in presenting the processing part that an object performs, which is then converted into methods in that object's class. Further, it helps to define processing alternatives through the use of centralised or distributed control during the distribution of the processing among the objects.
- **The class diagram** is used in several ways; first, to depict the static structure of the use case by presenting its classes with their attributes and operations, interfaces, and their relationships. These classes collaborate to carry out the behaviour of the use case. Second, the class diagram can be used to present the analysis model for the whole system by combining and structuring the various classes from the different class diagrams of the use cases. This model will be used to design the workflow system by adding the design aspects to produce the design model. Finally, the diagram can be used to depict the structure of the organisation by depicting the various organisation roles (organisational perspective) that are involved in performing the use case tasks. After modelling each use case and identifying the required classes of the organisational perspective, these

classes for all the use cases are combined into one class diagram to represent an organisational structure model. This model will be used to design the organisational part of the workflow system.

It is possible to use the formal UML 2 package diagram [149] to model the complex and complicated class diagrams that present the analysis model of the system and the organisation structure. The classes are combined into packages in two ways, adopting either the common closure principle (where classes in a package need changing for similar reasons) or the common reuse principle (where classes in a package are all reused together). The package diagram shows the various packages and their dependencies [149].

The research is about modelling the internal organisational business process through considering its soft and hard aspects to develop a workflow system. Thus SSM and UML are selected to handle these aspects. SSM is used to handle the soft issues because SSM is a well-known methodology to deal with human aspects and a general one that can be used as a framework for the SWfM approach. In addition, UML as a standard modelling language of object-oriented paradigm involves several diagrams that can be used to represent the various perspectives of the workflow system. Also there are several UML tools that assist in developing any system.

## 4.2 The Application of SSM to Modelling a Workflow System

In this section, SSM and its techniques are investigated to validate its capabilities to address the various aspects of the workflow system. SSM consists of the following stages and techniques.

### 4.2.1 The Implementation of SSM Techniques in a Workflow System

In the first stage, a problem situation is investigated and its various aspects are identified. These may include defining the key people in the problem situation, such as owner(s), problem solver(s) and client(s) through the Analysis One technique. Analysis One assists in identifying the different issues that are related to the problem situation. Analysis Two may also be used to analyse the social part of the situation by identifying the key roles and their norms and values. Finally, Analysis Three is concerned with analysing the political part of the situation by identifying how power can be obtained and transferred or delegated in an organisation. Through these techniques, the problem situation is studied and examined in detail. In addition, at this stage SSM supports the tackling of some soft issues by encouraging the involvement of the owners and clients to express their views and perceptions of the problem situation and its objectives through interview or workshop techniques. Further, this participation of the involved people promotes the acceptance of the system, which is one of the outputs or recommendations of this methodology.

Regarding the workflow system, this stage involves a high-level definition of the organisation, its business process and other issues and aspects of the surrounding environment that interact with and affect the organisation. SSM also deals with a soft perspective that involves some soft issues of the workflow system which encourage its acceptance, such as user involvement, dealing with various viewpoints and perceptions regarding the problem situation and its objectives, and promoting employee job satisfaction.

All these issues and aspects of the problem situation are described by the rich picture technique at the second stage. A rich picture is built to represent the organisational business process and its surrounding environment.

Then, at stage three, a root definition of the system is elaborated to describe the issues and aspects of the problem situation from a particular worldview. It expresses the aims and objectives of the organisational business process and the need to support it through developing a workflow system. (See Section 4.1.1 (iii) above for further details.)

A conceptual model of the system named in the root definitions is then constructed (stage 4). A conceptual model is a human activity system model, which consists of logically connected activities that are carried out to satisfy the root definition. The developed conceptual model defines the various activities of the organisational business process. The conceptual model with the constituent activities is used as a basis to develop a workflow system. These activities depict the ideal organisational business process that achieves the root definition.

In stage five, the developed conceptual model is compared with the real world (the existing organisational business process, if it is available) to identify any gaps or omissions of activities and to generate recommendations concerning changes to address the problem situation and improve the organisational business process.

In stage six, any proposals for change are introduced and evaluated to establish whether they are systemically desirable and culturally feasible.

It is necessary here to discuss the changes that have to be made in the situation to suit the workflow system and to study its effect on the organisational business process through the technical capability and willingness of the users to accept and utilise the system to carry out their tasks.

Finally, stage seven deals with the actions that are required to be carried out to implement the agreed change(s) and to improve the problem situation.

The required changes in the modelled workflow system are carried out at this stage. The various perspectives of the workflow system are considered below to show how to cover them by using SSM and its techniques.

### 4.2.2 SSM Techniques and Workflow System Perspectives

The various aspects of the soft perspective can be handled by SSM and its techniques. The rich picture technique promotes the involvement of the users to provide information and discuss the various issues of the organisational business process; the root definition technique also requires the participation of the users to express their points of view regarding the organisational business process; and the development of a conceptual model needs the users' involvement for discussion and debate to reach a consensus on the model. These involvements of the users through the SSM stages support the acceptance and use of the resultant system to carry out the users' tasks. This promotes employee job satisfaction.

The functional perspective is defined partially through the conceptual model that shows only the activities of the organisational business process. These activities are required to be decomposed into tasks or steps to depict the business process in detail.

The informational perspective is not supported in SSM. This perspective presents the input and output data of each activity.

The behavioural perspective is depicted partially through the conceptual model that exhibits the sequential order of the activities. However, this is not enough to model the behavioural perspective. Therefore, it is important to include other basic control flow constructs such as conditional branching, parallel branching, join and loop.

The operational perspective is not supported by SSM because it deals with the implementation issues of the workflow system.

The organisational perspectives can be defined in SSM either by circumscribing the activities that the various roles in the organisation structure can perform, or by the Analysis Two technique, which describes the various roles and their norms and values, and can be used to document these roles.

SSM does not include any technique to describe the business rules and policies or their relationships, to define the causality perspective.

The quality perspective can be supported through the conceptual model, which involves monitoring and control activities that track the performance, collect information and instigate control activities to rectify any deficiency in the system.

## 4.3 The SWfM Approach

The SWfM approach is introduced to study the organisational business process in order to model it as a workflow system. The business process consists of two parts: the soft part that deals with the human issues and the hard part that involves the tangible elements such as activities, data, applications, resources etc. A defined soft perspective and several workflow perspectives are considered from the workflow reference model in Chapter Two to present these soft and hard issues of the business process as a workflow perspectives framework. The SWfM approach addresses the capabilities of SSM to model the organisational business process as a workflow system that covers the workflow perspectives framework. On the other hand, it is possible to integrate into SSM other methods and techniques, such as the UML, which is the de facto standard of the object-oriented paradigm, or an Event driven Process Chain (EPC) technique, to handle the perspectives of the workflow framework that SSM techniques alone cannot deal with.

The SWfM approach consists of two stages: the organisational business process investigation and the workflow system modelling.

### 4.3.1 The Organisational Business Process Investigation

This stage of the SWfM approach will be dominated by the use of SSM to study the organisation, its business processes and related issues, from a soft perspective. Thus, the organisation and its business processes are considered as a problem situation that SSM will be used to investigate. SSM is used to improve ill-structured problems that have many perceptions. It uses the concept of a 'human activity system', in which 'a particular interpretation of the word "system" has been adopted'. SSM is 'perceived as an organised use of systems ideas in a methodology for learning one's way to purposeful action to improve a problem situation' [42]. SSM is used when requirements are unclear, there are conflicting interests or the proposed system is contentious. Further, SSM is applied when there are changes in organisational structure or business processes [150]. In addition, SSM supports the broad view of the situation and strategic planning for business improvement [151]. The organisational business process investigation includes the following steps.

**Step1: Studying and investigating the organisation**

The organisation with its business process and surrounding environment are modelled using the rich picture technique of SSM. A rich picture is 'a cartoon-like representation that identifies all the stakeholders, their concerns, and some of the structure underlying the work context' [152]. It describes the organisation using symbols and arrows to define the relationships between them. This technique should capture the organisation, its elements and its context without becoming over-detailed. A rich picture should depict the organisation and its environment, such as the organisational business process, the relationship between the information system and the business process, the people and the various roles that interact with the organisation and other organisational issues. It should be a broad, high-grained view of the organisation. The construction of a rich picture requires deep understanding of the organisation and relevant human activity systems, leading to conceptual model building. This technique assists in understanding the problem situation and its environment and promotes user involvement in the gathering of information and in discussing the problem situation in order to get a clear picture of it [152, 153].

**Step 2: Identifying the organisation's mission or purpose**

The main mission or purpose of the organisation is identified and is used to develop the root definition. This step includes some discussion and debate with people who may have different views regarding the main mission of the organisation. SSM deals with these different views by accommodating them and removing any contradictions within them. The rich picture technique can also be used to reach a consensus organisation mission statement. The root definition (RD) that represents the organisation mission statement must be validated by using certain rules, which stipulate that the RD should be one sentence in which the major verb represents the transformation process which describes the organisation's purpose; its structure is tested by using the mnemonic CATWOE, in which the elements T (transformation process) and W (world view) must be identifiable in every RD, while the other elements are optional [153].

**Step 3: Constructing a conceptual model of the organisation**

A conceptual model or consensus primary task model (CPTM) is constructed to represent what such a system would have to do to be the one that is defined in the root definition. Wilson [153] defines CPTM as 'the consensus of the problem-solving group that this is what they are taking the organisation unit to be doing (now and/or in the future)'. The CPTM has become a major development in the analysis of organisation-based problems; it 'represents a unique statement of the set of the activities taken to represent the particular organisation unit of concern' [153]. The conceptual model is considered as a 'human activity model' that includes the activities required to accomplish the root definition, which are depicted without considering the organisation structure. Its importance is derived from the fact that it provides the starting point for a whole range of investigations. It can be used as a basis to study the organisation and to support the process of organisational restructuring or organisational design.

Wilson [153] describes four ways to develop a CPTM. The first is mission-statement based, where the mission statement of the organisation is used to format the RD and then model it by using a CPTM. Another way is to collect the different world views (Ws) of the people involved within the situation and translate these into RDs, which are modelled by conceptual models that are combined later to develop a

CPTM. The third way is a wider-system extraction, which consists of two steps: mission statements and job specification that are related to the situation are used to construct a wider system for the situation. Then, a feasible role can be identified by mapping various role boundaries onto the model developed in the previous step. The activities within the defined boundary are taken as the CPTM of the required role. The last way is an enterprise model assembly, in which CPTM is developed from the models that describe the four aspects of the enterprise: its core business or process, its support services, its relationship with environment and its managing process.

The developed conceptual model should be validated and tested by the rules of the formal system model (FSM). These include the requirement for a complete logical consistency between the activities in the model and the purpose as stated in the RD; there should be complete logical connections among the activities of the model because it is a system; and there should be monitor and control activities in the model to monitor the performance, collect information and take control activity to rectify any deficiency in the system.

The conceptual model consists of the various activities and the logical dependencies among them. The conceptual model notation consists of the symbols shown in Figure 4-2 [taken from [153].

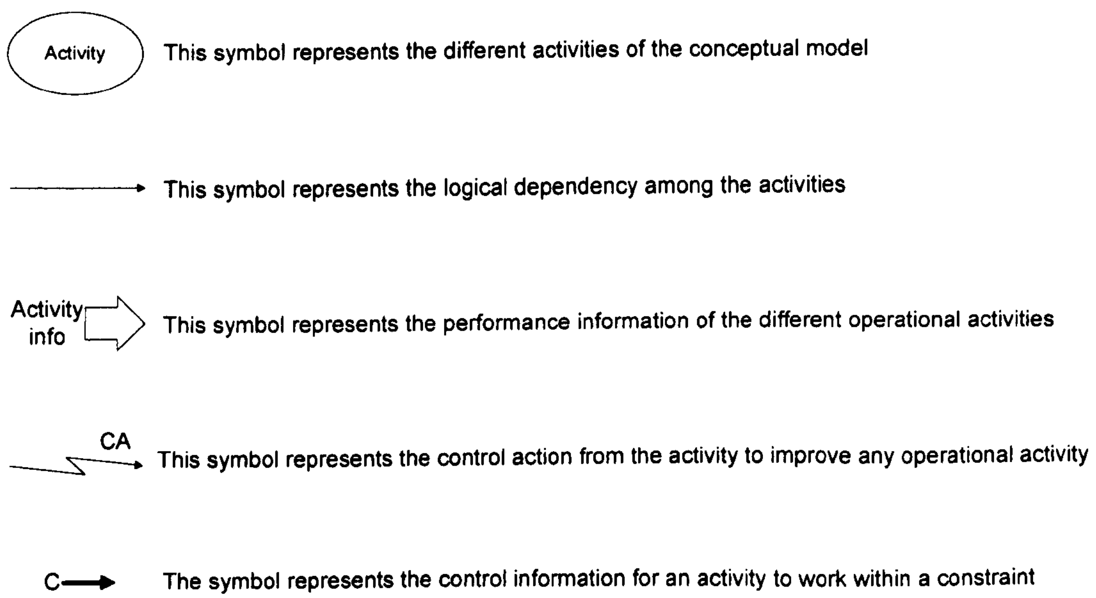



**Figure 4-2: The conceptual model notations**

**Step 4: Comparing the conceptual model with the real world**

The developed conceptual model is compared with the current organisational business process (if it is available) to validate that the resulting conceptual model fits with the organisation, to identify any missing activities and to propose any change to improve the organisational business process.

This stage handles the various aspects of the soft perspective.

### 4.3.2 The Workflow System Modelling

The second stage of the SWfM approach deals with modelling the organisational business process as a workflow system. SSM and its various techniques are used to represent the other perspectives of the workflow perspectives framework. The conceptual model of the previous stage is used as a basis for a workflow model. It includes all the activities that have to be carried out to achieve the required mission of the organisation and its business process. The activities of the conceptual model are connected with each other by arrows that represent the logical relationships among them. Therefore, the conceptual model with its activities can be considered as a representation of the functional perspective of the workflow model in overview form, which requires decomposing them further into several tasks or steps. In addition, the sequential connection of the activities within the conceptual model depicts the workflow behaviour perspective, which is considered as one construct that needs the addition of other behavioural or control constructs to optimise the workflow modelling. The following steps show how to depict the organisational business process as a workflow system using the SSM conceptual model.

**Step 1: Constructing a Subsystem description table**

The conceptual model of the organisational business process is considered a complex model that should be decomposed into several subsystems to facilitate the management of each subsystem. A table called a 'subsystem description' is developed to describe the various subsystems of the organisational business process by identifying each subsystem according to its number, name and constituent activities, as shown below in Table 4-1.

| | |
|---|---|
| Subsystem Number: | Subsystem Name: |
| Subsystem Head: | |
| Subsystem Activities:<br>1.<br>2.<br>3.<br>4.<br>5. | |

**Table 4-1: A subsystem description**

### Step 2: Constructing an activity description table

Another table, called an 'activity description', is developed to define in detail each activity within a subsystem. The activity information is collected by examining each activity in detail to assist with workflow modelling. The layout is shown below in Table 4-2.

<table>
<tr><td colspan="2">Subsystem Number:</td><td colspan="2">Subsystem Name:</td></tr>
<tr><td colspan="2">Activity Number:</td><td colspan="2">Activity Name:</td></tr>
<tr><td colspan="2">Preceding Activity:</td><td colspan="2">Following Activity:</td></tr>
<tr><td colspan="4">Precondition:</td></tr>
<tr><td>Input Data:<br>1.<br>2.<br>3.<br>4.<br>5.</td><td colspan="2">Activity Tasks:<br>1.<br>2.<br>3.<br>4.<br>5.</td><td>Output Data:<br>1.<br>2.<br>3.<br>4.<br>5.</td></tr>
<tr><td colspan="4">Business Rules and Constraints:</td></tr>
<tr><td colspan="4">Postcondition:</td></tr>
<tr><td colspan="2">Required Skills and Capabilities:</td><td colspan="2">Role Name:</td></tr>
<tr><td colspan="4">Performance Criteria:</td></tr>
</table>

**Table 4-2: An Activity Description**

### Step 3: Modelling the other perspectives of the workflow system

The conceptual model and the two tables above are used to define the other perspectives of the workflow system.

The functional perspective is defined completely through the conceptual model and the activity table, showing all the activities and tasks or steps of the organisational business process. Suitable techniques for depicting this perspective include the UML activity diagram, the data flow diagram (DFD) and the Event-driven Process Chain (EPC).

The informational perspective is specified by using the activity description table that presents the input and output data of each activity. To model this perspective, it is necessary to use data techniques such as the entity relationship diagram (ERD) or a UML class diagram.

The behavioural perspective is depicted partially through the conceptual model that exhibits the sequential order of the activities. However, this is not enough to model the whole behavioural perspective. Therefore, it is important to include other basic control flow constructs such as conditional branching, parallel branching, join and loop. A UML activity diagram or the EPC technique can be used to model these control constructs of the behavioural perspective.

The operational perspective is not supported by SSM, because it deals with the implementation issues of the workflow system. However, a UML deployment diagram can be used to model this perspective by presenting the required software and hardware for a workplace in the workflow system.

The organisational perspective can be defined in SSM by circumscribing the activities of the conceptual model that the various roles in the organisation structure can perform, and then the Analysis Two technique, which describes the various roles and their norms and values, can be used to document these roles. This organisation structure can be modelled using a UML class diagram or an ERD.

The activity description table describes the business rules and policies and their relationships to define the causality perspective. These rules can be presented in a UML class diagram and an activity diagram, or EPC.

The quality perspective can be supported through the conceptual model, which involves monitoring and control activities that track performance, collect information and undertake control activities to rectify any deficiency in the system.

From the above discussion, it is obvious that SSM needs to integrate or adapt some techniques to handle the other perspectives it cannot cover directly. There are two possibilities to integrate SSM with other methods or techniques to represent the various perspectives of the workflow system. The first choice is integrating UML with SSM; the other is using the EPC technique to model the other workflow perspectives.

Both UML and EPC can be used to represent the functional, informational, behavioural, operational and organisational perspectives. EPC [154] models these perspectives in one diagram but it requires other diagrams or techniques to show how to design and implement them. For example, the functional perspective is depicted using the hierarchy diagram or function tree and entity diagram or UML class diagram can be used to represent the structure of the data or informational and organisational perspectives. On the other hand UML involves several diagrams that are used to model these perspectives in a consistent way through using the class and object concepts.

Here, the discussion will be about integrating UML with SSM.

**Step 4: Unified Modelling Language (UML) and workflow perspectives**

**Step 4.1: Converting the activities of the conceptual model into use cases**

The conceptual model and use cases techniques can be used to identify and document the different activities of a system. The conceptual model is a human activity system which is constructed from a root definition that represents a specific point of view or the various points of view of the people who are involved in the situation, such as owners, users or managers. The conceptual model consists of several activities that a system must perform to be the system specified by the root definition. Also, use-cases are used to identify and represent the system behaviour from the users' view. Therefore, it is possible to map the activities of the conceptual model onto use cases. These use cases are represented in a use case diagram, which is considered as the basis for modelling a business process as a workflow system by using the various UML diagrams.

The activities of the SSM conceptual model will be converted into use cases. Before that, it is useful to compare the activity of the conceptual model with a use case of UML. The conceptual model represents the minimum necessary set of activities that a system must perform to be the system defined by a root definition [128]. Cambridge Advanced Learner's Dictionary [155] defines an activity as 'the work of a group or organization to achieve an aim'. There is no clear or specific definition of the activity of the conceptual model in SSM. The following definition of 'activity' is therefore adopted: *part of a human activity system, which could be a set of actions or procedures that are carried out to achieve an aim.* On the other hand, a use case is 'a sequence of actions that an actor performs within a system to achieve a particular goal' [156]. The two approaches are compared in Table 4-3 by using three features: composition (parts), beneficiary and purpose.

| **Feature** | **Conceptual model activity** | **UML use case** |
|---|---|---|
| Composition | A sequence of actions or procedures | A sequence of actions |
| Beneficiary | A Customer | An Actor |
| Purpose | To achieve an aim (goal) | To achieve a particular goal |

**Table 4-3: The CM activity and UML use case comparison**

From this comparison, it is possible to adapt and convert the conceptual model activities into use cases by examining each activity and identifying its goal.

Next, the activities are mapped into activities with goals that have to be achieved. This step may involve decomposing an activity into two or more activities that have different goals or combining some activities into one activity that accomplishes one goal. If that happens, it is important to update the sub-system description table by adding new activities or deleting or modifying existing ones. Also, the new activities can be described either through constructing an activity description table for every new activity or just defining their details in the use case template when they are mapped onto use cases. This depends on the time available for a project.

The new activities with the required goals that must be accomplished are mapped one-to-one onto UML use cases that achieve the same goal. Figure 4-3 shows the mapping of a conceptual model activity onto a UML use case.

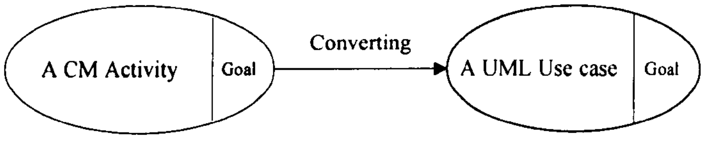



**Figure 4-3: Transition of a CM activity to a UML use case**

Next, the use cases of the business process are depicted in use case diagrams. The aim of the use case model, which is a collection of use case diagrams, is to present and document the organisational business process. This model is used as a basis to model the workflow perspectives by using the various UML diagrams. Finally, it is possible to design and implement the workflow system.

**Step 4.2: Analysing the use cases of the business process**

Each identified use case from the conceptual model is considered as a part of the business process that will be modelled as a workflow system. A use case contains information about the system behaviour and the organisation groups or roles that will perform this behaviour.

The use cases are defined in detail and documented in the approach dictionary. The use case specification template includes:

- Name

  Specifies the name of the use case.

- Brief description

  Describes briefly the use case, which is part of the business process.

- Performance goals

  Defines the required level of performance for this part.

- Benefit/value/purpose

  Defines the required goal of the use case.

- Workflow/ flow of events.

  Describes in detail or step-by-step the use case events.

- Special requirements

  Defines any non-functional requirements (performance criteria, data volume, security features and quality issues).

- Extension points

  Specifies where the use case can be connected to or extended by other use cases that perform or extend part of its functionality.

The use case documentation can be done in an iterative and incremental way, which means that detailed information can be added throughout the project. Some of these features are documented during the workflow modelling stage, some are added during the analysis stage, and others during the design stage. Also it is possible to use the prototype technique, which can be screen and window designs or a program with limited functionality, to test some aspects of the use case and obtain information from users who have reviewed the prototype. The prototype may show how the users perform their tasks through interaction with the workflow system, by modelling some of the system interfaces. The prototype enhances system usability through the involvement of the users to evaluate the prototype and make suggestions regarding some parts of the use case steps.

The use case diagram is then used to represent the various use cases of the business process and their relationships. The use case technique uses the notations shown in Figure 4-4 below.

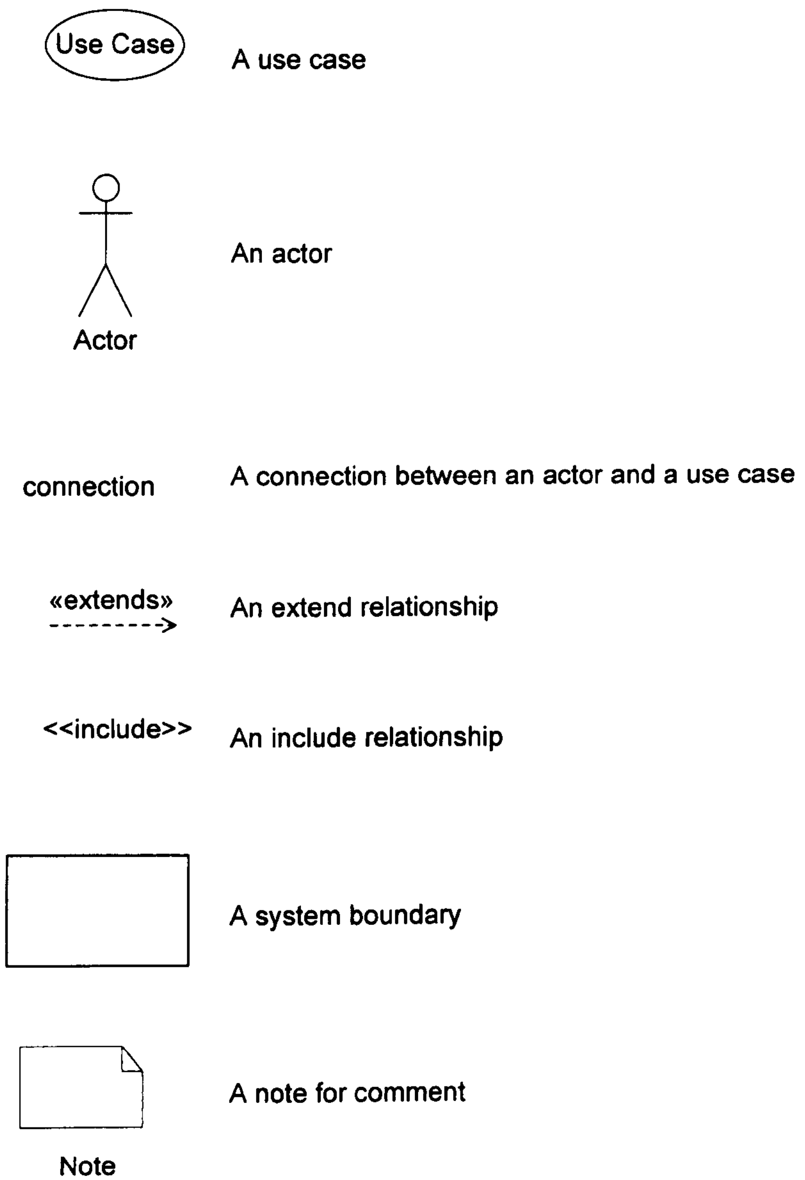



**Figure 4-4: The use case diagram notations**

### Step 4.3: Developing an activity diagram for each use case

A UML activity diagram is developed for each use case forming a part of the business process. The UML activity diagram models some of the workflow perspectives that exist in the use case through its features, which are suitable for modelling these perspectives. First, each activity state in the activity diagram can represent a task, step or an activity of a use case, which represents the functional perspective. Secondly, it is possible to model the informational perspective through the object-flow that may show the different objects that are involved in the activities or tasks, and the different parameters of the activities or operations. Also, it is possible to specify any change in the object's attributes by putting it in a section under the object's name. Thirdly, an activity diagram can model the behavioural perspective by using the various control flow constructs such as *sequence*, *branch*, *fork*, *join* and *loop* which can be used to model the different dependencies between activities in a workflow system. Fourthly, the business rules control the business and define constraints, conditions and policies concerning how the business and its processes are to be carried out. A rule can be defined as 'a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business' [157]. There are three types of business rules: derivation rules (defining how one piece of information can be derived from another), constraint rules (constraining either the structure or the behaviour of objects or processes) and existence rules (defining when something may exist and when it should come into existence; that is, when an object is created or destroyed). Rules can be specified formally in a particular language or informally in plain English [158]. These rules are modelled in the activity diagram, the class diagram and other UML diagrams; for example, an activity diagram has behavioural constraints in its activity flow and the relationships in the class diagram are declarative rules that specify how objects of those classes can be combined and related to each other. It is also possible to use a note to define the business rules attached to the model element to which it applies. Fifthly, the activities of the activity diagram can be grouped depending on criteria such as organisational unit or a role that will perform them, by using a swimlane feature. This represents the organisational perspective [145, 146]. Figure 4-5 shows the notations that are used to model the activity diagram.

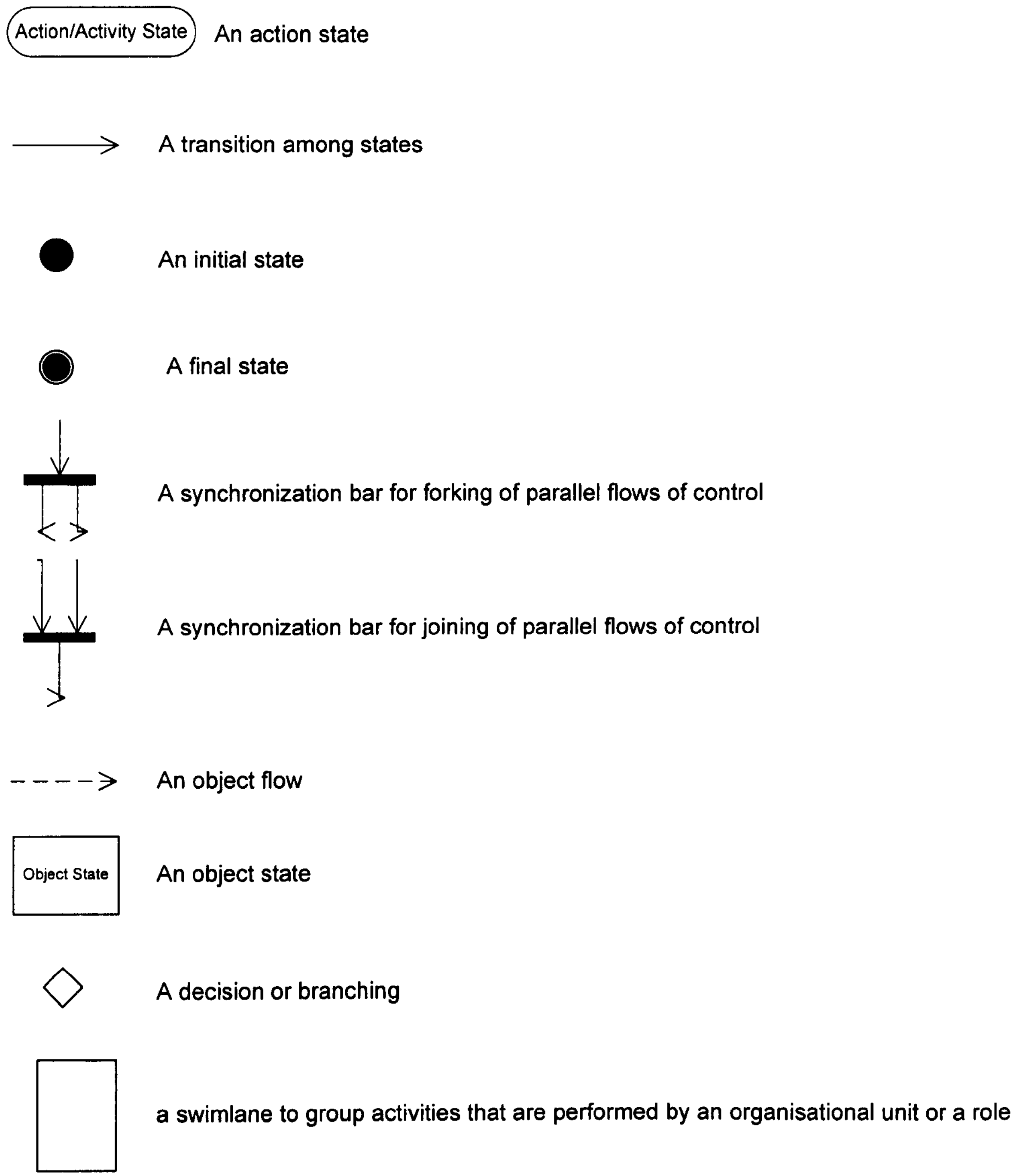



**Figure 4-5: Activity diagram notations**

## Step 4.4: Developing a sequence diagram for each use case

A sequence diagram is used to depict the interaction of objects to carry out the behaviour of each use case. The sequence diagram is an interaction diagram that shows the time ordering of the messages. It is used to model the dynamic aspect of a system. It shows the objects, their relationships and the messages between them. It has two features. Firstly, it shows the lifeline of an object. Secondly, it shows *focus of control*, which represents the period of time that an object is carrying out an action. It is possible to model the object's changing state in a sequence diagram by representing the different states under the lifeline of the object with a direct message that changes its state. In addition, it can show the interaction between roles instead of objects, without underlining the roles [146].

In object-oriented systems, the focus is on objects carrying out the processing necessary for the overall system to achieve its objectives. The sequence diagram can be used in two ways. First, it is used to model the interaction between the objects of the system that perform the different messages or operations to carry out a task or activity. It is also possible to establish a link with the activity diagram through the object flows of the activity diagram and the objects in the sequence diagram or through the names of the messages or operations of the activity diagram and the sequence diagram. Second, the sequence diagram is used to model the different roles of the organisation and the interactions among them. The purpose of developing a sequence diagram is to facilitate identifying objects, classes and their operations to build the class diagram of the workflow system.

The sequence diagram can be depicted by using the notations shown in Figure 4-6 below.

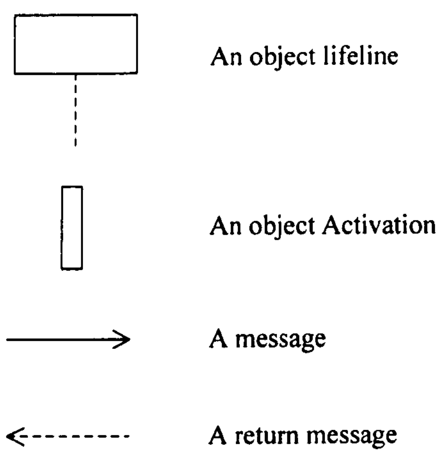



**Figure 4-6: The sequence diagram notations**

## Step 4.5: Developing a class diagram for the static structure of each use case

A class diagram is used to model the static structure of each use case. The class diagram models the static part of the system by showing classes, interfaces and their relationships [146]. It can be used to model the different classes of the use case that collaborate to carry out its behaviour.

The class description includes a specification of attributes, operations and relationships. *Attributes* specify the class characteristics and can be modelled in the second compartment of the class symbol, below the class name. The attribute has a value which corresponds with its type; for example, integer, real or string. On the other hand, *operations* define the class behaviour and they are actions that can be carried out by an object. The operations describe what services a class offers to other classes. They are modelled in the third compartment of the class symbol below the attribute compartment. Operations are implemented by methods. The third specification, *relationship*, defines the relations among the different classes in the diagram. The relationship is specified by using the term 'multiplicity' to define the number of objects that participate in the relationship. A multiplicity is considered as a business rule that constrains the way that activities can occur. Relationships among classes take different forms, such as association, generalisation, aggregation and composition. The relationships in the class diagram are declarative rules that specify how objects of those classes can be combined and related to each other.

The purpose of developing a class diagram is to present the various classes with their attributes, operations and relationships that together accomplish the required behaviour of the use case.

**Step 4.6: Developing a class diagram for the organisation structure of each use case**

A class diagram is constructed to model the static structure of the organisation, which represents the different aspects of the organisational perspective. It shows the various roles that interact with each other to carry out the use case behaviour. The organisation is divided into groups and each group consists of roles, which represent the different users or agents that carry out the different tasks or activities of the business process. An organisation is modelled as a class with stereotype <<organisation>>, where its attributes represent the required information about the organisation, such as its name, address and telephone number, and its operations represent the organisation's responsibilities. Also, each group is modelled as a class with stereotype <<group>>, where its attributes represent the required information about the group, such as its name, telephone number, manager and number of members, and its operations represent the group's responsibilities. Finally, each role is modelled as a class with stereotype <<role>>, where its attributes represent the required information about the role, such as its name and number of users, and its operations represent the responsibilities of the role. The class diagram uses the Symbols set out in Figure 4-7 below.

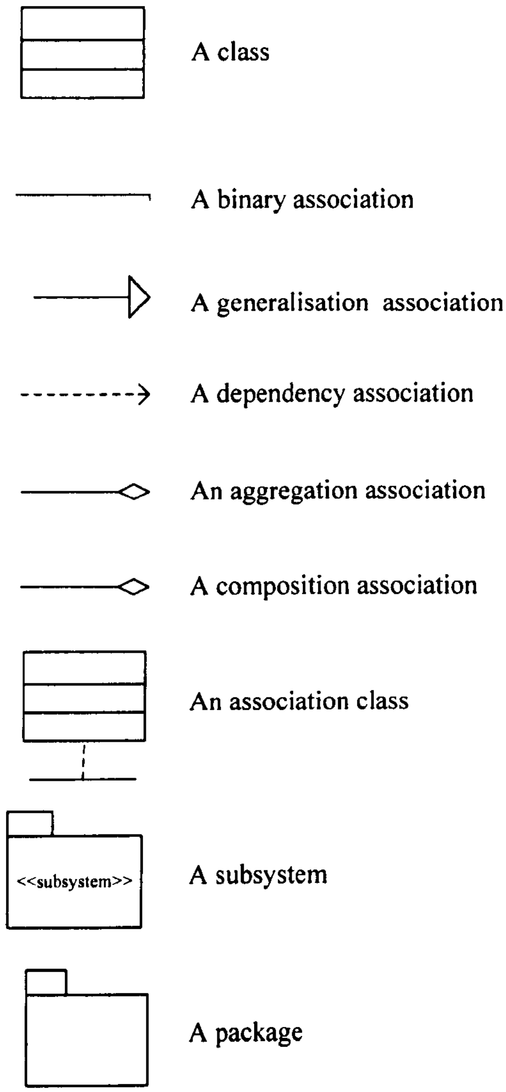



**Figure 4-7: The class diagram notations**

### Step 4.7: Constructing a class model for the business process

After modelling all use cases, the various class diagrams for the use cases are combined into a single class diagram, which is called a business process class model. This model includes all the classes of the workflow system, which collaborate with each other to carry out the system functionality.

The resultant class model will be scattered and complex. To overcome the model's complexity and structure it for reuse at the later stages of development, the classes will be arranged into different structures and hierarchies, using the *generalisation* and *aggregation* relationships. Generalisation specifies the logical relationship among classes that share some characteristics and behaviour. These classes are arranged into a generalisation structure or hierarchy. The most general class, called the *superclass*, is the root of the hierarchy, while the more specialised *subclass* may be seen as a leaf. A subclass inherits all the characteristics of its superclass and includes at least one detail not derived from its superclass. It is represented as a 'kind-of' or 'is-a' relationship. On the other hand, the aggregation relationship is where a class encapsulates a group of classes. It is represented as a 'whole/part' or 'has-a' relationship. The composition relationship is a type of aggregation relationship in which the composite object is responsible for the creation and destruction of its parts.

Further, it is possible to use the package mechanism to arrange the class diagram into different packages that represent the various subsystems of the required system. The use of packages aids understanding of the overall structure of the system. The package may include classes, interfaces, components, nodes, use cases, diagrams and even other packages. The relationship between the package and its elements is one of composition, which means that every element is owned by exactly one package. These elements can be presented textually or graphically. The access between the various packages is done through import and access mechanisms whereby, when a package imports another package, the elements of the imported package can access the elements in the importing package, but not the other way round. This is modelled as a dependency relationship. It is also possible to model the relationship between packages by using the generalisation relationship, which follows the same rules of generalisation among classes.

Decomposition can be based on several factors, such as similar or related functions, shared structure or organisation, or related locale [158]. The decomposition of the classes of the model into packages is done on the functional basis that each package represents a subsystem. The subsystem is depicted as a package with a stereotype <<subsystem>>. This subsystem includes the classes that collaborate to perform its functionality. This class model is used as the basis for the design and implementation of the workflow system.

### Step 4.8: Constructing a class model for the organisation structure

All the class diagrams that model the organisational perspective of each use case are combined into one class diagram, called the organisation structure model, which depicts the whole organisation. Also, the package mechanism is used to arrange the various classes of the organisation structure model into different packages. Each package depicts one of the organisation's groups.

### Step 4.9: Developing a deployment diagram for the operational perspective

The deployment diagram is used to model the physical aspects of a system by showing the configuration of run time processing nodes and the components that reside in them. It shows the static aspect of these physical nodes and their relationships and may specify the details of their construction. A node is 'a physical element that exists at run time and represents a computational resource that has at least memory and processing capability' [146]. Nodes are used to model the topology of the hardware on which the system will run. A node represents a processor or a device on which components may be deployed [146]. The operational perspective is considered as the implementation of the workflow system operations using different application programs, which can be modelled as components in the nodes of the system. These nodes are connected to each other to show the distribution of the business process operations among the different groups of the organisation. The notations used in the deployment diagram are illustrated in Figure 4-8 below.

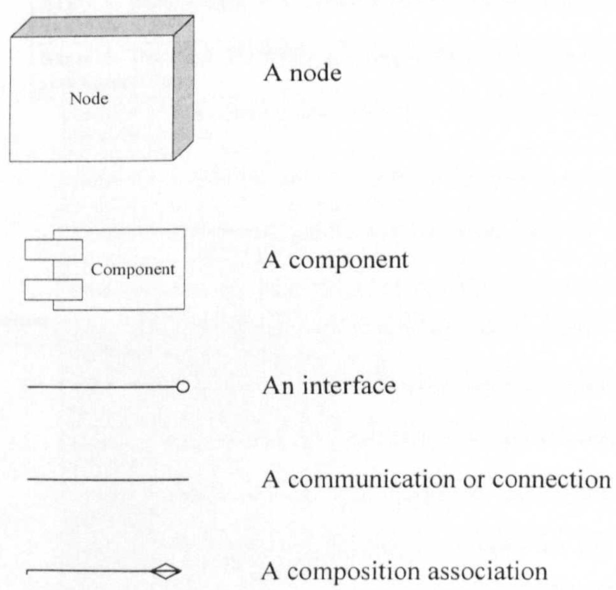



**Figure 4-8: The deployment diagram notations**

All these diagrams, models, technical terms and the other required information about the workflow system are stored in the approach dictionary.

## 4.4 Chapter Summary

The SWfM approach covers two stages of the workflow system development: the organisational business process investigation and workflow modelling stages. The first stage is performed to investigate and analyse the organisation or the problem situation within the organisation and the second stage is where the business process is modelled as a workflow system.

## The SWfM Approach

The approach consists of two stages, as represented in Figure 4-9 below:

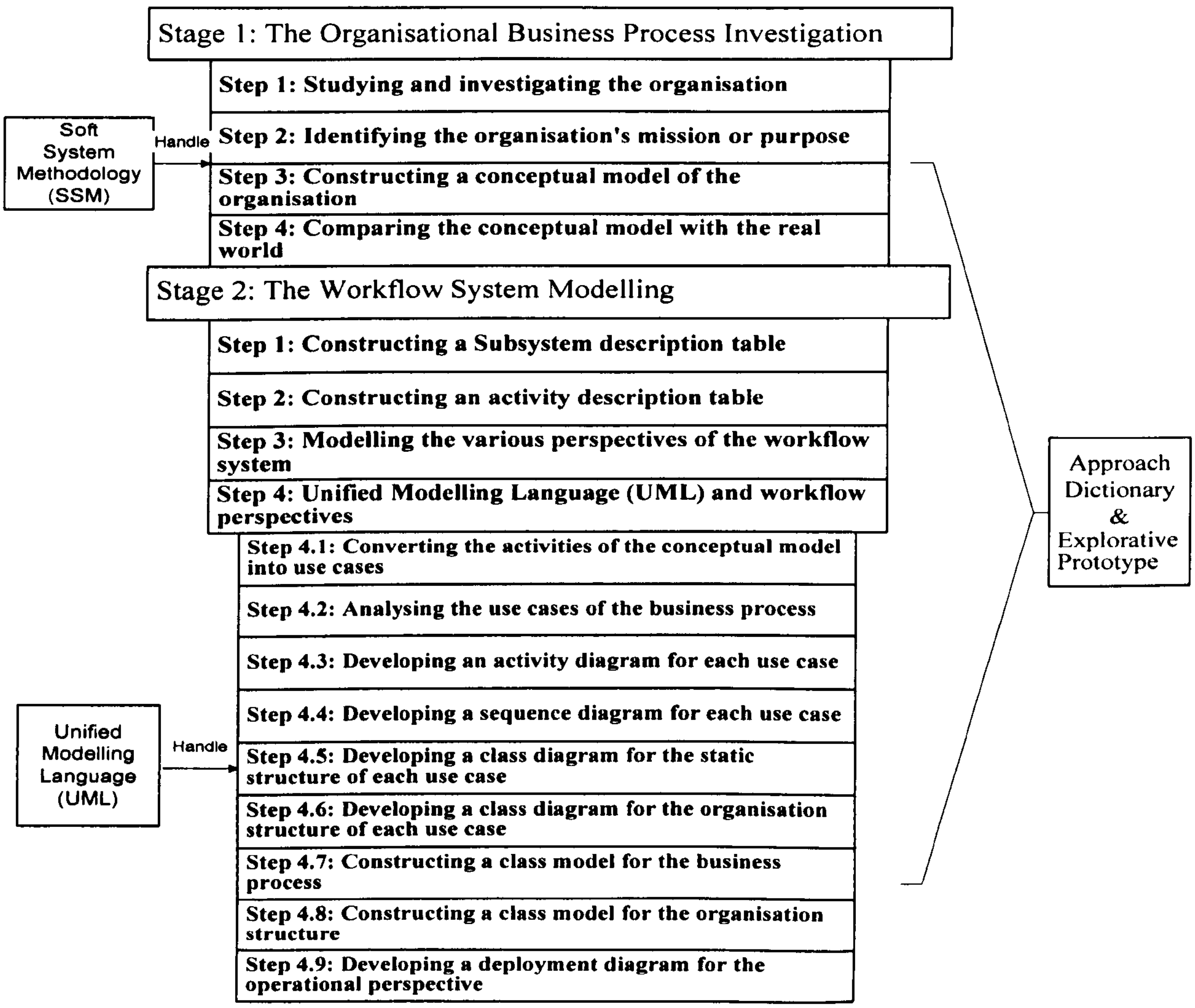



**Figure 4-9: The SWfM Approach: Stages & Steps**

There are also a data dictionary facility and an explorative prototype technique used with the SWfM approach:

**i. The Approach Dictionary**

This is used to store the definitions of the technical terms and the required information on the workflow system. This is important to minimize communication problems between developers and users.

**ii. Explorative Prototypes**

Explorative prototyping is used as a demonstration tool for validating and reviewing the use cases of the workflow system. It is used as a medium for communicating with users about the workflow system. Explorative Prototypes are sequences of dialog designs which present use cases.

The output of the SWfM approach is the business process class model, which consists of two models: the class diagram, showing all classes of the workflow system, and the organisation class diagram, depicting all classes that represent the various roles in the organisational perspective.

The main objective of this chapter was to develop an approach to modelling an organisational business process as a workflow system. This has been achieved by introducing the SWfM approach, which models the various perspectives of the workflow perspectives framework of Chapter Three.

The SWfM approach will be tested in the next chapter by implementing it on an illustrative case study, the management of a university course.

# Chapter 5: The Implementation of the SWfM Approach in a Case Study

In this chapter the SWfM approach, which was introduced in Chapter Four, will be implemented in an illustrative case study of course management in a university to demonstrate the applicability of the approach. First, the case study will be introduced and then the SWfM approach will be implemented in detail.

## 5.1 The Case Study

The case study is related to course management in a university. Due to the limited time available for research and the difficulty in obtaining information from users or practitioners in the real world, the course management case study is considered as illustrative only. The information on which the study is based was collected by interviewing my supervisor, who is a member of the Board of Studies, and reviewing the course development documents in the School and on the University's web site. My supervisor provided some ideas and documents concerning the course management.

The scenario of the case study as follows. The university offers a number of courses to students. There are many aspects and procedures involved in planning, developing, running and managing these courses, each of which has its aims and objectives that students are expected to meet during their studies. The course planners organise these aims and objectives in different modules. The module is a unit of education, covering a particular subject area, e.g. databases, system requirements and analysis, or programming. Each module has its own aims and objectives which are parts of the overall course aims and objectives. Each course also requires various resources, such as staff, buildings and other facilities which assist in achieving the course aims and objectives. In addition, it is important in the education process to measure the quality of the educational service provided and the students' satisfaction with it, through the implementation of a control subsystem that monitors and controls all aspects of the course. Thus, there are many aspects of both the course management and the module management that should be considered in order to achieve the required results from the course. This study attempts to model these by applying the SWfM approach to the course management case study.

## 5.2 The Implementation of the SWfM Approach

In this section, the SWfM approach, as described in Chapter 4, Section 4.3, will be implemented by applying it to the course management case study. The approach has two stages: organisational business process investigation and workflow system modelling.

### 5.2.1 The Organisational Business Process Investigation

This stage handles the soft perspective by implementing the following four steps.

**Step1: Studying and investigating the organisation**

The course management case study was investigated to identify a future vision of the course management. It is analysed and depicted below by using the rich picture technique.

The rich picture is a graphical representation of the course management situation that shows its different aspects. This was done by reading/consulting the course documents. Each course has different aims and objectives that can be used as the purpose of the course system to construct a root definition. The following figure (5-1) depicts the rich picture of the course management.

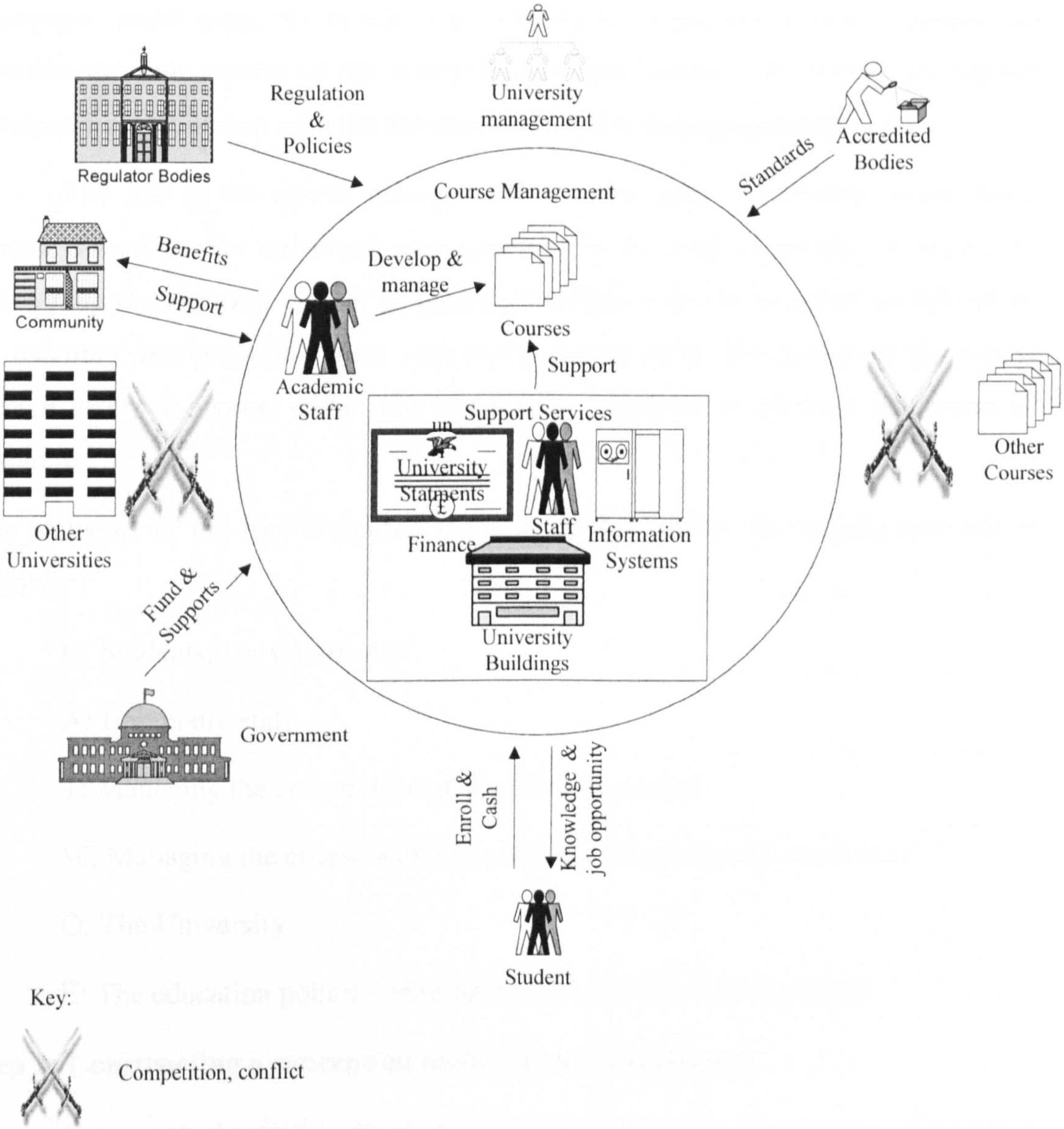



**Figure 5-1: The course management rich picture**

### Step 2: Identifying the organisation's mission or purpose

A root definition which was relevant to the problem situation was constructed and analysed using the CATWOE classification.

The root definition (RD) incorporates the purpose of the system and consists of different elements that are used to build a conceptual model of the situation. The RD of the case study was developed from the mission statement of course management, according to the first way to develop a consensus primary task model (CPTM), and by also considering the fourth way of developing a CPTM, which is an enterprise model assembly in which the CPTM is developed from the models that describe the four aspects of the enterprise: its core business or process, its support services, its relationship with the environment and its managing process [153].

The RD of the course management is: *developing a workflow model for a system owned by the university and operated by its staff to manage a course by developing the required course, graduating students with the required qualifications by teaching and providing them with the required skills and assessing the course though external bodies, within the constraints of education policies and financial resources.*

The elements of the root definition in terms of the CATWOE classification are as follows:

C: Students, University staff

A: University staff

T: Managing the course through a workflow system

W: Managing the course with the available resources and constraints

O: The University

E: The education policies, external bodies and financial resources

**Step 3: Constructing a conceptual model of the organisation**

A conceptual model is developed from the RD and is considered as the human activity system that consists of activities that achieve the aim or purpose of the RD. The following figure (5-2) shows the course management conceptual model.

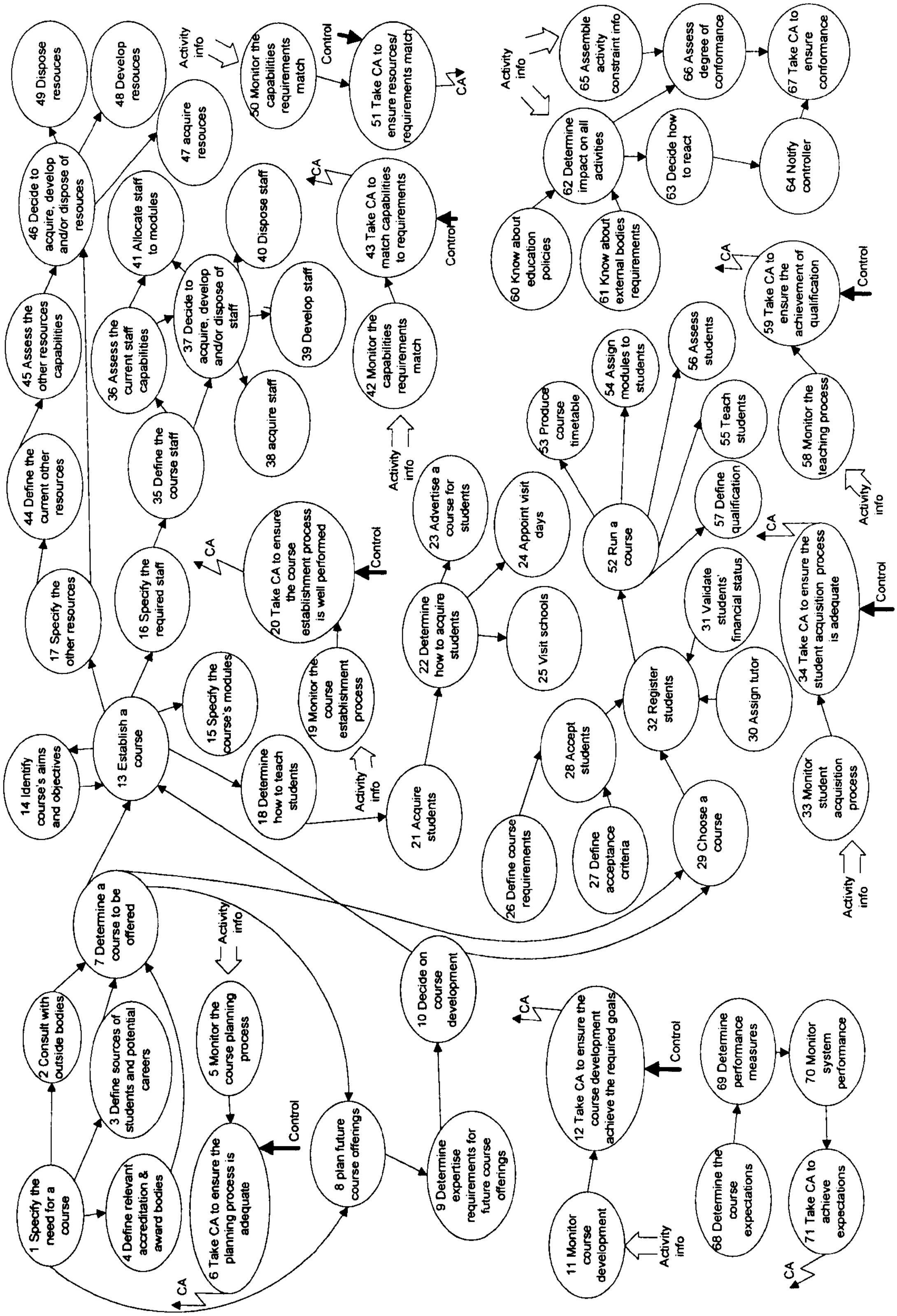



**Figure 5-2: Course management conceptual model**

The various activities of the conceptual model can be grouped into subsystems. The following figure (5-3) shows the high-level conceptual model of the course management system that contains the various subsystems.

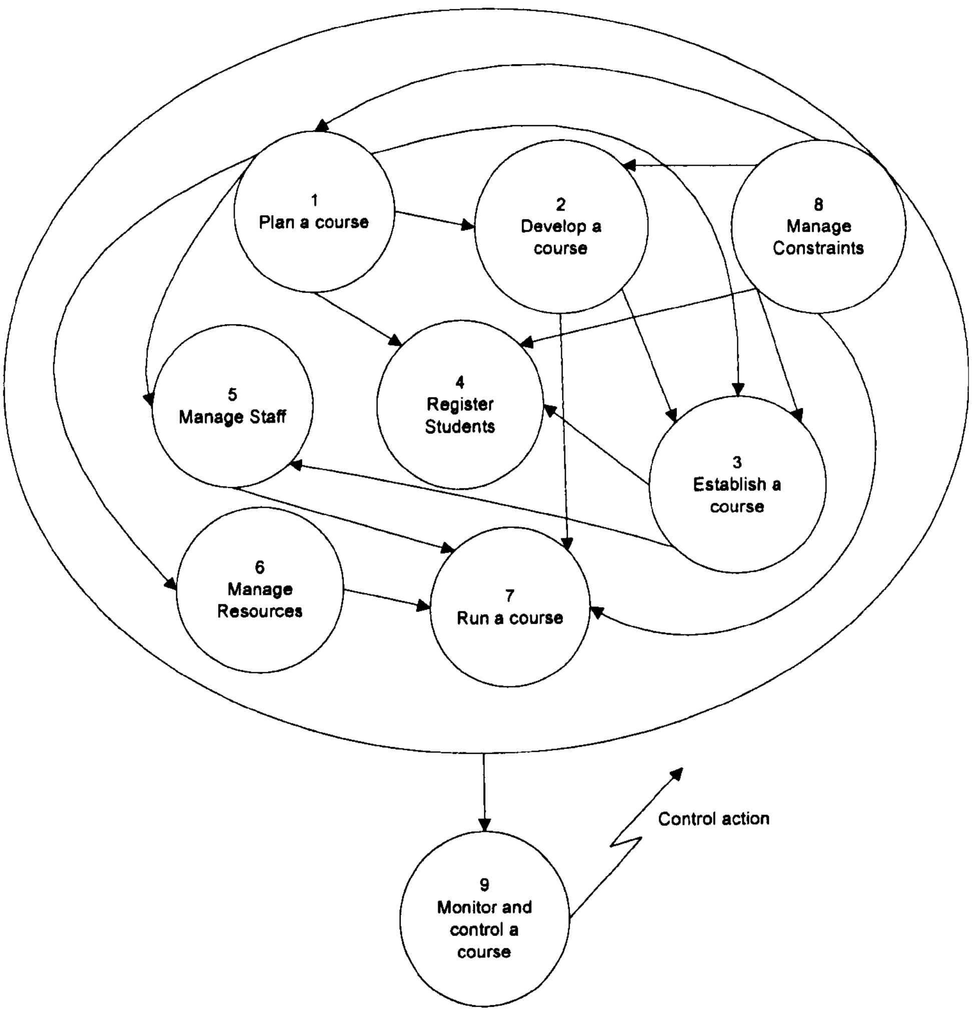



**Figure 5-3: The high-level conceptual model for course management**

These high-level subsystems are as follows:

1) Course Planning

2) Course Development

3) Course Establishment

4) Student Registration (can be split into Acquisition and Registration subsystems)

5) Staff Management

6) Resources Management

7) Course Running

8) Constraints Management

9) Overall Control

The following figure (5-4) shows the grouping of the various activities of the conceptual model into high-level subsystems after excluding the monitoring and control activities and grouping them into the control subsystem.

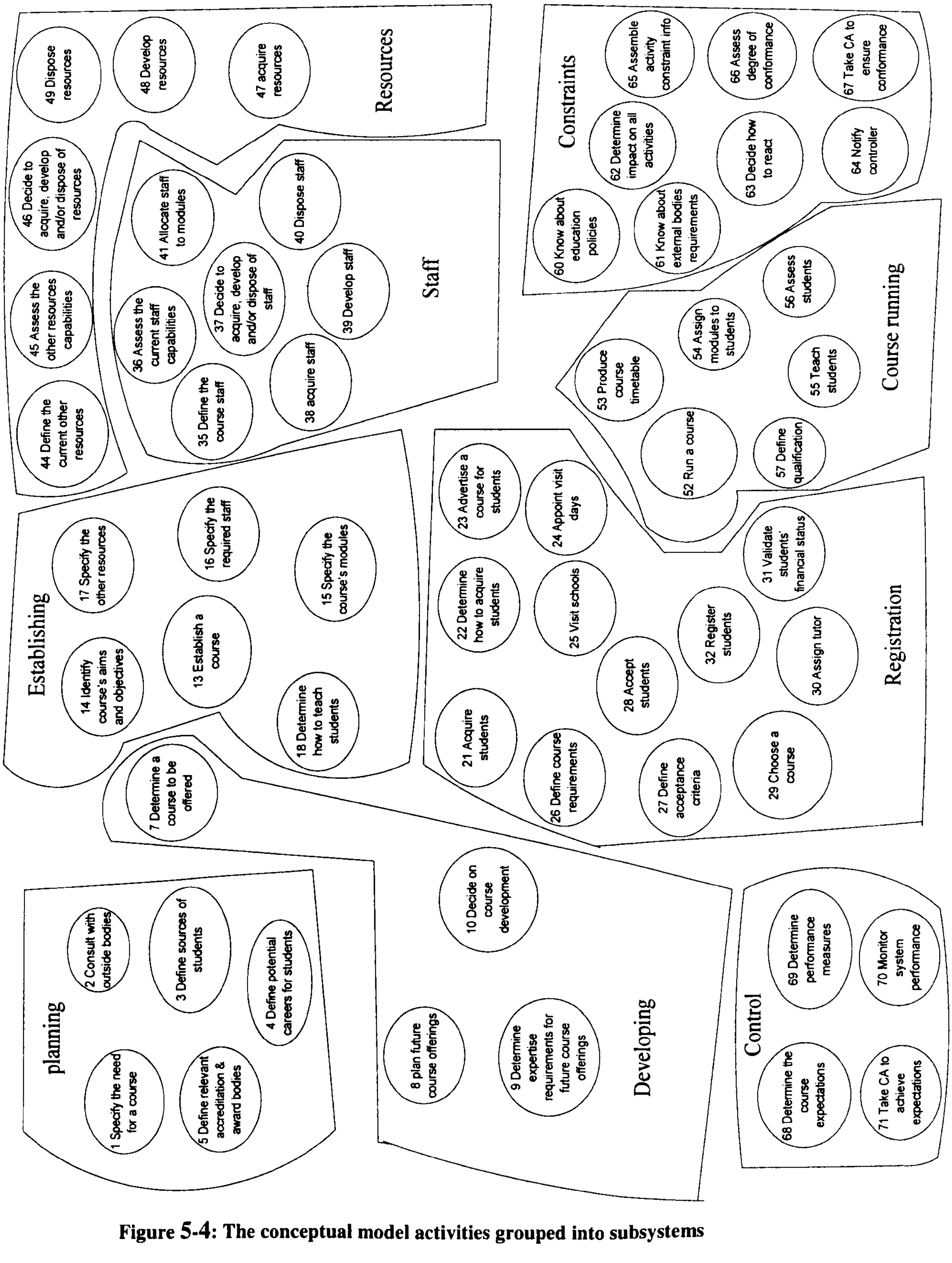



**Figure 5-4: The conceptual model activities grouped into subsystems**

The course management conceptual model can be tested or validated against the formal system model rules. All the activities of the conceptual model reflect and satisfy the RD. Also, the activities are connected to each by logical dependency arrows, which means that an activity at the head of the arrow depends on the activity at the other end of the arrow and will not be executed until the completion of that activity. For example, the activity 'identify the course aims and objectives' depends on the activity 'establish a course'. Finally, there is an 'overall control' subsystem which monitors the performance of each activity, collects information about it, assesses it with the setting measures for each activity and performs control actions to rectify any deficiency.

**Step 4: Comparing the conceptual model with the real world**

The developed conceptual model is compared with the current organisational business process (if it is available) to validate that the resulting conceptual model is fit for the organisation and to identify any missing activities or processes.

In this case study, there is no existing model for an organisational business process. Thus the conceptual model is considered as the basis to model the organisational business process as a workflow system at the following stage.

One of the features of Soft Systems Methodology (SSM) is the flexibility that allows it to be adapted to the situation under consideration. Because the case study is an illustrative one and most of the information was collected from my supervisor, who was also a user, and through the course documents and the university web site [159, 160], it was applied differently from the ideal implementation of the SWfM approach in the real world, in which the actual users participate in all steps of this stage, allowing the use of various SSM techniques to tackle the human aspects that are represented as a soft perspective.

### 5.2.2 The Workflow System Modelling

**Step 1: Constructing a Subsystem description table**

A subsystem description table was developed for each subsystem of Figure 5-3 to present its number, name and constituent activities. Table 5-1 shows the course planning subsystem.

| | |
|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning |
| Subsystem Head: (head of subsystem) | |
| Subsystem Activities:<br>1. Specify the need for a course<br>2. Consult with outside bodies<br>3. Define sources of students and potential careers<br>4. Define relevant accreditation and award bodies | |

**Table 5-1: The course planning subsystem**

### Step 2: Constructing an activity description table

An activity description table was developed for each activity of the subsystem to define in detail each activity within a subsystem. Table 5-2 shows the activity 'specify the need for a course'.

<table>
<tr><td colspan="2">Subsystem Number: 1</td><td colspan="2">Subsystem Name: Course planning</td></tr>
<tr><td colspan="2">Activity Number: 1</td><td colspan="2">Activity Name: Specify the need for a course</td></tr>
<tr><td colspan="2">Preceding Activity: None</td><td colspan="2">Following Activity: Consult with outside bodies</td></tr>
<tr><td colspan="4">Precondition: None</td></tr>
<tr><td>Input Data:<br>1. Initial Course Proposal (ICP)</td><td colspan="2">Activity Tasks:<br>1. A lecturer discusses with Senior Staff (SS).<br>2. The lecturer presents ICP to Board of studies (BoS).<br>3. BoS discusses paper (ICP)<br>4. BoS makes recommendations (RICP or no course) possibly with changes.</td><td>Output Data:<br>1. Refined Initial Course Proposal (RICP)</td></tr>
<tr><td colspan="4">Business Rules and Constraints: an initial course proposal should be ready for discussion.</td></tr>
<tr><td colspan="4">Postcondition: A decision to accept the course proposal or not is taken by BoS.</td></tr>
<tr><td colspan="2">Required Skills and Capabilities:<br>Some experience in developing a course proposal</td><td colspan="2">Role Name: BoS</td></tr>
<tr><td colspan="4">Performance Criteria: - The required time to evaluate the course proposal.<br>- The course proposal standard.</td></tr>
</table>

**Table 5-2: The 'specify the need for a course' activity**

### Step 3: Modelling the other perspectives of the workflow system

In this section, the other workflow perspectives are reviewed to address the required technique or techniques to model each one. The conceptual model and the two tables above are used to define the other perspectives of the workflow system.

The functional perspective is defined completely through the conceptual model and the activity description table showing all the activities and tasks or steps of the organisational business process. A Unified Modelling Language (UML) activity diagram will be used to model this perspective.

The informational perspective is specified by using the activity description table that presents the input and output data of each activity. To model this perspective, a UML class diagram is used.

The behavioural perspective is depicted partially through the conceptual model that exhibits the sequential order of the activities. However, this is not enough to model the behavioural perspective. Therefore, a UML activity diagram is used to depict these control constructs of the behavioural perspective.

The operational perspective is not supported by SSM because it deals with the implementation issues of the workflow system. However, a UML deployment diagram can be used to model this perspective by presenting the required software and hardware for a workplace of the workflow system.

The organisational perspective can be defined in SSM by circumscribing the activities of the conceptual model that the various roles in the organisation structure can perform; then the Analysis Two technique, which describes the various roles and their norms and values, can be used to document these roles. This organisation structure can be modelled using a UML class diagram.

The activity description table describes the business rules and policies and their relationships to define the causality perspective. These rules can be presented in a UML class diagram and an activity diagram.

The quality perspective can be supported through the conceptual model, which involves monitoring and control activities that track the performance, collect information and perform control activities to rectify any deficiency in the system.

In the next step, UML is used to complement the role of SSM, addressing the perspectives that it cannot handle. The following table (5-3) summarises the techniques required to represent the other perspectives of the workflow system.

| Perspectives | Modelling techniques |
|---|---|
| 1) Functional perspective | The SSM Conceptual model, activity description table and UML activity diagram |

| | |
|---|---|
| 2) Informational perspective | The activity description table and UML class diagram |
| 3) Behavioural perspective | The SSM conceptual model and UML activity diagram |
| 4) Operational perspective | The UML deployment diagram |
| 5) Organisational perspective | The SSM conceptual model, SSM analysis Two and UML class diagram |
| 6) Causality perspective | The activity description table, UML class diagram and UML activity diagram |
| 7) Quality perspective | The monitoring activities in the SSM conceptual model |

**Table 5-3: The techniques required to represent the workflow perspectives**

**Step 4: UML and workflow perspectives**

**Step 4.1: Converting the activities of the conceptual model into use cases**

The conceptual model activities are mapped onto UML use cases by examining each activity and identifying its goal.

Next, these activities are mapped onto activities with goals that have to be achieved. This step may involve decomposing an activity into two or more activities that have different goals or combining some activities into one activity that accomplishes one goal. Figure 5-5 below shows the conceptual model of Figure 5-4 with the inclusion of the goals of the activities of the first three course management subsystems. Time constraints prevented the consideration of further subsystems.

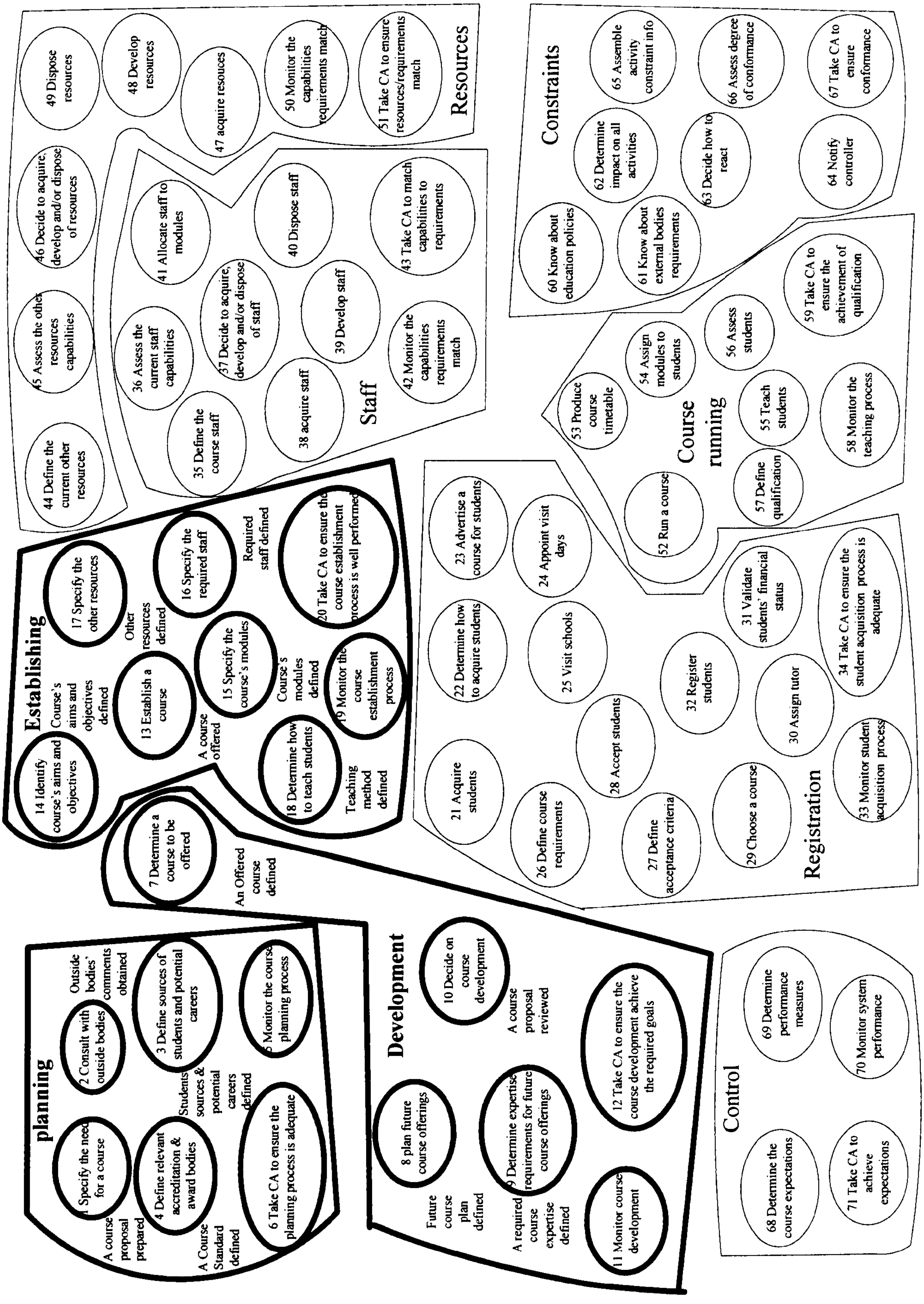



**Figure 5-5: The conceptual model activities with their goals**

After that, the new activities with the required goals that must be accomplished are examined to map them one-to-one onto a UML use case that achieves the same goal. Some activities can be decomposed into two or more activity-use-case maps. For example, the activity 'define student sources and potential careers' in the course planning subsystem could have two different goals. Thus, the activity is decomposed into two activity-use-case maps: 'define sources of students' and 'define potential careers for students'. Therefore, it is important to update the subsystem description table by adding these two activities. Also, these two activities can be described either by constructing an activity description table for every activity or just defining their details in the use case template when they are mapped onto use cases. This is shown in Appendix A, which describes the implementation of the SWfM approach to the planning subsystem of the course management.

Next, each activity-use-case map in the model can be mapped onto a use case. Further, the monitoring and controlling activities are combined in the control subsystem to be considered in one subsystem as a centralised control subsystem.

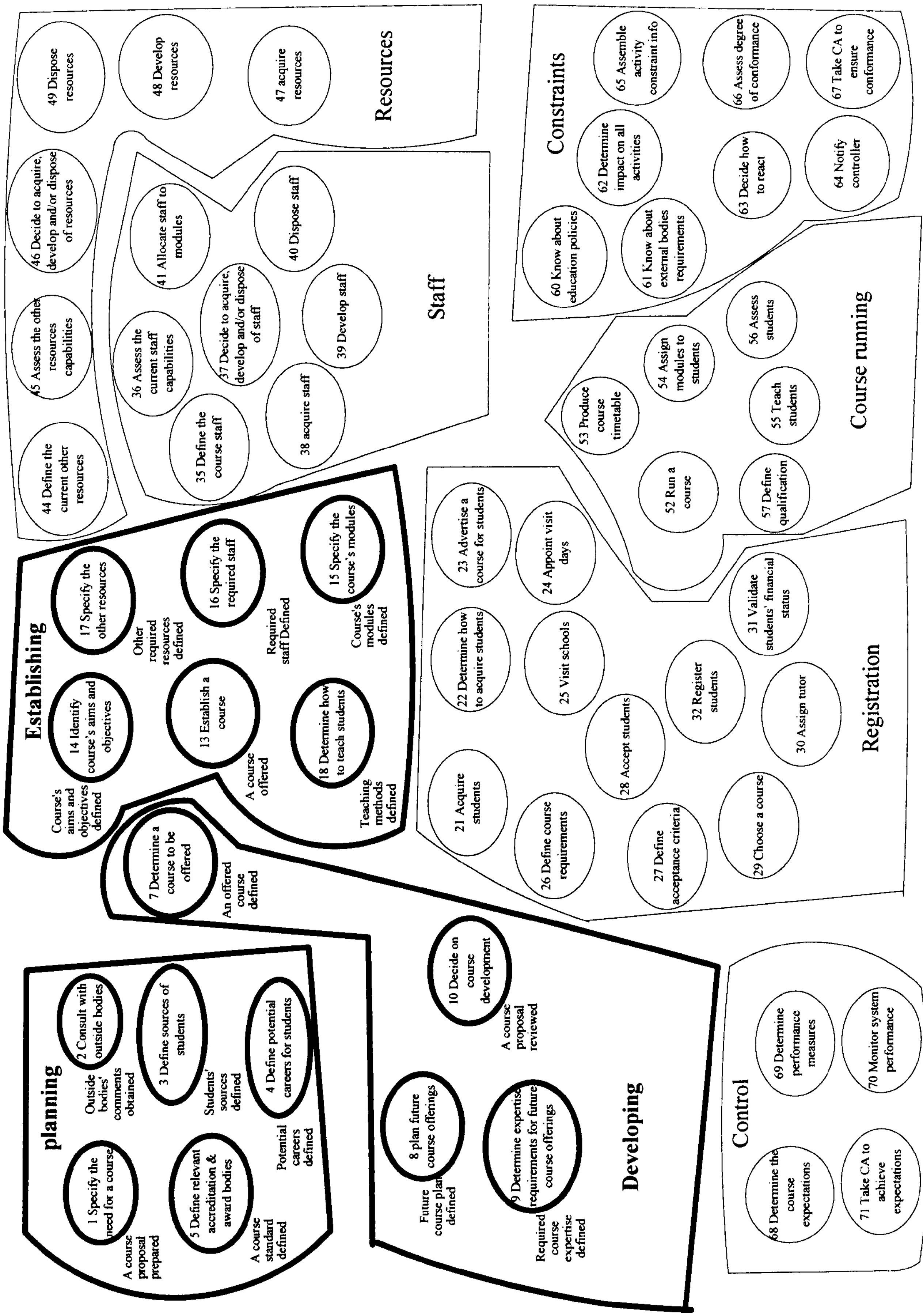



**Figure 5-6: Activities to use cases mapping**

In the next step, the use cases of the business process are depicted in the use case diagram consisting of the use cases with their actors. The aim of the use case model, which is the set of use case diagrams, is to present and document the organisational business process. This model is then used as a basis for modelling the workflow perspectives through the various UML diagrams. Finally, it is possible to design and implement the workflow system. Figure 5-7 shows the use case diagram for the course planning subsystem.

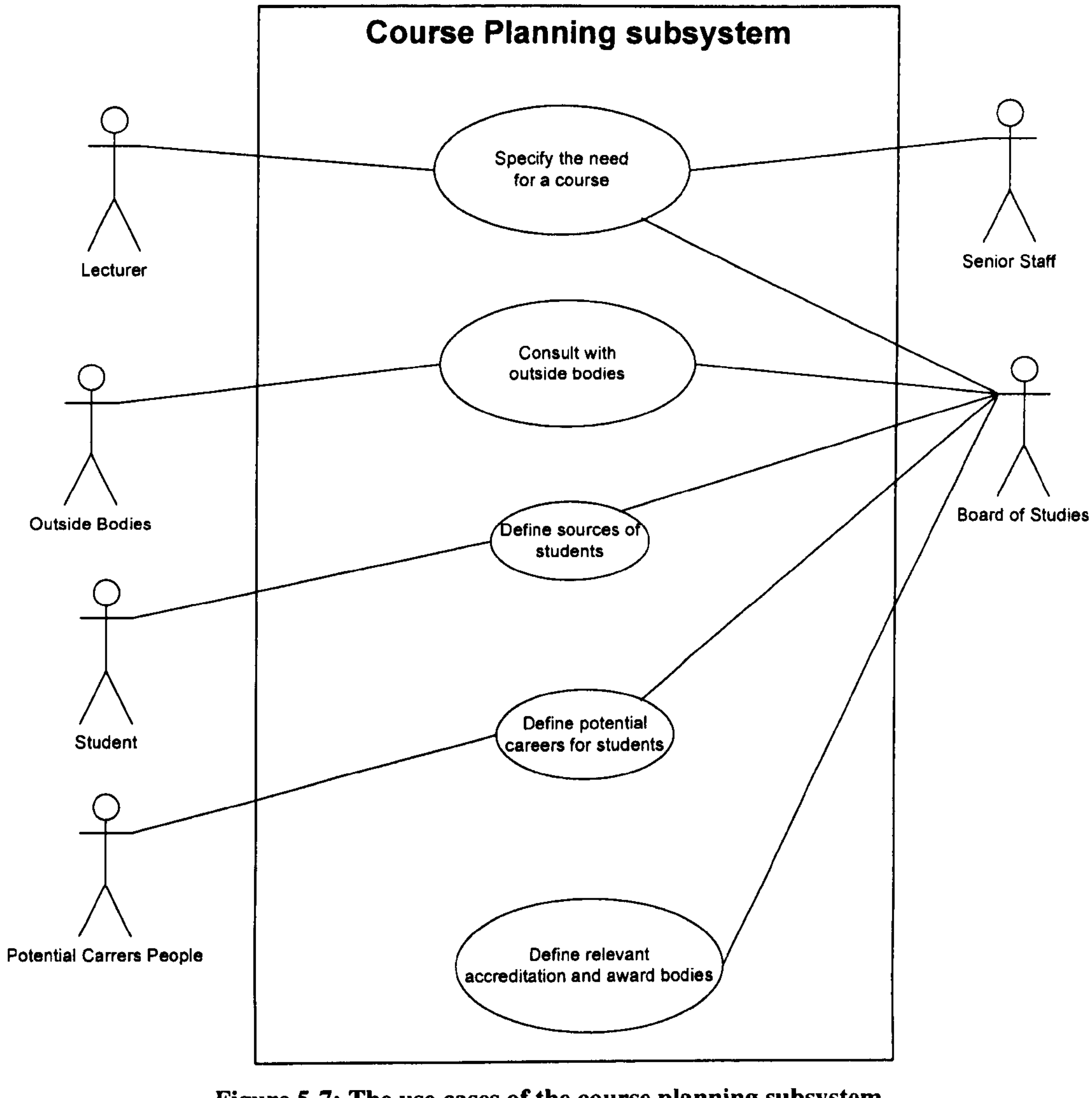



**Figure 5-7: The use cases of the course planning subsystem**

## Step 4.2: Analysing the use cases of the business process

These use cases are considered as the basis for modelling the workflow system by using UML and the object-oriented approach and later for designing and implementing it.

For each use case, it is essential to give a description in detail by using a textual format of the use case template defined in Chapter Four. The next use case template represents 'specify the need for a course'.

- Name: specify the need for a course
- Brief description

  This use case describes how to specify the need for a course through the following steps:

  a. A lecture discusses with Senior Staff (SS).

  b. A lecture presents ICP to Board of Studies (BoS).

  c. BoS discusses paper (ICP)

  d. BoS makes recommendations (RICP or no course) possibly with changes.

- Performance goals
- Benefit/value
- Workflow/ flow of events
- Special requirements
- Extension points
- Relationships

Some of these features are added in an iterative way by adding some detail in every iteration when the detail is available and through the progression of the project. It is possible to prototype some of the use cases by showing the interfaces that the users will use when they are carrying out their tasks.

**Step 4.3: Developing an activity diagram for each use case**

Uses cases are modelled by using activity diagrams, which represent and document the functionality of the use case, showing an overview of the use case by depicting its various components. They are used to model the functional, informational, behavioural and organisational views of the workflow system. However, the activity diagram needs to be improved to include some aspects of the business process such as the resources and some other control symbols. The activity diagram in Figure 5-8 below models the workflow of the 'specify a need for a course' use case. It shows the roles involved in this use case, which are lecturer, senior staff and Board of Studies, representing the organisational perspective of the workflow system and the activities. These represent the functional perspective that is carried out, for example *discuss the need for a course, prepare a course proposal, and submit a course proposal.* A swimlane is defined for each role to include the activities that are performed within its boundaries. The activity diagram starts with the initial state (black circle) followed by the activities (lozenges). The control flow is specified by using transitions (directed lines) that define the path from one activity to another and indicate sequential activities. When an activity is completed, the control flow passes to the next activity. Other control symbols such as decision (a diamond with a guard condition to define various paths), fork (a synchronisation bar with one input and several outputs to represent the splitting of a single control flow into two or more parallel flows of control) and join (a synchronisation bar with several inputs and one output to represent the joining of one or more control flows into one control flow) are also used to model the flow of control. The control flow of the use case represents the behavioural perspective. The informational perspective is modelled by using the object flow, which represents the object that is used within the activity as an input or output parameter and it is possible to depict its state if the activity modifies it. The object is connected to the activity through a dependency relationship (a dotted line) and its state is depicted in brackets below the object's name. The activity diagram concludes with the final state (a black circle within a white circle within a black border).

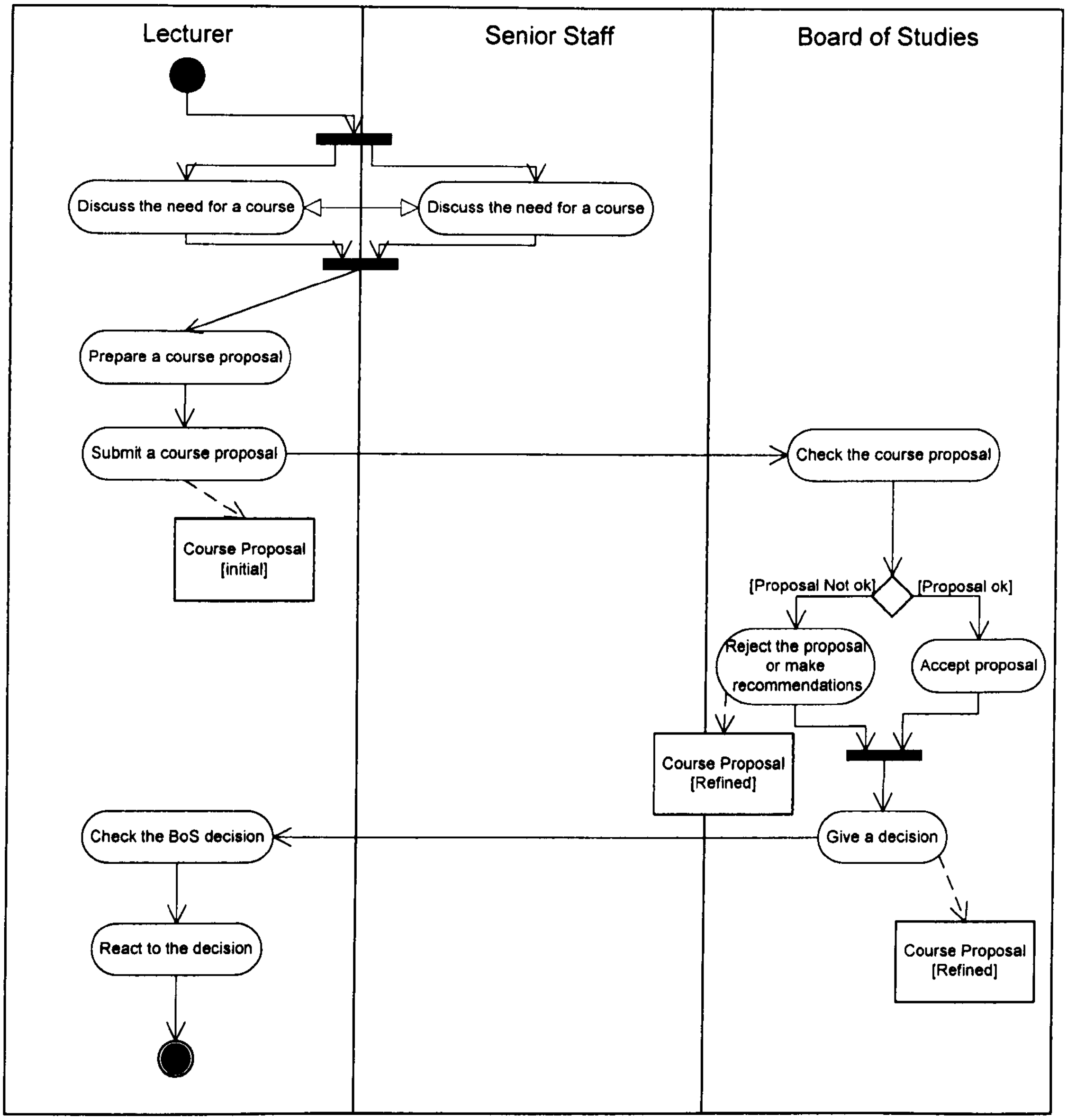



**Figure 5-8: Activity diagram for the 'specify the need for a course' use case**

### Step 4.4: Developing a sequence diagram for each use case

The interactions among the objects of the use case are modelled by using the sequence diagram to provide more detail about the workflow system through depictions of the various objects and the exchangeable messages between them. This assists in designing and implementing the workflow system. This diagram is used to model the dynamic aspects of the workflow system by showing the interaction between the system objects in carrying out the various tasks or activities. The interaction messages that are received by the objects are considered as operations on these objects. Figure 5-9 below shows the sequence diagram for the 'specify a need for a course' use case. It models the interactions between the various objects to carry out the use case behaviour through the various messages. The various objects (lecturer, senior staff and Board of Studies) are modelled at the top of the diagram with dotted lines to represent their lifelines. The diagram flows from top to bottom to depict the passing of time. The interaction between objects occurs by sending messages (a directed line). When a message is sent to an object, it calls an operation of that object to be executed. The period of time during which an operation executes is known as an *activation* (a rectangular block covers the object lifeline). It is possible to model the construction and destruction of objects on the sequence diagram. The created object is depicted directly by the message arrowhead, such as the creation of a course proposal object, while the destruction of an object is modelled by a large X on the object lifeline. An object can send a message to itself. This is called a reflexive message (a message arrow that starts and ends at the same object lifeline). The 'check the decision' message in Figure 5-9 represents such a reflexive message. A return message is optional and can be included in the diagram to show the return of control to the object that initiates the message which started the activation. Iteration is represented by an asterisk before the message name, and the condition that specifies the iteration limit can be added before the asterisk symbol and the message.

Branching can be added to the sequence diagram if the use case has more than one execution path. Each path is depicted by a condition guard inside square brackets and a message name that starts from the same object lifeline. Each path is followed only if the path condition is true.

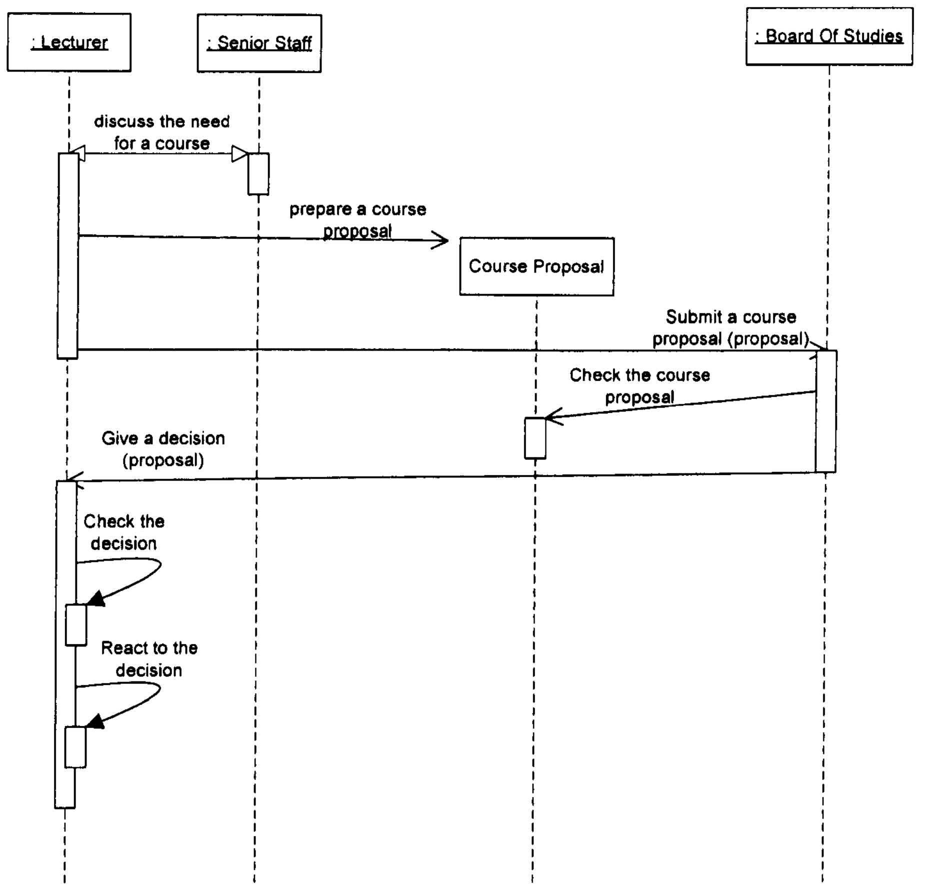



**Figure 5-9: Sequence diagram for the 'specify the need for a course' use case**

## Step 4.5: Developing a class diagram for the static structure of each use case

A class diagram is used to model the static structure of the use case by representing its classes and the relationships between them, which collaborate to carry out its behaviour. The class modelling includes specifying its attributes, operations and its relationships with the other classes. Figure 5-10 below shows the class diagram for the 'specify the need for a course' use case. This diagram includes four classes with their attributes and operations and the relationships among them.

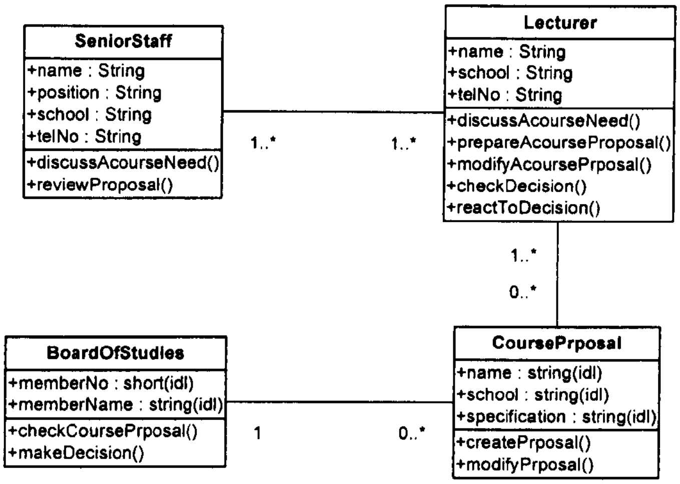



**Figure 5-10: Class diagram for the 'specify the need for a course' use case**

## Step 4.6: Developing a class diagram for the organisational structure of each use case

The different roles of the organisational perspective are modelled by using the class diagram. This diagram is used to model the different roles of the organisation that interact with each other to perform the use case behaviour by representing each role as a class with a stereotype <<role>>. The attributes, operations (responsibilities) and relationships with the other classes are added to each role. Figure 5-11 below depicts the class diagram of the organisational perspective of the 'specify the need for a course' use case. This diagram includes three classes that represent the three roles involved in carrying out the use case behaviour and their relationships.

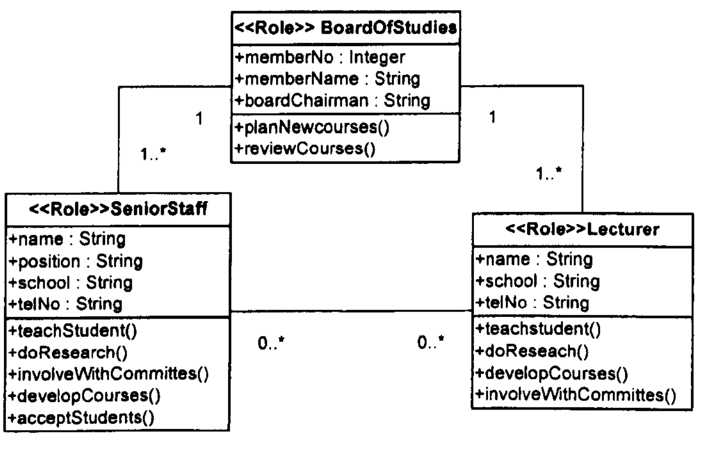
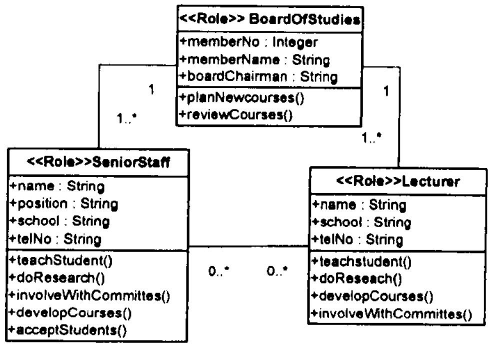
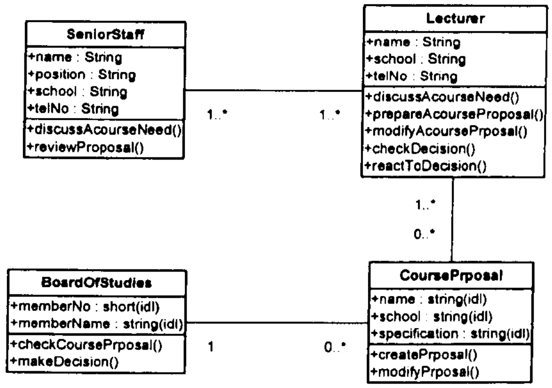



**Figure 5-11: Class diagram for the organisational view of the 'specify the need for a course' use case**

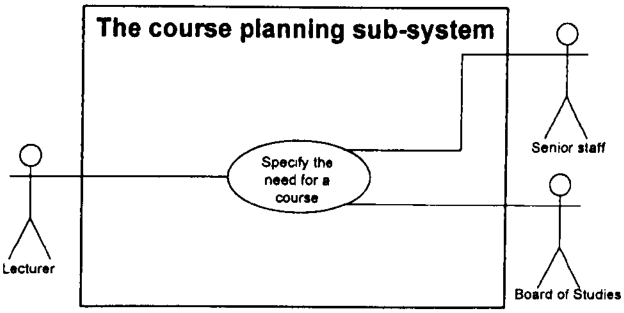
All the diagrams of the 'specify the need for a course' use case are combined into one model that shows the different views or perspectives of the workflow system. Figure 5-12 thus shows an overview of a use case. In the middle of the figure is the use case that is analysed and modelled. There is an activity diagram showing the different perspectives of the use case: the organisational perspective (roles), the functional perspective (activities), the behavioural perspective (flow of control) and the informational perspective (object flow). The sequence diagram depicts the dynamic aspects of the workflow system by presenting the interactions among the objects to perform the use case behaviour. Finally, the two class diagrams model the static structure of the organisational perspective and the workflow system.

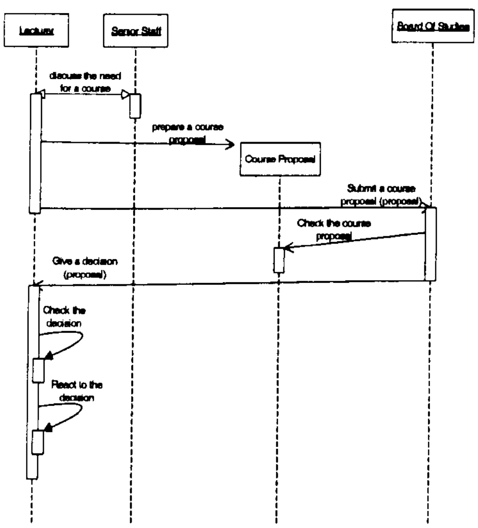



A class diagram for the organisation view of a 'specify the need for a course' use case

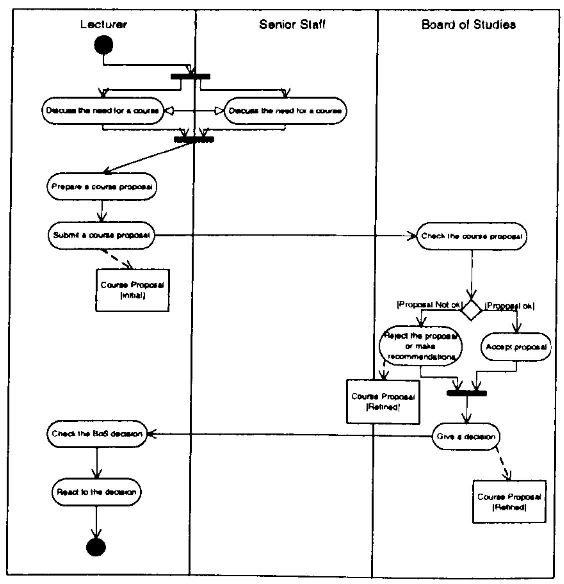



A class diagram for a 'specify the need for a course' use case




A sequence diagram for a 'specify the need for a course' use case




An activity diagram for a 'specify the need for a course' use case

**Figure 5-12: All diagrams for the 'specify the need for a course' use case**

**(Larger versions of the components are given in previous figures)**

## Step 4.7: Constructing a class model for the business process

An analysis model is constructed from the different class diagrams of the use cases. After modelling each use case and identifying the required classes that collaborate to carry out the behaviour of each use case, the classes of all use cases are combined into one class diagram, which is called an analysis model. This model is used to design the workflow system by adding the design aspects to produce the design model. Figure 5-13 below shows the classes of the 'course planning' subsystem. The classes are arranged by using the generalisation, aggregation and composition relationships. These relationships structure the classes of the model to overcome the model complexity and make them easy to reuse in the later stages of development. There is a generalisation relationship between the *staff* class as a superclass and its subclasses *lecturer* and *senior staff*. There are also two composition relationships, between the staff and the Board of Studies, and the Board of Studies and the student representative.

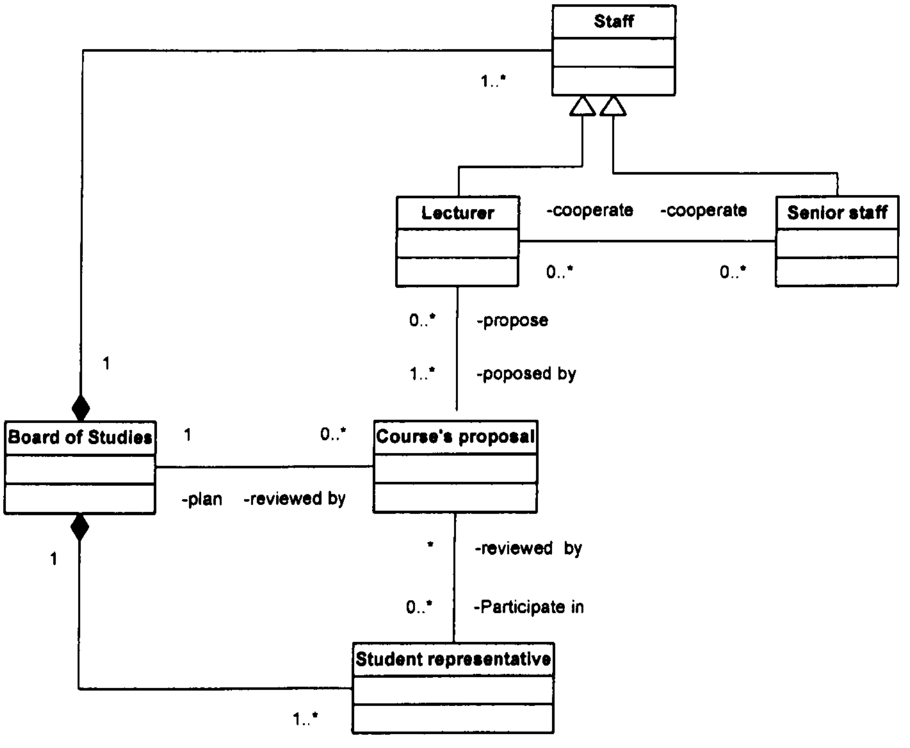



**Figure 5-13: A class diagram for a course planning sub-system**

In addition, it is possible to use the package mechanism to arrange and structure the class diagram of the course management analysis model. A package represents a subsystem, so each subsystem of the system is depicted by a package with stereotype <<subsystem>>. Figure 5-14 depicts the course management system with its subsystems that are modelled as packages. The relationship between the course management system and its subsystems is one of aggregation.

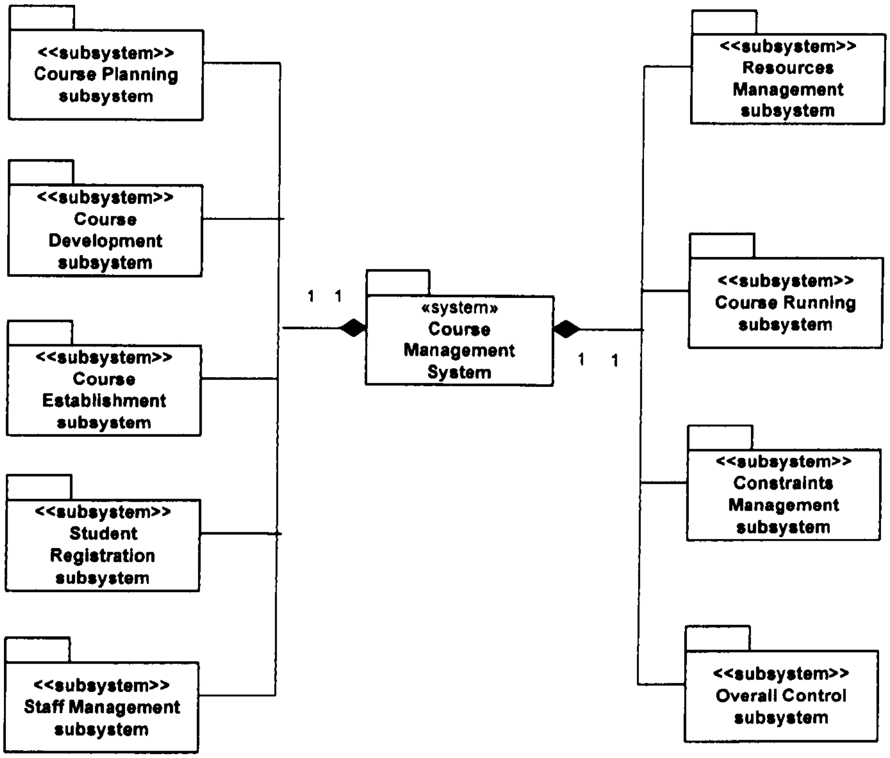



**Figure 5-14: Package diagram for the course management system and its subsystems**

### Step 4.8: Constructing a class model for the organisation structure

A class diagram for the organisational view is developed from the various class diagrams that represent the organisation view of each use case. The diagram depicts the class diagram for the organisation view of the 'course planning sub-system'. Figure 5-15 shows the overall structure of the organisational perspective of the developed system. The different groups of the organisation are modelled as a class with stereotype <<group>>, and the roles within these groups are modelled as a class with stereotype <<role>>. The relationships among these groups and between groups and their roles are modelled by using the different class diagram relationships, such as generalisation and aggregation.

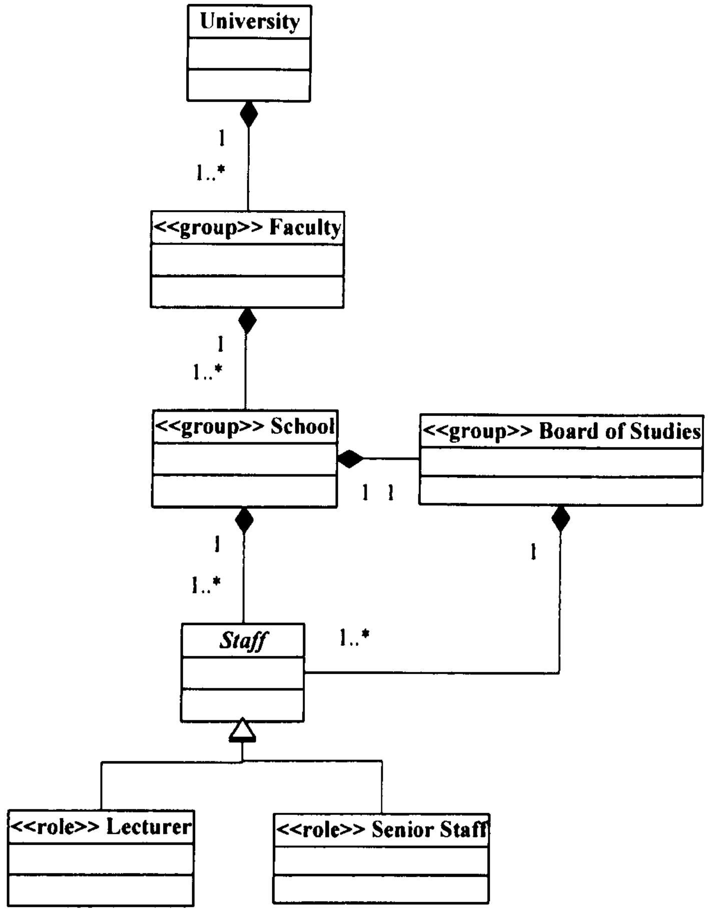



**Figure 5-15: A class diagram for the organisational view of the 'course planning' subsystem**

### Step 4.9: Developing a deployment diagram for the operational perspective

The operational perspective is considered as the implementation of the workflow system operations by using different application programs, which are modelled as components in the nodes of the system. These nodes are connected to each other to show the distribution of the business process operations among the different groups of the organisation. A deployment diagram is developed to depict this perspective. It is not possible to construct the deployment diagram at this stage of the workflow modelling, but it should be built after the workflow design stage, in which the required application programs are specified.

## 5.3 The Effectiveness of the SWfM Approach

The SWfM approach handles all the various perspectives of the workflow perspectives framework, soft, functional, informational, behavioural, operational, organisational, causality and quality, at different levels of coverage. SWfM may also be considered feasible and easy to implement and use in the real world through the systematic application of its various stages and steps. It involves two combined methodologies: SSM, which is supported by the tool that is developed in this research, and UML, which is the standard modelling language of the object-oriented paradigm and can be supported by many tools that assist in developing the required diagrams. Further, the approach seems to be flexible in offering the possibility of adapting the implementation of SSM to the situation under consideration and of adopting any modelling language that seems suitable to model the various perspectives of the workflow system, by replacing UML in the second stage of the approach.

## 5.4 Chapter Summary

In this chapter, the SWfM approach developed in Chapter Four has been implemented using an illustrative case study. The approach utilises SSM with UML to model the organisational business process as a workflow system. This approach consists of two stages, organisational business process investigation and workflow system modelling, to model the various workflow perspectives. The case study shows that implementation of the SWfM is feasible and practical. Although the implementation takes a long time, this is justified by considering the various issues of the business process at the investigation stage. Further, the SWfM approach covers the soft and hard issues of the business process by using both SSM and UML, which between them subsume adequate techniques to handle these issues. Finally, the models generated by this approach should be supported by a CASE tool that would assist in developing them, with the results being exported into any workflow system to be executed as a business process specification.

In the next chapter, a comparison is made of the outcomes of the SWfM approach with the leading workflow models discussed in Chapter Three: COSA as a production workflow model and WIDE as a research or prototype workflow model.

# Chapter 6: A Comparison of the SWfM Approach with Other Workflow Models

In this chapter, the approach proposed by the thesis, Soft Workflow Modelling (SWfM), will be compared with two of the leading workflow models reviewed in Chapter Three. The comparison is used to show the capabilities of the SWfM approach to cover the perspectives of the workflow perspectives framework, comparing that with other workflow models dealt with in Chapter Three. First, introductions to the three approaches are given; then they are compared by using the workflow perspectives framework of Chapter Three.

## 6.1 Comparison Models

The two workflow models chosen for comparison are the COSA and WIDE workflow models. This is because:

- They are the leading workflow systems according to the assessment made in Chapter Three. COSA [85, 86, 107-110] leads all the workflow systems and WIDE [84, 122-124] leads the workflow prototypes.
- They cover most of the perspectives of the workflow perspectives framework.

## 6.2 COSA

COSA has a client-server based structure. It is also a relational database-based application. It supports the distributed execution of workflows between several COSA systems and serverless computing, in which a user can carry out an assigned task while he is disconnected from the network. COSA provides several graphical tools to define the process and to manage and control the execution of a workflow. Further, there is a language to define cases, workflows, processes, data and organisation structures and a group language is used to assign users for activities. COSA provides a set of Application Programming Interfaces (APIs) that include all functions exported by the COSA server.

The following table (6-1), derived as a subset from Chapter Three, shows how the COSA workflow system handles the various perspectives of the workflow perspectives framework.

| Workflow Model / Perspectives | COSA |
|---|---|
| 1) Soft perspective | It provides several facilities for users to carry out their tasks. It also supports the ability to manually reroute work items to another workflow user. Further, it provides a facility that allows a user to reject a work item that has been allocated to a work queue. It supports delegation, in that it redirects all work items corresponding to a specific task to a specified user. It allows users to control the allocation and execution time of work items offered to them. It manages the workload across a process through the redistribution function. It links external work instructions to an activity. It provides a substitution feature. Finally, it handles exceptions by providing the definition of external 'triggers' or events. |
| 2) Functional perspective | COSA provides all the elements to model a business process (case, workflow, activity and task). |
| 3) Informational perspective | Files and unstructured variables, global or local, are supported. |
| 4) Behavioural perspective | The Petri nets technique and condition functions are used to depict the control flow. |
| 5) Operational perspective | A task is performed by a user or an application. Further, the application program is integrated through operating system invocation, and remote procedure call can be implemented indirectly. |
| 6) Organisational perspective | COSA involves expressive concepts to model an organisation structure. A group language is used to assign tasks to users, who are notified through memos. |
| 7) Causality | The case concept is used to group related workflows. COSA |

| | |
|---|---|
| perspective | Activity Manager manages cases and ad-hoc workflow. |
| 8) Quality perspective | There are several tools to maintain quality, such as a history access tool, a statistics tool, a form-based administration tool and reporting facilities, and a simulator tool. |

**Table 6-1: COSA Model perspectives**

## 6.3 WIDE

The WIDE project aims to provide the existing workflow management systems with advanced technology support through active database support and advanced transaction management in a distributed environment. WIDE has a client-server based architecture.

Table 6-2, derived from Chapter Three, shows how the WIDE workflow prototype covers the different workflow perspectives.

| Workflow Model / Perspectives | WIDE |
|---|---|
| 1) Soft perspective | The users provide a set of requirements for the application domain. Also, the users who execute the tasks have full freedom in the way the tasks should be executed. WIDE provides default exception handlers for all exceptional events defined in the system. |
| 2) Functional perspective | A workflow schema is used to define a business process that involves supertasks, tasks and multitasks. |
| 3) Informational perspective | Four types of data are supported: workflow schema information, shared databases information, documentation elements and temporal information. |
| 4) Behavioural perspective | Most of the basic control flow constructs are supported. |

| | |
|---|---|
| 5) Operational perspective | A task is carried out through an authorised agent that could be a human user, a software process or a combination of a user and a computer application. A wrapper module is used to integrate applications into WIDE. |
| 6) Organisational perspective | It includes several constructs to model an organisation structure. A task is assigned to an agent through a scheduler engine according to the assignment policy for automatic assignment, or directly to a human user. The push/pull model is used to assign a task to a user. |
| 7) Causality perspective | The supertask concept combines related tasks. |
| 8) Quality perspective | WIDE provides a workflow monitoring support tool and statistical information on the workflow execution for analysis. |

**Table 6-2: WIDE Model perspectives**

## 6.4 The SWfM Approach

The framework of the SWfM approach is the integration of Soft Systems Methodology (SSM) [41, 42, 142] with the Unified Modelling Language (UML) [145, 146]. It uses most of the SSM and UML techniques to model the various aspects of the organisational business process as a workflow system.

The SWfM approach was developed to model the various perspectives of the workflow perspectives framework of Chapter Three.

Table 6-3 below, derived from the SWfM approach in Chapter Four, demonstrates how the SWfM approach deals with the various perspectives.

| Workflow Model / Perspectives | SWfM Approach |
|---|---|
| 1) Soft perspective | SSM and its techniques handle this perspective by supporting the involvement of users in constructing the rich picture. The various points of view are considered during construction of the root definition and the conceptual model. These promote the acceptance of the resultant system and use it to perform the users' tasks and promote employee job satisfaction. |
| 2) Functional perspective | The process can be defined as a set of activities in the SSM conceptual model, modelled by a UML activity diagram. |
| 3) Informational perspective | An activity description table is used to present the informational perspective, which is modelled by a class diagram. |
| 4) Behavioural perspective | This is depicted partially using the SSM conceptual model and modelled in detail through the UML activity diagram. |
| 5) Operational perspective | A UML deployment diagram presents the various software and hardware components of the operational perspective. |
| 6) Organisational perspective | This is handled through the SSM conceptual model and Analysis Two technique and can be modelled using the UML class diagram. |
| 7) Causality perspective | This is described through the activity description table and is presented by the UML activity diagram and class diagram. |
| 8) Quality perspective | This is managed through the SSM conceptual model. |

**Table 6-3: The SWfM Approach perspectives**

## 6.5 Comparison

The comparison between the COSA workflow system which is discussed in detail in Section 3.2.2, the WIDE workflow prototype which is reviewed in detail in Section 3.2.10, and the SWfM approach is made using the workflow perspectives framework which was introduced in Chapter Three. The framework involves eight perspectives: the soft perspective and the seven other perspectives (functional, informational, behavioural, operational, organisational, causality and quality) which are a subset of the comprehensive workflow model of Jablonski and Bussler [38]. For convenience of the reader, a brief summary of the SWfM approach is provided here.

### 6.5.1 The SWfM approach

The SWfM approach combines SSM with UML to model the organisational business process as a workflow system. The approach covers almost all the perspectives of the workflow perspectives framework.

- SSM and its techniques handle the soft perspective by supporting the involvement of users in investigating the organisational business process and constructing the rich picture. Their various points of view are considered during construction of the root definition and the conceptual model. All these promote the acceptance of the resulting system and use it to perform users' tasks and promote employee job satisfaction.
- The functional perspective is defined completely through the conceptual model and the activity table, which show all the activities and tasks or steps of the organisational business process. Also, a UML activity diagram may be used to model this perspective.
- The informational perspective is specified by using the activity description table that presents the input and output data of each activity. To model this perspective, a UML class diagram is used. UML supports all the basic data types such as *integer*, *real* and *string*, and can model complex data structures.
- The behavioural perspective is depicted partially through the conceptual model that exhibits the sequential order of the activities. However, this is not enough to model the behavioural perspective. Therefore, A UML activity diagram is used to depict the basic control constructs of the behaviour perspective.

- The operational perspective is concerned with the implementation issues of the workflow system. A UML deployment diagram can be used to model this perspective by presenting the required software and hardware for a workplace within the workflow system. Further, it is possible to execute a task automatically or by assigning it to a user.
- The organisational perspective can be defined in SSM by specifying the conceptual model activities that the various roles in the organisation structure can perform and then use the Analysis Two technique to document these roles and their characteristics. This organisational structure can be modelled using a UML class diagram. Tasks can be assigned to users (human or machine) using the role concept. It is possible to adapt any assignment and resolution policy to this perspective.
- The activity description table describes the business rules and policies and their relationships to define the causality perspective. These rules can be presented in a UML class diagram and an activity diagram.
- The quality perspective can be supported through the conceptual model, which involves monitoring and control activities that track the performance, collect information and take control activity to rectify any deficiency in the system. These mechanisms can be implemented by providing suitable tools to perform their functionalities.

The following table (6-4) demonstrates the relationship between the COSA workflow product, the WIDE workflow prototype and the SWfM approach, using the workflow perspectives framework.

The workflow models are evaluated as a whole without considering a conceptual or implementation level. The conceptual model represents the basic concepts of the workflow model without implementation issues. The features of the conceptual level will be implemented with other features that are added through the implementation level. Because the SWfM approach is a conceptual workflow model it is recommended to implement it to include these features that are relevant to the implementation level. Thus COSA and WIDE workflow models are evaluated in general without distinguishing between the conceptual and implementation levels. On the other hand, the SWfM approach will be evaluated firstly on the conceptual level which is the objective of this research. Then the approach will be evaluated on the implementation level.

The schema that was defined to assess the workflow models in Chapter Three is used to compare these three workflow models in this chapter. This schema is as follows:

- 4 points: all the features of the workflow perspective are presented;
- 3 points: more that half of the features of the workflow perspective are presented;
- 2 points: at least half of the features of the workflow perspective are presented;
- 1 point: less than half of the features of the workflow perspective are presented;
- 0 points: the perspective is not supported by the workflow model.

| Workflow Model / Perspectives | COSA | WIDE | SWfM Approach | |
|---|---|---|---|---|
| | | | Conceptual level | Implementation level |
| 1) Soft perspective | 4 | 3 | 4 | 4 |
| 2) Functional perspective | 4 | 4 | 4 | 4 |
| 3) Informational perspective | 3 | 4 | 4 | 4 |
| 4) Behavioural perspective | 4 | 3 | 4 | 4 |
| 5) Operational perspective | 4 | 3 | 2 | 4 |
| 6) Organisational perspective | 4 | 3 | 3 | 4 |
| 7) Causality perspective | 3 | 2 | 3 | 3 |

| | | | | |
|---|---|---|---|---|
| 8) Quality perspective | 4 | 3 | 1 | 4 |
| Total | 30 | 25 | 25 | 31 |

**Table 6-4: The comparison of the workflow models**

### 6.5.2 The three Models compared

As it is shown in the above table the SWfM approach is compared with the other workflow models in two levels: conceptual and implementation levels.

#### 6.5.2.1 The conceptual level

From the above comparative data, it is possible to conclude that:

The COSA workflow system handles completely the soft, functional, behavioural, operational, organisational and quality perspectives. On the other hand, it supports more than half the aspects of the informational and causality perspectives.

The WIDE workflow prototype covers completely the functional and informational perspectives. It also supports more than half the aspects of the soft, behavioural, operational, organisational and quality perspectives and at least half of the aspects of the causality perspective.

The SWfM approach deals completely with the soft, functional, informational and behavioural perspectives, while supporting more than the half of the aspects of the organisational and causality perspectives, at least half of the aspects of the operational perspective and less than the half of the aspects of the quality perspective.

In conclusion, the SWfM approach covers completely the informational perspective, while the COSA model provides more than half of the data structures required in the informational perspective. The SWfM approach is better than WIDE in covering all aspects of the soft perspective, offering all the basic control constructs in the behavioural perspective and handling business policies in the causality perspective.

#### 6.5.2.2 The implementation level

In this level the deficiencies of the other three perspectives, operational, organisational and quality, which are related to the verification and implementation issues of the workflow model, can be improved through the provision of these facilities and tools. Thus it is possible to add these features during the implementation stage to improve the quality of the approach. In the following the required facilities and tools to improve the SWfM approach will be discussed.

For the operational perspective, it is possible to specify a notification method to inform the users about the assigned tasks (worklist mechanism for each user or a shared place for all users based on a role concept) and an integration mechanism (such as Remote Procedure Call (RPC), Object Linking and Embedding (OLE), or through CORBA) to integrate the required application programs to the workflow model to perform the assigned tasks automatically or to assist the users in carrying out their tasks.

For the organisational perspective, users notification can be implemented through using the pull or push model. The pull model improves the users' abilities to manage and execute the assigned tasks without forcing them to do that.

For the quality perspective, it is possible to provide several facilities and tools to implement the required quality aspects. These facilities can be a history tool, a statistical and analysis tool and a simulation tool.

If these issues are implemented, they will improve the SWfM approach and increase its score levels. As a result of that the approach will cover completely all the perspectives except the causality perspective in which the missing feature is covering the strategic level of the organisation during considering the plan for the workflow system. Hence the SWfM approach will lead the two workflow models, COSA and WIDE.

## 6.6 Chapter Summary

In this chapter, the SWfM approach, using two levels (conceptual and implementation levels) has been compared with two workflow systems, using the context of the workflow perspectives framework. The COSA workflow system covers almost all of the perspectives; while the WIDE workflow prototype and the SWfM approach each handles most of the perspectives in the conceptual level. If the SWfM approach is implemented, it will almost handle all the perspectives and will lead the two other workflow models.

In the next chapter, the CASE tool is discussed as an important element of the modelling approach and an SSM CASE tool is developed, using the Microsoft Visio drawing software and Visual Basic for Applications.

# Chapter 7: Tool Support for the SWfM Approach

This chapter introduces the drawing tool of the Soft Workflow Modelling (SWfM) approach, which is used to construct a Soft Systems Methodology (SSM) conceptual model, considered as the basis for the other developed models. This tool illustrates the practicality of the SWfM approach and can be implemented in the real world. The chapter begins with a brief history of Computer Aided Software Engineering (CASE) tools and their types in general and of workflow definition tools in particular. Then the SSM notation, which is a part of the SWfM approach, is introduced for modelling the SSM conceptual model. After that, an attempt is discussed to integrate the SSM tool with Visual Studio .net and some workflow systems. Finally, a discussion of the disadvantages and problems of CASE tools concludes the chapter.

Further information on CASE tools (an introduction, a business view, CASE tool structure) and on the choice of drawing software for CASE tools is given in Appendix B.

Before the term 'CASE tool' is defined, the developmental history of CASE tools must be outlined.

## 7.1 The History of CASE Tools

CASE tools have been around since computers were invented, originally in the form of mainframe-based utilities, which were programs supporting software development, such as compliers, linkers and editors. Then, during the 1970s and early 1980s, further CASE tools were introduced, most designed to support the back end of the system development lifecycle (coding and testing), while a few were used to support software project management. Further, some complex tools emerged and several simple tools were combined to offer support to the whole system development lifecycle. When the analysis and design methods became widely used, the analyst workbench, which was used to support the analysis and design activities, started to emerge; then project management workbenches began to become available. Finally, the term 'CASE tool' was introduced to describe tool-sets that provided support to the system development stages in what was called the project-support environment [161].

Avison and Fitzgerald [43] define a CASE tool as 'any integrated computer software system that is specifically designed to support a significant part of the information system development process of an information system and the management of these tasks and processes'. More broadly, Mair [161] defines CASE tools as 'programs to assist in the development and maintenance of computer systems'. Finally, the CASE tool can be considered as the automation of the entire systems development process or part of it.

## 7.2 The Classification of CASE Tools

There are many classifications of CASE tools, one of which groups them into upper and lower CASE tools, depending on the support provided to the stages of the development process: upper CASE tools aid the strategy planning, analysis or logic design stages, while lower CASE tools support physical design, programming and implementation. There are also integrated CASE tools, which integrate tools horizontally or vertically. Horizontal integration is the bringing together of various tools at a particular stage of the development process, such as the analysis stage, which includes several techniques: data flow diagrams, entity data diagrams and so on. In this case, any change in one diagram must be reflected in the other diagrams where appropriate. On the other hand, vertical integration concerns the integration of tools between several stages of the development process, which means that the deliverables of one stage should be available in an automated way to the other stages. The passing forward of deliverables from an earlier stage to subsequent stages is known as forward integration, while the passing backward of deliverables from later to earlier stages is known as reverse engineering [43]. In addition, Mair [161] classifies CASE tools into four groups depending on their functionalities and their scope to support the various stages of the system development life cycle. These are simple tools, workbenches, integrated CASE (ICASE) tools and open environments.

### 7.2.1 Simple Tools

These are used to perform one specific task. There are two types: mainframe/mini tools and micro tools. The former subsume tools such as compilers, linkers and editors, while micro tools include simple diagrammatic and project management tools. These store data in their file format and some provide interface facilities with other tools to import and export files.

### 7.2.2 Workbenches

A workbench is a set of tools that support one stage of the system development lifecycle. There are three types of workbench: analyst, programmer and project management.

### 7.2.3 Integrated CASE

ICASE products include tools that assist both system development and project management activities, thus supporting the whole system development lifecycle. They incorporate a collection of one or more specialist workbenches that share a common database. The possibility of adding tools to the ICASE range of products is very limited, because such tools are tightly coupled and support one specific method.

### 7.2.4 Open Environments

These are developed to assist all stages and roles of the system development life cycle. They offer facilities to support system development, project management and control and office automation. They have the ability to integrate other tools and methods and evolve with the organisation's development.

## 7.3 Workflow Definition Tools

Workflow tools facilitate the modelling, design and implementation of workflows [162]. There are three types: process definition tools, resource classification tools and analysis tools, which are integrated into a workflow definition and analysis tool [163]. The Workflow Management Coalition (WFMC) uses the term 'process definition tools' to refer to the three types of tools.

Process definition tools are used to describe the process. They usually involve a graphic tool for drawing a workflow using icons that represent steps and tasks in the business process [162]. The basic functionality of the tool includes:

- the ability to define a process and model the various routings;
- supporting the version management;
- supporting workflow attributes and task specifications;
- validating the syntactical correctness and consistency of the process [163].

Resource classification tools are used to classify resources and to define the relationships between the tasks and the various resources. Resources can be classified according to roles or organisational units [163].

Analysis tools are used to analyse and verify the process definition before putting the process into production. This includes validating the semantic correctness of a process definition and performing simulations to determine the completion time for process instances and the process bottlenecks [163].

## 7.4 Rationale for a Drawing Tool in the SWfM Approach

As a part of the SWfM approach, a drawing tool was developed to assist the users to build a conceptual SSM model. Although there are a few tools that are used to draw such models (e.g. MooD, from the Salamander Organisation), most practitioners use pen and paper. A simple SSM conceptual model notation is constructed below by using Microsoft Visio for designing the various shapes and Visual Basic to program the properties of these shapes and enforce the conceptual model rules. This tool will help the faster development of the conceptual model, allowing it to be presented in a professional way. Modelling the conceptual model by using the SSM tool will make the conceptual model more easily modifiable to reflect any changes in organisational activities and processes.

## 7.5 The SSM Tool

### 7.5.1 Visio Software

Visio is a drawing package from the Microsoft Corporation. It is used to create several types of diagram for business and professional purposes, and can be integrated with other Microsoft Office products. Visio includes several stencils and templates that are used to develop the required diagrams. Each diagram is made of shapes which can be anything: circles, squares, arrows, maps, etc; they come in three types: 1D shapes, such as lines, 2D shapes, such as squares, and 3D shapes, such as boxes. These shapes are organised into stencils, which are arranged by topic and type of shape. Also, stencils, macros and drawing page settings are combined into a template, which is 'a Visio file that opens one or more stencils and contains styles and settings for a particular kind of drawing' [164].

### 7.5.2 Tool Features

An SSM tool was developed to draw the SSM conceptual models that are used in the first part of the SWfM approach. This tool was designed by using the Visio software and Visual Basic, which is a Microsoft package used to develop an application environment as a fourth generation language. The developed tool had to involve several features to assist the user in building the various models. These features are as follow:

- The tool should have a graphical user interface which makes it easy to learn and use to develop models.
- It should also have the ability to do anything in several ways, such as through menus or shortcut buttons/keys.
- It is desirable that the tool can provide symbols that allow the user to create professional diagrams by customising these shapes and to create new shapes by customising the existing ones or defining new ones.
- It is also desirable for the tool to provide facilities to manipulate the various diagrams by generating an XML (eXtensible Markup Language) format for them.

Visio satisfies the required features of the developed tool. Thus Visio was the obvious choice, but some other possibilities were considered, as described in an evaluation of drawing packages (Appendix B).

### 7.5.3 SSM Drawing Tool

The drawing tool has the following functions:

i. Develop a stencil for the SSM conceptual model notation

The SSM notation stencil includes the various shapes or symbols used to develop a conceptual model, consisting of the different activities and the logical dependencies among them. Some of these shapes are adapted from the predefined Visio shapes and others are developed by grouping the predefined shapes with text. Some of the shape features are modified, either by altering their formulae in their ShapeSheets, which are databases used to keep track of all shapes and contain programming information for them, or by the use of macros written in Visual Basic to alter the behaviour of existing shapes. These shapes were introduced in Figure 4-2 in Chapter Four and are repeated in Figure 7-1 below, derived from [153].

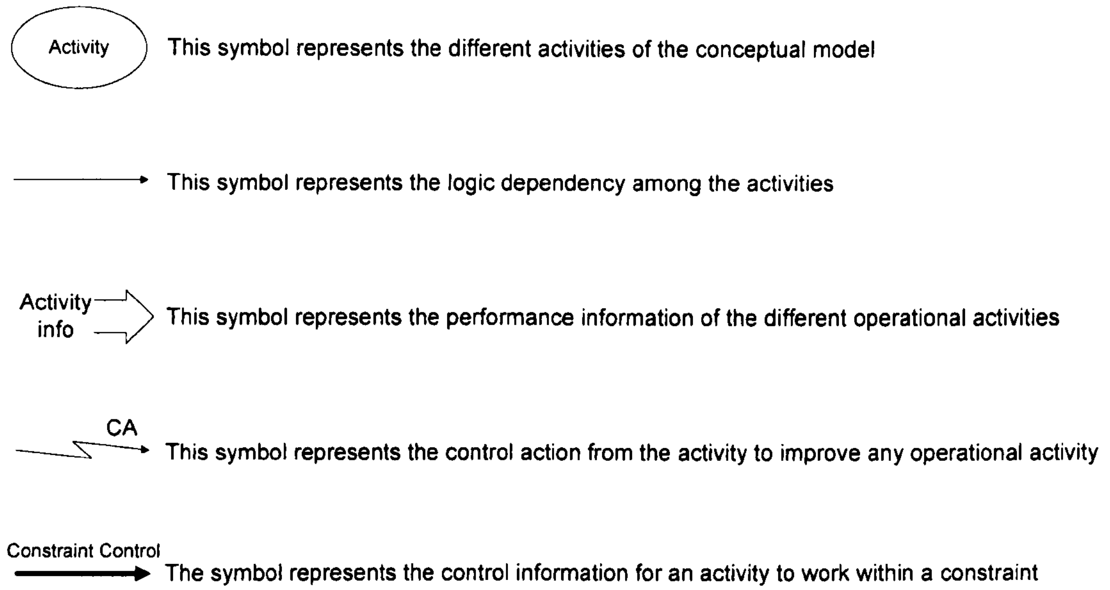



**Figure 7-1: The SSM conceptual model notation**

ii. Develop an SSM conceptual model

The SSM notation is used to draw a conceptual model in the drawing page, which is provided by the SSM notation template. This template includes some macros to be implemented in the SSM notation stencil and the drawing page, facilitating the drawing of conceptual models and providing a professional look for the diagram, which makes it better than using pen and paper.

iii. Check the conceptual model syntax

This feature is to check that the conceptual model shapes are used properly. It may be considered as a monitoring tool to help the user during the construction of the conceptual model. The user has to choose the required shape to construct the conceptual model from the SSM notation stencil or template provided. If the user chooses a shape from a different Visio stencil or template, an error message is displayed: "The shape is not an SSM notation shape! Choose a shape from the SSM stencil"; and the shape is deleted from the drawing page in line with the restrictive philosophy of the methodology support [165].

iv. Check the conceptual model connection

This feature is used to check the connection of the conceptual model; each shape in the model should be connected. The user can invoke this function to verify the conceptual model connection. If the model is connected, it returns a message to indicate that this is so; otherwise, it returns a warning message if there is an unconnected shape. This illustrates a flexible aspect of the methodology support philosophy [165].

v. Provide help and assistance for the users of the SSM notation

The stencil offers help to the user by providing introductory information about the SSM notation and some information about each shape. There are two ways to access the help information: through the help button in the tool bar of the drawing page or through the right menu of each shape in the stencil.

vi. Possibility of integration with other packages to generate application code

One of the features of Visio is the ability to generate an XML format for the developed diagrams. Thus, the SSM notation can be used to develop a conceptual model that can be transferred into a use case model of a Visio template to develop the other UML diagrams of the SWfM approach. Then it is possible to generate an XML format for these diagrams to export them to other workflow products which have the ability to import an XML format of these diagrams and execute them through the workflow engine.

### 7.5.4 An Example of Using the Tool

The SSM tool runs under the Windows operating system and is implemented by using the Microsoft Visio software. In this example, the tool is used to model the registration subsystem within the course management case study set out in Chapter Five. The SSM notation is a Visio template that consists of the SSM notation stencil, which contains the various shapes of the notation of Figure 7.1, a drawing page to model the conceptual model, and the various macros that are written by using Visual Basic for Application to implement the different functions of the notation template. The shapes of the stencil are used to model a conceptual model, or part of it, by following the rules for constructing an SSM conceptual model. Any shape in the stencil can be dragged and dropped into the drawing page. The tool checks the source of the shape; if it is taken from the SSM notation stencil, it can be dropped into the drawing page, but if it is not, an error message is returned and the shape is deleted. The activity shapes are connected by logical dependency arrows; however, the other shapes are used on certain occasions, such as collecting performance information, providing control information or taking action to improve the situation. The activity shape is the only shape in which it is possible to type text by double-clicking on its icon. The other shapes return the error message "Illegal operation" when they are double-clicked. The activities of monitoring the system and taking control action to improve the situation are added to complete the construction of the conceptual model.

The following screen shot shows the final conceptual model for the registration subsystem that involves its various activities and the SSM notation template.

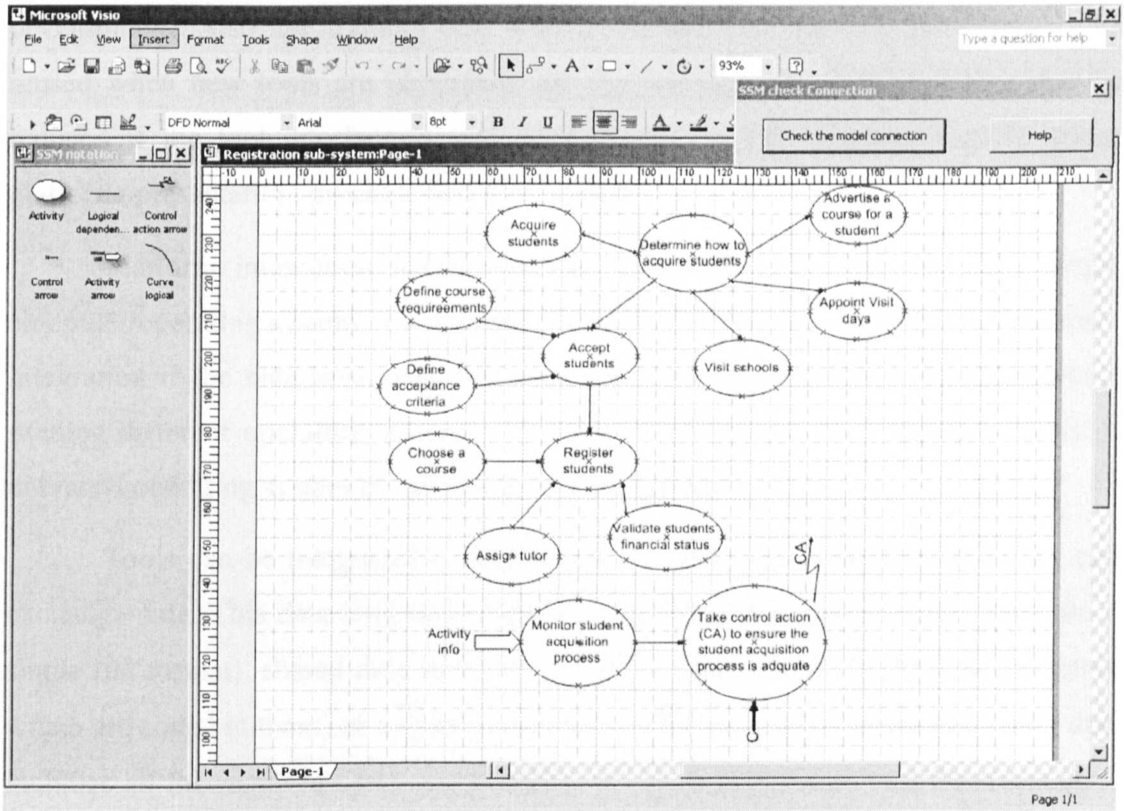



**Figure 7-2: A Screen shot of the Registration Subsystem**

After constructing the conceptual model, it is possible to check its connection by choosing the toolbar option "Check the model connection". If all the shapes are connected to each other, using the logical dependency arrow and the other arrows, the tool returns the message "All connections are valid"; otherwise, an error message is returned, declaring that "The model is not completely connected" and requiring the checking of the unconnected arrow by name and pointing to the arrow if it is a logical dependency arrow only. Then the conceptual model file is saved by giving it a name and indicating the directory in which it is to be stored.

The template contains help files that contain general information about the SSM notation and the various shapes of the stencil. These files are available through the Help button on the SSM notation tool bar or through the SSM stencil by right-clicking on a shape.

## 7.6 CASE Tools Integration

Case tools integration is an important issue that allows different tools from different vendors to be integrated so they can be used to cover the entire system development. Also, the training cost will be decreased, as existing software may be reused when new tools are integrated and the learning time and error rate will be reduced if the tool interface is integrated. Tools can be integrated in five ways: platform, presentation, process, data and control.

Platform integration means that the integrated tools will run on a single computer/operating system or a network of systems. There is a problem with platform integration if the enterprise uses heterogeneous networks, with different computers running different operating systems. To solve this problem, it is important to use a universal operating system or operating system standards.

Tools can be integrated through using the same data model so that they can exchange data. This data integration can be done through shared files (all tools use a single file format), shared data structure (all tools agree to use shared data structures which are coded in them) or a shared repository (all tools share a single repository that includes all the required data). Each approach has its advantages and disadvantages, which should be considered when integrating tools around data models. The simplest approach is to use a shared set of files. It is difficult to integrate tools by using a shared data structure, because this requires knowing the details of the shared structures. The integration of tools using a shared repository is considered the most flexible approach, in which a pubic database schema that defines the required entities and their relationships is used to read and write data to the database.

Presentation or interface integration allows the user to interact with the various tools through a common interface that implements interface standards. This helps to reduce the learning curve for the user. There are three types of interface integration: window system integration, in which a common interface for the window system is used for all tools; command integration, in which a common format of commands is used to perform certain tasks on all tools; and interaction integration, in which the user interacts with the various tools in the same way and uses certain commands to perform the same tasks.

Some tools can control the activation of other tools in the integrated environment through the mechanism of control integration. A tool can invoke the services of other tools through a program interface that define the facilities which a tool offers to other tools. The tools communicate and exchange information by using the message-passing mechanism of a generic message server that manages the communication among them.

Finally, tools can be integrated through process integration, in which they support a specified software process. The integrated tools or environment must have a process model that defines the software process phases, activities, deliverables, rules to implement it, and the tools which are required to support its activities. This model is interpreted by a process engine to guide the software process [166].

Thus, it is important to define and use standards that allow vendors to develop well-integrated tools to construct an integrated environment covering the whole lifecycle of the system development. These standards cover four important areas: the CASE Data Interchange Format (CDIF) standard, developed by America's Electronic Industries Association, is used to define the exchange of information between CASE tools and repositories; the Semantic Transfer Language Standard, developed by IEEE, is concerned with tool-to-tool communications; the Information Resource Dictionary System (IRDS) standard defines the data model for the information that will be stored in the repository and the methods to be used by tools to access this information; and the Portable Common Tool Environment (PCTE) standard, developed by the European Computer Manufacturers' Association (ECMA), allows the use of tools across different operating systems and local area networks (LANs) [167].

There have been several attempts to develop frameworks for the integration of CASE tools.

Bussow et al. [168] introduce a framework to combine established modelling techniques and tools with formal tools as part of the ESPRESS project funded by the German Bundesministerium fur Bildung, Wissenschaft, Forschung und Technologie. The Statecharts technique, which is a graphical notation for reactive behaviour, is combined with the formal Z notation and with temporal logic. The resulting notation is supported by a specification methodology and an open tool integration environment. Bussow and colleagues tried to implement three types of integration among tools: data integration, by developing uniform data formats into which tool data can be mapped, control integration, through the use of the PIROL workbench, and presentation integration, by implementing modern design principles for graphical user interfaces (GUIs) to allow all tools to be accessible from a common user interface.

Sattler [169] presents a framework approach for control integration in tool environments. It is based on a model in which tools are connected through interactors as semantic relations. The framework consists of two parts: the tool interconnection language (TIL) which is used to describe the structural and behavioural properties of the tool integration, and the run-time environment of the framework, which supports the instantiation and interconnection of the tool components.

Karsai and Gray [170] describe a framework or an approach to tool integration by integrating their data models, using generators that create componentised semantic translators. The approach is one of semantic mapping or semantic translation, in which the semantic relationships among the data models of the tools to be integrated are assembled, after which semantic translator tools can be developed to implement the mapping. This approach is based on developing an integrated data model rich enough to incorporate the data models of the tools to be integrated. The architecture of the approach has two major components: the Integrated Model Server (IMS) and the Tool Adaptors (TA), with Common Object Request Broker Architecture (CORBA) used to communicate between them. The Integrated Model Server (IMS) provides semantic translation services to transform data from one data model into another, while preserving the semantics of the input data model and enforcing the semantics of the output data model. On the other hand, the Tool Adapter (TA) reads the tool specific data and changes it into a form that can be understood by the IMS and vice versa. It is thus a bi-directional adaptor.

### 7.6.1 Visio Integration

The SSM notation tool can be used to develop an SSM conceptual model which is used to construct a UML use case model of the organisational business process; then the various UML diagrams may be used to model the different perspectives of the workflow system. Microsoft Visio provides support for creating object-oriented models using the Visio UML template. Visio integrates with other tools by supporting XML Metadata Interchange (XMI), allowing for the exchange of models. There are several workflow systems that import Visio UML diagrams: Microsoft's tools for Business Process Management (BPM), Scrittura's web-based technology platform, IBM's WebSphere Business Integration Modeller, Proforma's ProVision Suite, COSA, Staffware and SAP.

Microsoft provides tools for BPM and supporting technologies facilitate the creation of processes. These tools include BizTalk Server, Visual Studio. net, SQL Server, Office, Active Directory, Host Integration Server and BizTalk Server Adapters, and Microsoft Operations Manager and Application Center. Microsoft Visio generates XML representations of the UML diagrams of a business process in a Business Process Execution Language (BPEL) document. The BPEL, using the Microsoft Tools for BPM representation of the business process, is imported into Visual Studio .net, which is a visual development tool for building sophisticated workflows and processes. The assembled process generates an XML based run-time script (BPEL) of the process that is executed in BizTalk Server, which is Microsoft's central platform for Enterprise Application Integration (EAI) and BPM and involves the integration and automation capabilities of XML and Web Services technologies. BizTalk Server functions as a process execution engine and as a multi-transport hub for messaging and document transformations [171].

Scrittura enables customers to use Microsoft Visio to visually describe their business needs with diagrams created by Workflow Manager's graphical workflow design tool. The design tool plug-in to the Visio program uses the standard UML Activity Diagram model and converts the Visio diagram to workflow commands [172].

IBM WebSphere Business Integration Modeller is a business process modelling tool that is used to model, design, analyse and generate reports for business processes. The Business Integration Modeller facilitates the import of models and definitions from Microsoft Visio (.vdx). It is then possible to export models in Flow Definition Language (FDL) format to the WebSphere MQ Workflow product to automate and execute the process [173].

Proforma's ProVision Suite provides modelling, analysis and simulation tools for all aspects of the enterprise involving strategy, processes, systems and technology. There are two versions of ProVision: ProVision Enterprise and ProVision Process. Data Exchange of the suite allows the importing of business and UML/object models from Microsoft Visio through MS COM (Component Object Model) or XML [174].

COSA workflow defines the business process through using the COSA process Designer. The business process models' script files are stored in XPDL format which is an XML format that is created by the WfMC for archiving business processes which are stored in the underlying relational database. Also COSA workflow supports importing XML format file. Thus it is possible to import the workflow models of Visio into COSA workflow and execute them by the workflow engine [175].

Staffware workflow specifies the business process through TIBCO Staffware Process Suite which provides a user-friendly and graphical modelling environment. Further, Staffware workflow includes a complete XML infrastructure solution such as TIBCO Turbo XML, TIBCO XML Validate, TIBCO XML Transform and TIBCO XML Canon for developing and deploying XML-based business processes. Thus the workflow models of Visio can be imported into the Staffware workflow and can be executed by the TIBCO iProcess Engine. The XML infrastructure solution involves: TIBCO Turbo XML for generating and amending XML schemas and Document Type Definitions (DTDs), TIBCO XML Validate for certifying and executing XML in a run-time environment, TIBCO XML Transform for mapping and transforming XML documents and TIBCO XML Canon for storing, managing and distributing XML aspects [176].

SAP workflow defines the business process though using SAP NetWeaver that incorporates the SAP business workflow tool which can be used to enable three level of process design: Business level, configuration level and execution level [177, 178]. The business process models can be imported from Microsoft .Net to SAP which supports bidirectional communication with Microsoft .Net through SAP .Net Connector in which SAP applications can access and integrate .Net services while projects developed in a .Net environment can access SAP business functionality. SAP .Net Connector involves a comprehensive support for visual studio .Net IDE. Further, SAP NetWeaver supports interoperability at the process level through SAP Exchange

Infrastructure (SAP EI) which includes an integration broker and business process management (BPM) capabilities. SAP Exchange Infrastructure can import Web Services Description Language (WSDL) from the Microsoft .Net into its repository and directory. Also, SAP EI supports Extensible Stylesheet Language Transformation (XSLT) mappings that can be used to map to and from SAP and Microsoft .Net. On the other hand, BPM of SAP maintains the Business Process Execution Language for Web Services (BPEL4WS) standard to facilitate the import and export of business processes specifications with Microsoft .Net [179].

## 7.7 Tool Support for Developing a Workflow System

It is important to use a tool to develop and maintain the required models throughout the various stages of the approach.

The SWfM approach tries to integrate the models or deliverables of the soft and object-oriented paradigms by using SSM techniques such as rich picture, root definition (RD), conceptual models and transition models for the former and various UML diagrams to cover the latter. The SWfM approach consists of two stages: the organisational business process investigation and the workflow system modelling, as set out in Chapters Four and Five.

The organisational business process investigation is dominated by the use of SSM and its models.

The rich picture model is used to depict the various issues of the problem situation in an informal way. Therefore, a suitable drawing package includes several symbols or a large clipart library to model the various aspects of an organisation. This model is not connected with the other models of the SWfM approach. Its aim is to present the problem situation in a way that first stimulates debate and discussion about it and later facilitates documentation of the problem situation.

The RD is a description of one world view among the various ones that exist in the problem situation. It is considered the basis for building the other models and diagrams in the SWfM approach.

The conceptual model (CM) depicts the RD by presenting the various activities that fulfil it. The tool assists in drawing the SSM conceptual model and then validates its characteristics by implementing the various rules of the formal model.

The resulting models are saved and can be updated to reflect any change in the organisation or its environment.

The workflow system modelling stage uses various UML diagrams to model the different perspectives of the workflow system.

The SSM conceptual model activities are decomposed into subsystems to facilitate their management and a table is constructed for each subsystem to present its activities.

Another table is built for each activity of the conceptual model to show its various parts in detail. This can be connected with the next model, which is the transition model of the tool, where clicking any activity in the transition model should display its detail.

The transition model is similar to the conceptual model, but includes a goal for each activity to facilitate the transition of the CM activities to UML use cases. It is considered a bridge between SSM models and UML diagrams.

UML diagrams involve the use case, activity, sequence, class and deployment diagrams that are used to model the various perspectives of the workflow system. The use case diagram is constructed from the transition model by developing an interface between the activities of the transition model with the use case diagram of the Visio UML template. Next, the other required diagrams of the Visio UML template are used to depict the various perspectives of the workflow system. Finally, it is possible to export the Visio UML diagrams to any workflow system that supports the importing of model specifications in XML format. A workflow system has a definition tool that is used to describe the business process or workflow model. This tool can be used either directly to define a business process (workflow) or to import the business process (workflow) specification from another tool. Visio generates files in XML format, facilitating interoperability with other tools. These diagrams are executed by the workflow system through its workflow engine. For example, Microsoft Visio generates XML representations of the UML diagrams of a business process in a BPEL document. The BPEL using the Microsoft Tools for BPM representation of the business process is imported into Visual Studio .net to build sophisticated workflows and processes. The assembled process generates an XML-based run-time script (BPEL) of the process that is executed in BizTalk Server [171].

## 7.8 Maintenance of the SWfM Approach Models and Descriptions

A CASE tool assists in changing the models in an easy way and in many cases supports the retention of consistency between models. Therefore, any change in one model should propagate relevant changes to the other models.

The SWfM approach consists of successive steps, each based on the previous one. The nature of any system and its environment is dynamic and continually changing; thus it is essential to manage any change and reflect that in the models.

The ramification of any change should be assessed and its impact on the other models of the approach determined. This effect can work in two ways: forward and backward. In the forward direction, the change will affect the successive models that are based on the model where a change has occurred, while in the backward direction, the change in a model will affect the models on which the changed model is based. It is important to determine the effect of a change and to maintain consistency between the models by studying and analysing all the possible changes and their effects.

For each diagram or model, it is possible to carry out three basic actions on its constituent elements: addition, modification and deletion. In the following, each diagram, model and technique of the SWfM approach will be assessed against these three actions to quantify the effect of changes on the other models.

- Rich Picture Model Changes

  A change in the rich picture model may result in changing some issues inside an organisation or in an organisational environment. This causes a change in the RD that describes the various aspects of an organisation and its environment.

- Root Definition Changes

  An RD will be changed to reflect the change in the rich picture to include the new issues and to modify or delete some existing aspects.

- Conceptual Model Changes

  The conceptual model will be changed as a result of changing the RD. New activities and dependencies may be added and existing activities and dependencies may be modified or deleted. The new conceptual model will involve only the activities that present what the RD is supposed to do.

- Subsystem Table Changes

  The activities of the subsystem table will be changed to reflect the changes in the conceptual model. New activities are added and existing activities are modified or deleted. Thus, the subsystem tables include only the activities of the conceptual model.

- Activity Table Changes

  The activity tables will be built for the activities that exist in the subsystem tables. Therefore, new activity tables are developed for the new activities and the existing activity tables may be modified or deleted to reflect the changes in the conceptual model and the subsystem tables.

- Transition Model Changes

  The transition model will be changed as a conceptual model; new goals will be added to the new activities and the existing activities with goals might be modified or deleted.

- UML Diagram Changes

  In the UML diagrams, the central diagram is the use case one. The other UML diagrams present the various use cases of the use case model. Thus, any change in the use case model will be reflected in the other UML diagrams, namely the activity, sequence, class and deployment diagrams.

The following matrix (Table 7-1) summarises the way changes in each technique of the SWfM approach will affect each of the other techniques.

| Affected techniques / Techniques | Rich picture | Root definition | Conceptual model | Subsystem table | Activity table | Transition model | UML diagrams |
|---|---|---|---|---|---|---|---|
| Rich picture | | √ | | | | | |
| Root definition | | | √ | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Conceptual model | | | | √ | √ | √ | √ |
| Subsystem table | | | | | √ | | |
| Activity table | | | | | | √ | |
| Transition model | | | | | | | √ |
| UML diagrams | | | | | | | √ |

**Table 7-1: The effects of technique change on the other techniques of the SWfM approach**

## 7.9 Advantages and Disadvantages of CASE Tools

In this section, the advantages and disadvantages of CASE tools will be discussed. Most of the latter are related to the high cost of acquiring the CASE tool. This cost has several elements, including the training of staff to use the tool. This is considered a long-term cost, due to staff turnover and new versions of the tool. Also, it takes time and effort to adapt any tool to suit the organisation and set up its environment for use through customisation. Sometimes, the CASE tool requires consultants or other experienced people from outside the organisation to work with it, which adds some cost. Finally, the CASE tool may require an upgrade of the existing hardware to allow it to work properly. It may also restrain the freedom of developers or users by implementing the methodology rules to which they must adhere [43].

These problems associated with the cost of CASE tools can be balanced by considering its benefits. CASE tools can help to improve productivity through the rapid development of information systems by automating several parts of the development process and using fewer people to develop a system, which reduces communication problems, leading to a reduction in the cost of the development. Also, it is possible to reuse some parts of earlier projects that can be found in the repository, which reduces the time and effort required for system development. In addition, CASE tools help to reduce the maintenance time and effort and enhance the existing systems by providing several facilities, such as good and consistent documentation, which ease the maintenance and the automation of design and implementation, resulting in fewer errors at the programming, testing and implementation stages. Moreover, the tools can provide system testing by generating test cases, applying them and analysing and comparing the results obtained. These cases are derived from the tool repository. In addition, some tools provide reverse engineering, which assists in re-engineering the old or legacy systems to save some of the resources required to maintain them. Some tools also help to make the developed system portable by generating code for a variety of programming languages on different hardware platforms [43].

## 7.10 Chapter Summary

In this chapter, an outline of the history and classification of CASE tools has been presented and a tool developed for the SWfM approach. This shows that the SWfM approach can be implemented using a CASE tool and illustrates the practical applicability of the SWfM approach. The tool assists the analyst to draw and present the required models in a professional way. It also provides some validating mechanisms for some of the model characteristics. Finally, the possibility was discussed of integrating the developed tool with other workflow systems to execute and implement the diagrams of the SWfM approach, as were the advantages and disadvantages of CASE tools.

# Chapter 8: Discussion and Conclusions

This research has investigated modelling the organisational business process by considering its soft and hard issues and developed an approach to model it as a workflow system. A Soft Workflow Modelling (SWfM) approach was developed by combining Soft Systems Methodology (SSM) with the Unified Modelling Language (UML) to address some chosen perspectives of the workflow system. This chapter reviews briefly the whole thesis and draws out some results of the research. Then, the SWfM approach and the work done are evaluated and compared with a selection of related work and projects. Next, the limitations of the present approach are addressed. Finally, areas of further research and investigation are suggested, and some concluding remarks are made.

## 8.1 Overview

The research is concerned with the soft and hard aspects of the business process. The research question as stated in Chapter One is: *how to model the internal organisational business process by considering its soft and hard issues in order to develop a workflow system.*

Chapter One set the scene by specifying the need for an approach that models the workflow system by considering the hard and soft issues of an organisational business process. The internal organisational business processes, public business processes that cross an organisation boundary and workflow management systems were discussed in general.

In Chapter Two, aspects of the organisational business process and workflow system were reviewed in detail, and some selected workflow models were compared in Chapter Three in the context of a developed workflow perspectives framework. These perspectives are considered as essential constituents of any workflow system. The result of this comparison promoted the idea of developing an approach to modelling a workflow system by covering the hard and soft issues of the organisational business process.

This approach, SWfM, which was introduced in Chapter Four, utilises the Soft Systems Methodology (SSM) to address the soft issues and the object-oriented paradigm through standard modelling language, Unified Modelling Language (UML), to cope with the hard issues.

In Chapter Five a case study, course management in a university, was chosen to test the approach by implementing it.

The SWfM approach, using two levels (conceptual and implementation levels) was compared with two workflow models, COSA and WIDE, in the context of the framework of workflow perspectives, in Chapter Six.

In Chapter Seven the importance and the related issues of CASE tools were discussed and a tool was developed and used to construct the conceptual model of the SSM. It was shown to provide several features that assist the developer in constructing a conceptual model and verifying its correctness and consistency.

## 8.2 Contribution of the Research

This research has contributed to the workflow system field through the following advances:

- Developing a workflow perspectives framework including a soft perspective

A framework has been developed which includes seven perspectives of the workflow reference model in Chapter Three and a soft perspective is added to represent the soft issues of the organisational business process. This framework has been used as a guide for comparing some workflow models to show the capabilities of each workflow system to cover these perspectives (Chapter 3).

- Enhancing the modelling of organisational business process

This research has provided a new way to model the organisational business process that involves soft and hard aspects in order to implement it in a workflow environment. Most of the approaches focus on one side and neglect the other, which results in many problems concerning the use and acceptance of the developed workflow system. A new approach, SWfM, has been developed as a result of the analysis and the comparison of workflow models. The approach connects a soft system approach, SSM, with a hard system modelling language, UML, to gain the benefits of both approaches with minimal disadvantages. The SWfM approach adopts SSM to investigate the organisational business process, consider the various points of view of the people involved, model the organisation activities and provide alternatives to improve the organisational business process. Then UML, as a modelling language of the object-oriented paradigm, is used to handle the aspects that SSM cannot handle. This involves several diagrams that can be used to depict the various perspectives of the workflow system. Although some UML diagrams have limitations in modelling the perspectives, they are considered useful techniques; and some improvements are added into them in UML 2 (Chapter 4).

- Combining several methodologies or methods to complement their roles

This research shows the ability to combine various methodologies or methods to benefit from the positive features and strengths of each. The SWfM approach combines SSM with UML by transferring SSM conceptual model activities into UML use cases using the goal concept. Therefore, the resulting approach covers the soft and hard issues of the organisational business process. Moving from one methodology or method to another should be smooth and without loss of information through the transfer stage. Thus the UML use case is used as a general technique that can accommodate the detailed information of a conceptual model activity through its main scenario and the alternatives.

- Depicting the applicability of the research approach

The SWfM approach was tested and implemented using an illustrative case study, the management of a large university course, to demonstrate its ability to handle a complex scenario involving both human and hard aspects of a workflow system. The case study indicates that the approach is applicable through the use of the various techniques, as presented in Chapter Five.

- Comparing the SWfM approach with other workflow models

The SWfM approach has been evaluated against two of the workflow models of Chapter Three to represent its capabilities using the workflow perspectives framework of Chapter Three (Chapter 6).

- Developing a tool for the approach

The applicability of the SWfM approach has been shown by developing a tool based upon its steps. The tool is used to construct the SSM conceptual model, which is the connection point between SSM and UML. The tool includes several features that assist the development and validation of the conceptual model. It is possible to enhance these features, as will be suggested below in the recommendations for future work, to provide a practical application of the SWfM approach (Chapter 7).

## 8.3 Review of the Proposed Approach

As stated above, the aim of this research was to develop an approach that models the organisational business process as a workflow system by considering its soft and hard aspects. This approach covers two stages of the workflow system development: investigating the organisational business process and modelling the various perspectives of the workflow system.

In this section the approach that is presented in Chapter Four will be discussed from three different viewpoints: the role of SSM in the approach, the linking of SSM and UML, and the role of the latter in the approach.

### 8.3.1 The Role of Soft Systems Methodology

The development of SSM has been encouraged by the failure of the engineering approach to deal with the human aspects of management problems. Checkland states that SSM is 'a mature and much-tested process of inquiry into problem situations in human affairs' [142]. The main aim is to obtain a rich understanding of the situation before proposing potential improvements [133]. SSM considers every situation as a human situation in which people seek to take purposeful action which is meaningful to them [42].

SSM is used in the SWfM approach to tackle the various soft issues in the organisational business process that is modelled as a workflow system. The SSM stages and techniques handle these soft aspects through the following scenario. First, during the investigation of the problem situation which is the organisational business process, SSM promotes *the users' involvement* to obtain *their viewpoints* and to construct a rich picture which depicts the organisational business process and its relevant issues and aspects. Further, the SSM considers the *users' viewpoints* by developing a root definition (RD) for each point of view. A conceptual model is then developed to depict each point of view and these models are combined into one conceptual model that is compared with the real world. All these stages involve discussion and debate among the people involved to reach a feasible and desirable decision regarding the situation. All these mechanisms and factors support the *acceptance and use* of the resultant decision because all affected people share the responsibility of constructing it [42].

In addition, any organisation consists of two elements: the people and the technical aspects. Thus, SSM has two parts, which are the logical part and the cultural and political part [142]. These interact with each other in the study and investigation of any situation. The logical part includes the development of the RD and conceptual model to deal with the technical part of the situation, while the cultural and political part handles the people aspect of the situation through the use of Analysis One, Two and Three techniques. Each technique focuses on one part of the human aspects, for example the intervention, social and political or cultural parts.

Because of its comprehensive nature, SSM is considered one of the ideal methodologies for tackling the soft issues of the organisational business process.

### 8.3.2 The linking of SSM and UML

It is desirable to link soft and hard approaches to benefit from their advantages in handling the corresponding aspects of the organisational business process. Checkland [142] supports the idea that 'hard systems thinking is appropriate in well-defined technical problems and that soft systems thinking is more appropriate in fuzzy ill-defined situations involving human beings and cultural considerations'.

Although there are underlying philosophical differences between SSM and UML, it is advantageous to combine them to benefit from the best features of each. The link between them is made through the goal concept, which is used to connect the activities of the conceptual model with the use cases of UML.

The use case technique seems a suitable medium for connecting SSM and UML [180]. The conceptual model is a rich structure developed by combining the points of view of the various people concerned with the problem situation. Thus the activity, as an integral part of the model that includes a large amount of information, needs a technique that can retain this information during the transformation or conversion from one methodology to another. This information can be accommodated and defined by using the use case main scenario and the alternatives or extension points to describe the various possibilities for the activity of the conceptual model. The use case technique is considered powerful in specifying the business requirements, as stated by Bustard et al. [143]: 'The types of business analysis offered in current object-oriented analysis techniques are generally beneficial, especially those taking a scenario-based approach, such as use case analysis'. The activities of the conceptual model can be mapped onto use cases after a goal has been defined for each activity in the conceptual model, which is then transferred into a use case that achieves the same goal.

### 8.3.3 The Role of Unified Modelling Language (UML)

UML, as a standard graphical object-oriented modelling language, is used in the SWfM approach to complement the role of SSM in modelling the organisational business process as a workflow system. Being an Object Management Group (OMG) standard and a de facto standard of the object-oriented paradigm, UML can benefit from the advantages and benefits of the object-oriented paradigm to make it the current dominant paradigm for modelling organisational business processes, as discussed in Chapter Four. UML is a complex modelling language that can model any system through its various diagrams and techniques [43]. The use case diagram is used to represent and document the functionality of the business process. Also, the activity diagram depicts several workflow perspectives: the functional, informational, behavioural, organisational and causality perspectives. The sequence diagram shows in detail the interactions of the objects and roles in carrying out the system functionality. Finally, the class diagram represents the structure of the workflow system by showing the various classes with their attributes, operations and relationships that cooperate with each other to accomplish the required behaviour of the workflow system.

In conclusion, the SWfM approach achieves the main aim of the research which is to combine the soft and hard approaches to model the organisational business process that will be implemented through developing a workflow system. SSM is used to analyse, comprehend and model the organisational business process, while UML is used to model the organisational business process as a workflow system. Thus, the organisational business process is able to be implemented by using a workflow system. The two approaches may have different foundations, but can be complementary to each other and applied jointly to model the organisational business process. Bennett et al. [148] state that SSM 'is more useful in the earlier stages of the life cycle, particularly when there is uncertainty about the goals or strategy of the organisation as a whole', while 'a hard approach will be more appropriate once any initial uncertainties and ambiguities have been resolved, since the emphasis then shifts to a specific project with relatively clear goals and boundaries'. This result 'has led to the suggestion that, in certain situations, hard and soft methodologies can complement each other, and can be used together to help to overcome some of the perennial difficulties in systems development'.

## 8.4 Related Work

In this section, the related approaches will be classified into three types: those that deal with both soft and hard issues, those addressing the soft part only and those tackling only the hard aspects. They are reviewed and compared with the SWfM approach. Also, a new version of UML, UML 2, is introduced and the fitness of the approach of this thesis with respect to UML 2 is addressed.

### 8.4.1 Soft and hard approaches

Bustard et al. [143] discuss the possibility of integrating SSM with Shlaer-Mellor Object-Oriented Systems Analysis (OOSA) as a method of object-oriented analysis to define the requirements for software intensive systems as a part of the Requirements Acquisition and Controlled Evolution (RACE) project. They consider the analysis as a two-part phase that includes business and computing analyses. They attempt to integrate SSM as a business analysis technique with the OOSA method as a computing analysis technique. The business analysis is conducted through the use of SSM in four stages: 1) investigating the problem situation to find out the purpose of the business; 2) developing a behaviour model of the business for each perceived purpose; 3) examining in detail each activity of the activity model to construct an interaction model; and 4) finally merging all the separate interaction models into a single interaction model. The computing analysis starts by linking the resulting model, which is the interaction model of the business analysis, with the informational model of OOSA through a five-stage development process: extracting base objects from the interaction model in which entities, stores, inputs and outputs are considered as objects; extracting base relationships by examining interactions among objects; identification by the analyst of additional relationships; defining the relationships between base objects by adding the required relationship names, multiplicity and the required conditions for the relationships; and refining and improving the information model by adding other structures such as generalisation structure. Then, a state model is developed for some objects from the information model that have interesting behaviour. The approach is supported by a tool to build the initial models from the interaction model and maintain consistency between them.

Both this technique and the SWfM approach attempt to combine SSM with the object-oriented paradigm. The RACE technique integrates SSM with the OOSA method, while the SWfM approach combines SSM with UML. The RACE technique uses SSM as a business analysis technique to analyse the problem situation that results from developing the interaction model from the conceptual model. Then a computing analysis considers the interaction model as a start point to integrate with OOSA to develop an object model that will be used as a basis for the consecutive stages of the method. The SWfM approach also encompasses two phases: the organisational business process investigation phase, in which the use of SSM predominates, and the workflow system modelling phase, which uses the conceptual model with the use case technique as a connection mechanism between SSM and UML to model the various perspectives of the workflow system. The RACE technique implements the OOSA method to develop an information system, while the SWfM approach can use any object-oriented approach that is based on UML to design and implement the required workflow system.

- The feature distinguishing the RACE technique from the SWfM approach is that the former is used in defining the requirements of software, while the latter is used to model the organisational business process as a workflow system.

Wade and Hopkins [180] investigated the possibility of producing a framework for integrating some techniques of the SSM with the UML at the requirements elicitation stage. The framework consists of the following steps: first, the problem situation is explored and defined and then depicted as a rich picture. Next, relevant human activity systems are identified and their root definitions are formulated. A conceptual model is subsequently developed for each root definition or human activity system. Then the conceptual model is analysed to identify candidate use cases, which are fully specified by using a pre-defined use case template to show that their elements can be modelled through the various UML diagrams, such as a sequence diagram to present the objects collaborating to realise each use case, or a class diagram to show the classes required for the development of an information system.

Both the Wade and Hopkins framework and the SWfM approach are trying to integrate SSM with UML where the conceptual model is considered as the focal point for linking them. The means of this integration is unclear in Wade and Hopkins's framework, while in the SWfM approach, the transition of the conceptual model activities into use cases is closely defined by analysing the conceptual model activities in detail, identifying their goals and dividing the activities or combining them depending on their achieved goals to construct activities-use-cases to be mapped one-to-one onto UML use cases. These use cases are then depicted by using the different diagrams of the UML (activity diagrams, sequence diagrams, class diagrams and package diagrams) for modelling the various perspectives of a workflow system. Wade and Hopkins's framework is used to develop an information system from scratch without considering the previous or current information system, while the SWfM approach considers both cases, either developing a workflow system from scratch or, if there is a current system, making a comparison with it in step four of the organisational business process investigation stage.

- The aim of the Wade and Hopkins framework is to develop an information system, while the SWfM approach intends to model an organisational business process as a workflow system.

### 8.4.2 Soft approaches

Patel [181] introduces the Soft-System-Conceived Office Model with a Knowledge Representation (SCORE) framework, in which SSM is extended by adding some concepts to address both the soft and hard issues of the office environment. The framework encompasses four components: 1) SSM, which is the core component guiding the inquiry process in the office environment; 2) the expanding modelling constructs that extend the soft systems human activity model to assemble and integrate the office hard elements by introducing the concepts of event, entity, rule and constraint; 3) a knowledge representation language, Telos, which is used to model the knowledge of the office aspects; and 4) a knowledge-based software tool, ScoreBase, that assists the incremental modelling of the office aspects, based on the SSM learning process.

Patel's framework is used to investigate the office environment; it captures the essence of its activity by using the techniques and extended concepts of SSM as tools to represent, formulate and model the various aspects of that environment. By contrast, the SWfM approach analyses the organisational business process, models its activities conceptually and then connects them with the use cases technique of UML. Subsequently, the other UML diagrams are used to model the various perspectives of the workflow system.

- Patel's framework deals only with the office environment, while the SWfM approach handles the big picture of the organisational business process.

### 8.4.3 Hard approaches

Chang et al. [182] propose an approach to model the business process by using UML with a view to implementing it in e-commerce. The approach consists of four stages: process requirement elicitation, logical design, properties analysis and physical deployment. The various UML diagrams are used to model the business process. For instance, the use case diagram represents the context of the business process where the use case is considered as a sub-process of the business process; the class diagram models the information objects, roles and organisational units; the activity diagram depicts the control flow of the business process; and the deployment model shows the physical configuration of activities. The activity diagram is transferred to Petri nets to analyse the business process and verify the properties of its behaviour.

Chang's approach thus considers the hard aspects of the business process, while the SWfM approach handles both soft and hard issues. Both approaches use the mechanism of the various UML diagrams to model the workflow system, but SWfM also uses the sequence diagram to model the various perspectives of the workflow system and the SSM to deal with the soft aspects. In addition, Chang's approach uses the Petri nets technique to analyse and verify the modelling of the business process, where the SWfM approach uses a case study to test the approach and its implementation.

- Chang's approach is intended to model the web-based business process, while the SWfM approach aims to model the organisational business process in general.

Bastos et al. [183] introduce an approach to the modelling of business processes in production systems using workflow concepts. They propose an extension of the UML activity diagram, called Workflow Activity Diagram (WAD), which is used to model the concepts of the C-Wf model introduced by Bastos and Ruiz [184]. This model depicts the structural and functional enterprise objects involved in business processes, such as activities, human, machine and application resources. They also use the UML use cases technique. The C-Wf model involves two main contexts: process design and definition; and process instantiation and control. In process definition, a domain consists of a set of interactive domain processes which are decomposed into business processes and enterprise activities. The latter are elementary tasks which consume inputs to produce outputs and require allocation of time and resources for their execution. Each enterprise activity includes one or more functional operations that are carried out by the functional entities which are the production resources, such as humans, machines and applications. The assignment of enterprise activities to functional entities is done through the capability concept. The interactions between enterprise activities are defined through behavioural rules. In process instantiation, a C-Wf management system (C-WfMS) is responsible for the creation and control of instances of the defined domain processes when it receives one or more events. The workflow model for a company could be specified through the business use case model. The C-Wf classes are modelled in the workflow activity diagram by using stereotypes, UML extension mechanisms to create new constructs for a particular domain or problem, and new properties for this diagram. The C-Wf concepts which are represented in the WAD as stereotypes are <<event>>, <<Human>>, <<Application>> and <<Machine>>. The WAD depicts functional, behavioural, and resource perspectives of the domain or business process.

Both the Bastos and SWfM approaches adopt UML diagrams to model workflow concepts and use the use case model to present the business process; however, the Bastos approach realises the business use case through the WAD, which is an extension of the UML activity diagram, while SWfM utilises the UML activity diagram as well as sequence, class and deployments diagrams, to model the various perspectives of the workflow system. Further, the SWfM approach covers more workflow perspectives, as set out in Table 6-4 of Chapter Six, than the Bastos approach, which represents the functional, behavioural and resource (organisational and operational) perspectives of the workflow system.

- The SWfM approach deals with more aspects of the workflow system than does the Bastos approach.

Other techniques have been used to model (part of) a workflow system. Dong and Shensheng [185] use a Statechart, which has several features, such as OR-hierarchy and AND-orthogonal decomposition, event-based synchronisation and interaction mechanisms for modelling control structures, and priority and history mechanisms for defining the exception handling. Also, it is possible to verify and execute the workflow model by using the Statechart technique, due to its well-defined formal semantics.

Dong and Shensheng's approach handles the hard aspects of the workflow system. It uses one technique of UML, Statechart, to model the workflow system. On the other hand, the SWfM approach is concerned with both soft and hard aspects, and utilises most of the UML diagrams to model the various perspectives of the workflow system.

- The SWfM approach handles soft aspects of the workflow system in addition to the hard ones which Dong and Shensheng's approach handles only using one technique.

Van der Aalst et al. [186] describe an approach to integrate workflows with the organisational model. The model and its parts, such as resources, resource types, collections and various relationships between resources and resource types, are specified by using Extensible Markup Language (XML). The organisational perspective is modelled by using UML to depict an organisation consisting of three elements: resource entities (user and non-user), resource types (role, machine type and space type) and collection (departments).

The approach of van der Aalst et al. is concerned with the hard section of the workflow system. It proposes a model for the organisational perspective which is considered as one of several perspectives that the SWfM approach models. Both van der Aalst and SWfM use a class diagram to model the organisational perspective, but the SWfM approach also uses other UML diagrams to model the various perspectives of the workflow system.

- The SWfM approach is more general than that of van der Aalst, in that it handles most of the workflow system perspectives.

Finally, Ellis et al. [187] propose Information Control Nets (ICNs), derived from Elementary Nets (E nets), to model the office workflow. The E nets are modified by including a data flow model, control flow primitives and simplified semantics to be used in modelling the office procedure. ICNs consist of a set of activities which are depicted as large circles; OR control flow nodes, which are portrayed as small open circles; AND control flow nodes, which are depicted as small filled circles; edges to connect these nodes; and repository nodes, which are portrayed as square nodes to represent persistent data. The workflow modelling concepts include office work, which consists of procedures, activities and actions, a job (a work case), a role and an actor. Further, the conceptual architecture of the workflow system is represented using the entity relationship (E-R) model. This architecture illustrates the relationships among the above workflow concepts, including the data entities. A formal definition of ICN is also presented. Finally, an exception handling mechanism, the SendTo feature, is provided to assist the user in dealing with the various exceptions that occur during the execution of a procedure. The structure of the SendTo feature is as follows:

SendTo (recipients, message-type, activity-state, static routing, receive-time)

Both the ICN and SWfM approaches use a graphical notation to model the workflow system, which helps users and other people involved to understand the model. ICN models four perspectives of the workflow system (functional, informational, behavioural and organisational), while SWfM tries to cover the seven perspectives of the workflow perspectives framework set out in Chapter Three, albeit at different levels of coverage. The ICN approach considers the importance of the social, organisational and technical issues, but does not include any mechanism to handle them; the SWfM approach, in contrast, provides a framework in which SSM is used to handle the social and organisational issues and UML is used to depict the technical issues of the workflow system.

- The SWfM approach is considered more general than ICN in covering the social and organisational issues and the various perspectives of the workflow system.

### 8.4.4 The Impact of UML version 2 (UML 2)

A new version of UML, UML version 2 (UML 2), has recently been introduced by OMG [188]. UML 2 features a number of improvements over UML in general and its modelling usage to cover more domains. It is difficult to cover all the changes, but it is important to consider the essential ones and describe the various new features that are added to the diagrams which are used in this research. UML 2 introduces new diagrams: interaction overview diagrams, timing diagrams and composite structure diagrams. Also, it changes the name of 'collaboration diagrams' to 'communication diagrams'. Finally, it makes the use of object diagrams and package diagrams official, which they were not before. The following points discuss the various diagrams with the new features that are used in the SWfM approach, to show how this approach fits with UML 2.

i. There is no significant change in the use case diagram.

ii. Many changes have been made to the activity diagram so that it is no longer considered as a special case of the state diagram and there is a trend toward the Petri nets technique through the use of the terms 'token production' and 'consumption'. UML 2 calls the nodes in the diagram 'actions', not 'activities', so 'activity' indicates a sequence of actions and the diagram depicts an activity that is composed of actions. Further, it is possible to decompose the action into subactivities, depict these in a subsidiary activity diagram and call them actions in the usual way. Also, action implementations can be shown as subactivities, class methods or code fragments in the action node. The connection between actions, which is called 'flow' or 'edge', can be presented in several ways: as a simple arrow between two actions with or without a name; as a connector with a label, to simplify the drawing of the flow; or the transferring of data or objects can be depicted by a class box or pins, which are small squares on the action nodes labelled with the object names, between actions. The pins feature can be used to model the resources produced and consumed by the actions of a business process. In addition, the conditional behaviour has been altered to include decisions and 'merges', where the decision that specifies the start has one input flow and several guarded output flows, while the merge has multiple input flows and one output flow that labels the end of the conditional behaviour. UML 2 has substituted an implicit merge of multiple incoming flows for an action by an implicit join, so that the action will be executed only if all flows trigger, and has removed the balancing rules for the forks and joins. The activity diagram can model two types of signals: the time signal, which occurs depending on the passage of time that initiates an event to activate an action in the diagram, and the send and accept signals that are used to model sending messages and receiving replies. An expansion region mechanism is added to the diagram to model a situation in which one action output triggers multiple invocations of another action. Finally, the diagram swimlanes (partitions) mechanism has been modified to be a two-dimensional grid; it is possible to divide each partition into rows and columns hierarchically, with its name replaced by partitions [149]; and UML 2 also shows the person responsible for an activity or an action by placing his name in parentheses inside the activity or action notation above the name of the activity or action [189].

The aforementioned features improve the activity diagram and make it better able to represent the various elements of the business process. These include: decomposing an action and including its implementation in the action node; using a two-dimensional grid for the partition and dividing each one into rows and columns hierarchically; adding time, send and accept signals, and an extension region mechanism; and the use of pins as symbols to model resources.

iii. In UML 2, several aspects of the sequence diagram have been altered. The objects used in the diagram have been changed to object roles, but Fowler [149] calls them 'participants' that can be named in the form of *name: Class*, where both the name and the class are optional. A message that comes from an outside source is called a *found message*. Although the sequence diagrams are not suitable for modelling looping or conditional behaviour, UML 2 introduces interaction frames, which are ways of enclosing a piece of a sequence diagram to portray this. Each frame includes an operator for the loop or condition and each fragment in the frame may have a guard to define the loop or the condition. Finally, the arrowheads of the messages have been changed so that filled arrowheads represent a synchronous message, while stick arrowheads depict an asynchronous message.

UML 2 introduces a new diagram, the interaction overview diagram, that combines the features of both the sequence diagram and the activity diagram. It could be an activity diagram in which the activities are replaced by little sequence diagrams or a sequence diagram that is broken up with activity diagram notation used to represent control flow [149]. This diagram could be useful in modelling the workflow system, but the problem is that the diagram will be complex and hard to comprehend and follow.

iv. Several minor changes have been made to the class diagram in UML 2. Discontinuous multiplicities such as *[3, 5]* and the frozen property, which cannot change during the lifetime of an object, have been dropped. The notations of attributes, unidirectional associations and active class have been altered so that attributes can be modelled as associations, and double vertical lines are added to the active class instead of thick lines. UML 2 redefines the stereotypes. In addition, classes, as well as providing interfaces, can now be made to require interfaces. Finally, multiple classifications can be grouped by using a generalisation set that is labelled with its name [149]. The most important change is the stereotype mechanism, but there is no further detail about it in UML 2 because OMG has not yet revealed the official documentation of UML 2.

Although the changes from UML to UML 2 are quite significant, the method used in this thesis to link SSM and UML is unaffected, suggesting that the connection between the conceptual model of SSM and the use case of the UML has been done at an appropriate conceptual level.

## 8.5 The Limitations of the SWfM Approach

The development of a workflow system involves several phases, extending from defining the requirements of the workflow system, modelling and analysis, through design and implementation to finally verifying the workflow system and putting it into operation [106]. The SWfM approach deals with only the two most important and fundamental phases: the organisational business process investigation and the workflow system modelling. The other phases are based on these two; thus it is important to cover them in a suitable and efficient way.

Furthermore, it is important to evaluate the SWfM approach using several case studies in different areas. However, time limitations, the difficulty of communicating with actual users to collect information and the surrounding environment of the research limited the implementation of the approach to a single case study. Although this cannot be considered a sufficient basis to claim that the SWfM approach is generalisable to other situations, it does show the possibility of implementing the approach in the real world and achieves the required objective. It is recommended that further implementations of the SWfM approach should be conducted in industry to test its capabilities and strengthen it.

Finally, the workflow reference model in Chapter Three describes eleven perspectives of the workflow system. These perspectives cover most of the aspects in both the conceptual and implementation levels of the workflow system. The SWfM approach handles only the essential perspectives of the workflow reference model in addition to a soft perspective, which are related to the modelling phase of the workflow system to cover the soft and hard issues of the organisational business process. The other perspectives can be covered when the workflow system is implemented in the real world, as proposed in the following section.

## 8.6 Future work

The SWfM approach is a conceptual framework based on developing conceptual models for presenting organisational business processes and their requirements. These models are semi-formal techniques which may have different interpretations depending on the background of their users. Therefore, it is sometimes important to use formal techniques to provide precision and consistency in the system requirements, to detect errors and to facilitate execution by the supported tools. As a result, it would be possible to improve the formality of the approach by combining it with some formal techniques such as Z notations [190-192], process algebra [193, 194], formal specification language B [195-197], or the Petri nets technique [198-201]. There have been several attempts to combine SSM with formal techniques; for example, Bustard and Lundy [202] combine SSM with formal specification in the Language of Temporal Ordering Specification (LOTOS) as a part of the RACE project. Krishnan [203] tries to provide formality for UML dynamic diagrams by translating them to the theorem prover Prototype Verification System (PVS) to validate consistency between the diagrams. Yen-Liang Chang et al. [182] use UML diagrams to model the business process and then transform them into Petri nets to verify the integrity and accuracy of the business process definition by using the National Central University Petri Nets (NCUPN) toolkit, and Eshuis and Wieringa [204] introduce a tool to verify workflow models that are developed through a UML activity graph by using model checking, which is done by translating the UML activity diagram firstly into a formal activity diagram then into a transition system that is validated against the temporal logic formula of the formal requirements to identify any error or omission in the model. The SWfM approach can be formalised by using the Petri nets technique, for several reasons: Petri nets has formal semantics with clear and precise definitions; it is expressive and supports all the required workflow modelling primitives; there are several verification tools in which Petri nets are used to validate the workflow model; and it is vendor-independent, i.e. not based on a software package from a specific vendor [205], [206].

In addition, the workflow system is a complex one. The SWfM approach models the main perspectives of the workflow reference model, involving the functional, informational, behavioural, operational and organisational perspectives and some soft perspectives, such as causality and quality. It would be possible to extend the SWfM approach by incorporating some of the other related perspectives of the workflow reference model such as the security perspective, which is concerned with who has the right to access an object by defining suitable security and organisational policies, and the history perspective, which deals with retaining all the information about the workflow execution to use it in several contexts including workflow maintenance and analysis. These perspectives address the implementation issues of the workflow system beyond the workflow specification.

Further, with the increased use of e-business, the SWfM approach could be extended to model cross-organisational processes, which have several challenging aspects that should be addressed and modelled; for example, distributed process definitions, contracts among organisations, data transferring, making decisions and their effects on the other organisations and security arrangements. All these issues should be considered for implementation by using the workflow technology through Electronic Data Interchange (EDI) or web technology.

Finally, the tool which has been developed as a part of the SWfM approach to help in developing a SSM conceptual model would be enhanced by adding several functions to it. It would be possible to develop a connection between the activities of the conceptual model and the activity table description to show their details. It would also be desirable to add a GUI interface to transfer the SSM conceptual model activities into UML use cases and model them in a UML use case diagram, which could be used in the subsequent stages for developing a workflow system.

## 8.7 Conclusion

A new approach to workflow modelling, SWfM, was developed, considering more perspectives of the workflow system by involving soft, causality and quality perspectives in addition to the traditional ones, which are the functional, informational, behavioural, operational and organisational perspectives. Both SSM as a framework methodology and UML were used to depict these perspectives. Further, the SWfM approach was applied to an illustrative case study to test its applicability for implementation. It was then compared with other well-known workflow models to assess its capabilities and to discover its deficiencies, in order to eliminate them and improve the approach in the future. The workflow modelling produces several diagrams which must be maintained and managed. Thus, a simple tool was developed to support the SWfM approach in constructing and managing these diagrams. It would be good to improve the tool in the future to cover all the stages and steps of the approach.

This research has comprehensively addressed the research question. It has dealt with investigating the organisational business process by considering both its soft and hard aspects, represented as workflow perspectives which constitute the workflow perspectives framework.

Although this research has to end at this point, the researcher will pursue his career in the field of information systems and modelling through teaching and further research in this field in his home country, Saudi Arabia.

## References


[1] P. Feiler and W. Humphrey, "Software Process Development and Enactment: concepts and definitions," presented at Proceedings of the 2nd International Conference on the Software Process, Berlin, Germany, 1993.

[2] M. Hammer, *The Reengineering Revolution*: Harper Business, 1995.

[3] D. Platt, "Process Modelling and Process Support Environments for Design Management," in *Faculty of Engineering, Department of Civil Engineering*: University of Bristol, 1994.

[4] T. Davenport, *Process Innovation- reengineering work through information technology*: Harvard Business School Press, 1993.

[5] J. Lonchamp, "A Structured Conceptual and Terminological Framework for Software Process Engineering," presented at Proceedings of the 2nd International Conference on the Software Process, Berlin, Germany, 1993.

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

[7] W. van der Aalst, "Loosely Coupled Interorganizational Workflows: modeling and analyzing workflows crossing organizational boundaries," *Information and Management*, vol. 37, pp. 67-75, 2000.

[8] D. Chaffey, *E-Business and E-commerce Management: strategy, implementation and practice*: Pearson Education Limited, 2002.

[9] H. Ludwig and K. Whittingham, "Virtual Enterprise Co-ordinator-Agreement-Driven Gateways for Cross-Organisational Workflow Management," presented at the International Joint Conference on Work Activities coordination and Collaboration, WACC'99, San Francisco, California, USA, 1999.

[10] U. Dayal, M. Hsu, and R. Ladin, "Business Process Coordination: State of the Art, Trends, and Open Issues," presented at the 27th Very Large Databases Conference (VLDB 2001). Roma, Italy, 2001.

[11] N. Melao and M. Pidd, "A Conceptual Framework for Understanding Business Processes and Business Process Modelling," *Information systems Journal*, vol. 10, pp. 105-129, 2000.

[12] A. Cichocki, A. Helal, M. Rusinkiewics, and D. Woelk, *Workflow and Process Automation Concepts and Technology*: Kluwer Academic Publisher Group, 1998.

[13] R. Endl and M. Meyer, "Potential of Business Process Modeling with regard to available Workflow Management Systems," Report 20, February 1999.

[14] R. S. Aguilar-Saven, "Business process modelling: Review and framework," *International Journal of Production Economics*, vol. 90, pp. 129-149, 2004.

[15] P. Mangan and S. Sadiq, "On building workflow models for flexible processes," presented at thirteenth Australasian conference on Database technologies, Melbourne, Victoria, Australia, 2002.

[16] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of Workflow Management: From Process modeling to Workflow Automation Infrastructure," *Distributed and Parallel Databases*, vol. 3, pp. 119-153, 1995.

[17] W. M. Coalition, "Workflow Management Coalition Terminology & Glossary," vol. 2004. Winchester, 1999.

[18] C. Prior, "Workflow and Process Management," in *The Workflow HandBook 2003*, L. Fischer, Ed. Lighthouse Point, FL., USA: Future Strategies Inc., 2003, pp. 17-25.
[19] G. Florijn, "Workflow Management - A Limited View on Office Processes," presented at CSCW'94 Workshop on Workflow Systems and Office Information Systems, 1994.
[20] S. Deng, Z. Yu, Z. Wu, and L. Huang, "Enhancement of Workflow Flexibility by Composing Activities at Run-time," presented at 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, 2004.
[21] P. Dourish, J. Holmes, A. MacLean, P. Marqvardsen, and A. Zbyslaw, "Freeflow: mediating between representation and action in workflow systems," presented at 1996 ACM conference on Computer supported cooperative work, Boston, Massachusetts, United States, 1996.
[22] S. Choennia, R. Bakkera, and W. Baetsa, "On the Evaluation of Workflow Systems in Business Processes," *Electronic Journal of Information Systems Evaluation*, vol. 6, pp. 33-44, 2003.
[23] D. Hollingsworth, "The Workflow Reference Model," 2003, 1.1 ed: Workflow Management Coalition, 1995.
[24] W. Sadiq and M. E. Orlowska, "On Correctness Issues in Conceptual Modeling of Workflows," presented at 5th European Conference on Information Systems (ECIS '97), Cork, Ireland, 1997.
[25] J. Yang and M. P. Papazoglou, "Interoperation Support for Electronic Business," *Communications of the ACM*, vol. 43, pp. 39-47, 2000.
[26] F. Casati and M.-C. Shan, "Process Automation as the Foundation for E-Business," presented at 26th International Conference on Very Large Databases, Cairo, Egypt, 2000.
[27] S. Teufel and B. Teufel, "Bridging Information Technology and Business: Some Modelling Aspects," *ACM SIGOIS Bulletin*, vol. 16, pp. 13-17, 1995.
[28] Y. Lei and M. P. Singh, "A Comparison of Workflow Metamodels," presented at ER'97 Workshop on Behaviour Modeling, 1997.
[29] W. M. Coalition, "Glossary - A Workflow Management Coalition Specification," 1996.
[30] W. M. Coalition, "Interface 1: Process Definition Interchange- Process Model," vol. 2005: Workflow Management Coalition, 1998, WfMC TC-1016-P.
[31] B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, and A. K. Elmagarmid, "Business-to-Business Interactions: Issues and Enabling Technologies," *The VLDB Journal — The International Journal on Very Large Data Bases*, vol. 12, pp. 59-85, 2003.
[32] JointFlow, "Worflow Management Facility, Revised submission," OMG Document Number: bom/98-06-07, 1998.
[33] F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan, "eFlow: a platform for Developing and Managing Composite e-Services," presented at Proceedings of Academia/Industry Working Conference on Research Challenges (AIWORC'2000), Buffalo, NY, 2000.
[34] A. Lazcano, G. Alonso, H. Schuldt, and C. Schuler, "The WISE approach to Electronic Commerce," *International Journal of Computer Systems Science and Engineering*, vol. 15, pp. 345-357, 2000.

[35] D. K. Punia and K. B. C. Saxena, "Managing Inter-organisational Workflows in eGovernment Services," presented at the 6th international conference on Electronic commerce, Delft, The Netherlands, 2004.

[36] R. Tagg, "Workflow in Different Styles of Virtual Enterprise," presented at Workshop on Information Technology for Virtual Enterprises, Queensland, Australia, 2001.

[37] J. Klingemann, J. Wasch, and K. Aberer, "Adaptive Outsourcing in Cross-Organizational Workflows," presented at 11th International Conference CAiSE'99, Heidelberg, Germany, 1999.

[38] S. Jablonski and C. Bussler, *Workflow Management: Modelling Concepts, Architecture and Implementation*, 1st ed: International Thomson Computer Press, 1996.

[39] R. Medina-Mora, T. Winograd, R. Flores, and F. Flores, "The Action Workflow Approach to Workflow Management Technology," presented at the 1992 ACM conference on Computer-supported cooperative work, Toronto, Ontario, Canada, 1992.

[40] G. Mentzas, C. Halaris, and S. Kavadias, "Workflow Modelling in Project Management Processes: Applications to the EU Operational Program for the Energy Sector," presented at 5th International Conference of the Decision Sciences Institute, Athens, Greece, 1999.

[41] P. Checkland, *Systems Thinking, Systems Practice*. Chichester: Wiley, 1981.

[42] P. Checkland and J. Scholes, *Soft Systems Methodology in Action*: John Wiley & Sons Ltd., 1990.

[43] D. Avison and G. Fitzgerald, *Information Systems Development Methodologies, Techniques and Tools*, 3rd ed: McGraw-Hill Education (UK) Limited, 2003.

[44] D. Flynn, *Information systems requirement: determination & analysis*, 2 ed: McGraw-Hill Publishing Company, 1998.

[45] U. P. o. Cambridge, "Cambridge Advanced Learner's Dictionary-Cambridge Dictionaries Online," Cambridge University Press, 2003.

[46] R. N. Maddison, *Information System Methodologies*. Chichester, UK: Wiley Heyden, 1983.

[47] P. Beynon-Davies, *Information Systems Development*, 3 ed: Macmillan Press Ltd., 1998.

[48] T. Davenport and D. Stoddard, "Reengineering: Business Change of Mythic Proportions?," *MIS Quarterly*, vol. 18, pp. 121-127, 1994.

[49] T. H. Davenport and J. E. Short, "The New Industrial Engineering: Information Technology and Business Process Redesign," *Sloan Management Review*, vol. 31, pp. 11-27, 1990.

[50] M. Hammer and J. Champy, *Reengineering the Corporation*. New Jersey, New York: Harper Collins Books, 1993.

[51] D. McCauley, "What is Business Process Management? And How Can It Make a Difference to Your Business?,", 2005.

[52] I. Jacobson, M. Ericsson, and A. Jacobson, *The Object Advantage: Business Process Reengineering With Object Technology*: Addison-Wesley, 1995.

[53] M. Ould, *Business Processes: Modelling and Analysis for Re-engineering and Improvement*. Chichester: John Wiley & Sons, 1995.

[54] A. M. Ould, *Business Processes: Modelling and Analysis for Re-Engineering and Improvement*: Wiley, 1995.

[55] P. Loucopoulos, "The S3 (Strategy-Service-Support) Framework for Business Process Modelling," presented at CAiSE Workshops 2003, 2003.

[56] B. Curtis, M. I. Kellner, and J. Over, "Process Modeling," *Communication of the ACM*, vol. 35, pp. 75-90, 1992.

[57] A. Lindsay, D. Downs, and K. Lunn, "Business Processes- attempts to find a definition," *Information and Software Technology*, vol. 45, pp. 1015-1019, 2003.

[58] W. J. Kettinger, J. Teng, and S. Guja, "Business Process Change: A study of methodologies, techniques and tools," *Journal of Management Information Systems*, vol. 14, pp. 119-154, 1997.

[59] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business Process Management: A Survey," presented at Business Process Management: On the Application of Formal Methods to "Process-Aware" Information Systems, Eindhoven, The Netherlands, 2003.

[60] B. Scholz-Reiter and E. Stickel, *Business Process Modelling*. Berlin: Springer-Verlag, 1996.

[61] N. Patel and V. Hlupic, "Dynamic Business Process Modelling (BPM) for Business Process Change," *International Journal of Simulation Systems Science and Technology*, vol. 2, pp. 65-76, 2001.

[62] B. Hommes and V. Reijswoud, "Assessing the Quality of Business Process Modelling Techniques - introduction of the Q-ME framework and example," presented at 33rd Hawaii International Conference on System Sciences, Hawaii, USA, 2000.

[63] G. M. Giaglis, R. J. Paul, and G. I. Doukidis, "Simulation for Intra- and Inter-Organisational Business Process Modelling," presented at WSC '96, Proceedings of 1996 Winter Simulation Conference, Coronado, CA, USA., 1996.

[64] G. M. Giaglis, "A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques," *International Journal of Flexible Manufacturing Systems*, vol. 13, pp. 209-228, 2001.

[65] K. Kim and S. Paik, "Practical Experiences and Requirements on Workflow," *Lecture Notes in Computer Science*, vol. 1364, pp. 145-160, 1998.

[66] G. Alonso, D. Agrawal, A. El Abbadi, and C. Mohan, "Functionality and Limitations of Current Workflow Management Systems," Research Report, IBM Almaden Research Center 1997.

[67] Y. Yan, Z. Maamar, and W. Shen, "Integration of Workflow and Agent Technology for Business Process Management," presented at The Sixth International Conference on CSCW in Design, London, Ontario, Canada, 2001.

[68] Yen-Liang Chang, Sammy Chen, and C.-C. Chen, "Workflow Process Definition and Their Applications in e-Commerce," presented at 2000 International Symposium on Multimedia Software Engineering (ISMSE 2000), Taipei, Taiwan, 2000.

[69] C. Prior, "Workflow and Process Management," in *Workflow Handbook*, W. M. Coalition, Ed. Lighthouse Point FL, USA: Future Strategies Inc., 2003, pp. 17-26.

[70] D. Chiu, Q. Li, and K. Karlapalem, "A Meta Modeling Approach to Workflow Management Systems Supporting exception Handling," *Information Systems*, vol. 24, pp. 159-184, 1999.

[71] J. Veijalainen, A. Lehtola, and O. Pihlajamaa, "Research Issues in Workflow Systems," Finland October 2 1995.

[72] V. Patil and A. Chaudhari, "Cross Organizational Workflow Management Systems," presented at Product Data Technology Europe 2002, Centro Ricerche Fiat, Turin, Italy, 2002.

[73] W. M. Coalition, "Workflow Interoperability- abstract specification," vol. 2004, version 2.0b (draft) ed., 1999.

[74] W. M. Coalition, "Interface 1: Process Definition Interchange- Process Model," 2005: Workflow Management Coalition, 1999, pp. WfMC TC-1016-P.

[75] M. Zur Muehlen and J. Becker, "Workflow Process Definition Language-State-of-art-Art and Development and Directions of a Meta-Language for Workflow Process," presented at the 1st KnowTech Forum, Potsdam, 1999.

[76] W. M. P. van der Aalst, M. Weske, and G. Wirtz, "Advanced Topics in Workflow Management: Issues, Requirements, and Solutions," *Society for Design and Process Science*, vol. 7, pp. 49-77, 2003.

[77] J. Becker, M. Rosemann, and C. von Uthmann, "Guidelines of Buisness Process Modeling," *Lecture Notes In Computer Science-Business Process Management, Models, Techniques, and Empirical Studies*, vol. 1806, pp. 30-49, 2000.

[78] M. Bernauer, G. Kappel, G. Kramler, and W. Retschitzegger, "Specification of Interorganizational Workflows- A Comparison Approaches," presented at the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI), Orlando, USA, 2003.

[79] C. J. Bussler, "Policy Resolution in Workflow Management Systems," *Digital Technical Journal*, vol. 6, pp. 26-49, 1994.

[80] C. Combi, "Architectures for a Temporal Workflow Management System," presented at the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004.

[81] J. L. Zhao, "Knowledge Management and Organizaional Learning in Workflow Systems," presented at the AIS Americas Conference on Information Systems, Baltimore, MD, USA, 1998.

[82] F. Leymann and D. Roller, "Business Process Management With FlowMark," presented at COMPCON 1994, San Francisco, California, USA, 1994.

[83] I. FlowMark, "Programming Guide, Version 2.1," Interactional Business Machines Corporation, Vienna, Austria 1995.

[84] F. Casati, P. Grefen, B. Pernici, G. Pozzi, and G. Sanchez, "WIDE Workflow Model and Architecture," University of Twente 96-19, 1996.

[85] Software-Ley-GmbH, "COSA Reference Manual," Pulheim, Germany 1994.

[86] Software-Ley-GmbH, "COSA Administration Manual," Pulheim, Germany 1994.

[87] S. Sarin, K. Abbot, and D. McCarthy, "A Process Model and System for Supporting Collaborative Work," presented at Proceedings of the ACM SIGOIS Conference on Organisational Computing Systems, Atlanta, GA, 1991.

[88] D. McCarthy and S. Sarin, "Workflow and Transactions in InConcert," *Bulletin of the Technical Committee on Data Engineering*, vol. 16, pp. 53-56, 1993.

[89] WorkParty, "Technical Information," Siemens Nixdorf Informations Systeme AG Version 1.1, 1994.

[90] WorkParty, "Organiser Tool," Siemens Nixdorf Informations Systeme AG Version 1.1, 1994.

[91] P. Lawrence, "Workflow Handbook 1997." Chichester: John Wiley & Sons, 1997.

[92] Action-Technologies-Inc., "White Paper," Alameda, CA, USA 1993.

[93] T. Kreifelts, E. Hinrichs, K. Klein, P. Seuffert, and G. Woetzel, "Experiences with Domino Office Procedure System," presented at Proceedings of the 2nd European Conference on Computer-Supported Co-operative Work (ECSCW '91), Amsterdam, Netherlands, 1991.

[94] T. Kreifelts, E. Hinrichs, and G. Woetzel, "Sharing To-Do lists with a distributed Task Manager," presented at Proceedings of the 3rd European Conference on Computer-Supported Co-operative Work (ECSCW '93), Milan, Italy, 1993.

[95] G. Woetzel and T. Kreifelts, "The Use of Petri Nets for Modelling Workflow in the Domino System," presented at Workshop on CSCW, Petri Nets and Related Formalism, International Conference on Application and Theory of Petri Nets, Chicago, IL, 1993.

[96] C. Ellis and M. Bernal, "OfficeTalk-D: An Experimental Office Information System," presented at 1st Proceedings SIGOA Conference on Office Automation Systems, 1982.

[97] C. Ellis and G. Nutt, "Office Information Systems and Computer Science," *ACM Computing Surveys*, vol. 12, pp. 27-60, 1980.

[98] C. Ellis, "Formal and Informal Models of Office Activity," in *Information Processing 83*, R. Mason, Ed. North-Holland, Amsterdam., 1983.

[99] C. Ellis and J. Wainer, "Goal Based Model of Groupware," University of Colorado at Boulder, Department of Computer Science. Technical Report CU-CS-668-93, 1993.

[100] W. M. P. van der Aalst, "The Application of Petri Nets to Workflow Management," *Journal of Circuits, Systems, and Computers*, vol. 8, pp. 21-66, 1998.

[101] R. Eshuis and R. Wieringa, "Comparing Petri Net and Activity Diagram Variants for Workflow Modelling - A Quest for Reactive Petri Nets," *Lecture Notes in Computer Science-Petri Net Technology for Communication-Based Systems: Advances in Petri Nets*, pp. 321-351, 2003.

[102] R. Eshuis and R. Wieringa, "A real-time execution semantics for UML activity diagrams," *Proc. Fundamental Approaches to Software Engineering (FASE 2001), Lecture Notes in Computer Science*, vol. 2029, pp. 76-90, 2001.

[103] M. Weske, "Workflow Management Systems: Formal Foundation, Conceptual Design, Implementation Aspects," University of Munster, 2000.

[104] F. Casati, S. Ceri, B. Pernici, and G. Pozzi, "Workflow Evolution," *Data & Knowledge Engineering*, vol. 24, pp. 211-238, 1998.

[105] P. Grefen, B. Pernici, and G. Sanchez, *Database Support for Workflow Management: The WIDE Project*: Kluwer Academic Publishers, 1999.

[106] W. M. P. van der Aalst and K. van Hee, *Workflow Management: Models, Methods, and Systems*: The MIT Press, 2002.

[107] I. TRANSFLOW AG, "COSA 4- Product Description," 2005, 2003.

[108] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Workflow Data Patterns," Centre for Information Technology Innovation, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia FIT-TR-2004-01, 2004.

[109] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, "Workflow Resource Patterns," presented at Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, 2005.
[110] M. Adams, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Facilitating Flexibility and Dynamic Exception Handling in Workflows through Worklets," presented at The 17th Conference on Advanced Information Systems Engineering Forum (CAiSE05 Forum), (to appear) June 2005,, Porto, Portugal, 2005.
[111] B. Stegmaier, M. Ebbers, and T. Begovac, "Redbook- Image and Workflow Library: FlowMark V2.3 Design Guidelines," 2005, 1998.
[112] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. B. Kiepuszewski, and A. P. A.P. Barros, "workflow patterns," *Distributed and Parallel Databases*, vol. 14, pp. 5-51, 2003.
[113] S. Nurcan and C. Rolland, "A meta-model for goal-driven cooperative work processes," presented at the Workshop on the many facets of Process Engineering (MFPE'97), Gammarth, Tunis, 1997.
[114] K. R. Abbott and S. K. Sarin, "Experiences with Workflow Management: Issues for the Next Generation," presented at the 1994 ACM conference on computer supported cooperative work, Chapel Hill, NC., 1994.
[115] Staffware-Plc., "Staffware Process Suite Version 2- White Paper," vol. 2005: Staffware Plc, 2003.
[116] M. Georgeff and J. Pyke, "White paper: Dynamic Process Orchestration - Leading The Way In Business Agility," 2005, 2003.
[117] G. Norris, I. Wright, J. Dunleavy, J. R. Hurley, and A. Gibson, *SAP : an executive's comprehensive guide*. New York; Chichester: John Wiley & Sons Inc, 1998.
[118] N. H. Bancroft, H. Seip, and A. Sprengel, *Implementing SAP R/3- How to introduce a large system into a large organization*. Greenwich, CT: Manning Publications Co., 1998.
[119] M. Rosemann and M. zur Mühlen, "Evaluation of Workflow Management Systems - A Meta Model Approach," presented at the 2nd CAiSE/IFIP 8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '97), Barcelona, 1997.
[120] J. Ma, H. M. Kienle, P. Kaminski, A. Weber, and M. Litoiu, "Customizing Lotus Notes to Build Software Engineering Tools," presented at the 2003 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, 2003.
[121] C. Ellis and M. Bernal, "OfficeTalk-D: An Experimental Office Information System," presented at Proceedings of the SIGOA conference on Office information systems, Philadelphia, Pennsylvania, United States, 1982.
[122] D. Chan, J. Vonk, G. Sanchez, P. Grefen, and P. Apers, "A specification language for the WIDE workflow model," presented at ACM symposium on Applied Computing., Atlanta, Georgia, United States, 1998.
[123] S. Ceri, P. Grefen, and G. Sanchez, "WIDE-a distributed architecture for workflow management," presented at the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications, Birmingham, UK, 1997.
[124] L. Baresi, F. Casati, S. Castano, M. G. Fugini, I. Mirbel, and B. Pernici, "WIDE Workflow Development Methodology," presented at the international

joint conference on Work Activities Coordination and Collaboration (WACC), San Francisco, California, USA, 1999.
[125] T. J. Dobbin and D. W. Bustard, "Combining Soft Systems Methodology and Object-Oriented Analysis: the Search for a good fit," presented at the 2nd Information Systems Methodologies Conference, Edinburgh, 1994.
[126] P. Checkland and R. Griffin, "Management Information Systems: a system view," *Systems Engineering*, vol. 1, pp. 29-42, 1970.
[127] B. Wilson, *Systems: concepts, methodologies, and applications*, 1st ed. Chichester: Wiley, 1984.
[128] B. Wilson, *Systems: concepts, methodologies, and applications*, 2nd ed. Chichester: Wiley, 1990.
[129] R. Galliers, "An Approach to Information Needs Analysis," presented at IFIP WG 6.3 Conference, Amsterdam, North Holland, 1984.
[130] A. Le Fever and E. Pattison, "Planning for Hospital IS using the Lancaster SSM," *Australian Computer*, vol. 18, pp. 180-185, 1986.
[131] P. Checkland, "Soft Systems Methodology and its Relevance to the Development of Information Systems," in *Information systems provision, the contribution of soft systems methodology*, F. Stowell, Ed.: McGraw-Hill, 1995, pp. 1-17.
[132] F. Stowell, "Experience with Soft Systems Methodology and Data Analysis," in *Information Technology Training*, 1985, pp. 48-50.
[133] J. Mingers, "Using Soft Systems Methodology in the Design of Information Systems," in *Information systems provision, the contribution of soft systems methodology*, F. Stowell, Ed.: McGraw-Hill, 1995, pp. 18-50.
[134] R. Prior, "Deriving Data Flow Diagrams from "Soft Systems" Conceptual Model," *Systemist*, vol. 12, pp. 65-75, 1990.
[135] K. Sawyer, "Linking SSM to DFDs: the two epistemological differences," *Systemist*, vol. 13, pp. 76-80, 1991.
[136] R. Miles, "Combining "Hard" and "Soft" Systems Practice: grafting or embedding," *Applied Systems Analysis*, vol. 15, pp. 55-60, 1988.
[137] R. Miles, "Combining "hard" and "soft" systems practice: grafting or embedding revisited," *Systemist*, vol. Vol. 14, pp. 62-66, 1992.
[138] A. Savage and J. Mingers, "A Framework for Linking Soft Systems Methodology and Jackson Systems Development," Warwick Business School Research Bureau, Coventry, UK, Research Paper 80, 1993.
[139] D. Avison and A. T. Wood-Harper, *Multiview: an exploration in information systems development*. Oxford: Blackwell, 1990.
[140] D. Avison and A. T. Wood-Harper, "Reflections from the Experience of using Multiview: through the lens of soft systems methodology," *Systemist*, vol. 14, pp. 136-145, 1992.
[141] C. Atkinson, "The 'Soft Information Systems and Technologies Methodology' (SISTeM): an actor network contingency approach to integrated development," *European Journal of Information Systems*, pp. 104-123, 2000.
[142] P. Checkland, *Soft Systems Methodology: a 30-year retrospective*: John Wisley & Sons, LTD., 1999.
[143] D. W. Bustard, T. J. Dobbin, and B. N. Carey, "Integrating Soft Systems and Object-Oriented Analysis," presented at IEEE International Conference on Requirements Engineering, Colorado Springs, US, 1996.
[144] S. Kujala, "User involvement: a review of the benefits and challenges," *Behaviour & Information Technology*, vol. 22, pp. 1-16, 2003.

[145] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modelling Language, Reference Manual*: Addison Wesley Longman, Inc., 1999.
[146] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modelling Language User Guide*: Addison Wesley Longman Ltd., 1999.
[147] I. Graham, *Object-Oriented Methods, Principles & Practice*, 3 ed.: Pearson Education Limited, 2001.
[148] S. Bennett, S. McRobb, and R. Farmer, *Object-Oriented Systems Analysis and Design using UML*, 2 ed. Berkshire: McGraw-Hill Education, 2002.
[149] M. Fowler, *UML Distilled: a brief guide to the standard object modelling language*, 3rd ed.: Pearson Education Inc., 2004.
[150] C. C. a. T. Agency, "Applying Soft Systems Methodology to an SSADM Feasibility Study," London: HMSO, London 1993.
[151] D. W. Bustard, Z. He, and F. G. Wilkie, "Soft Systems and Use-case Modelling: mutually supportive or mutually exclusive?," presented at Proceeding of the 32nd Hawaii International Conference of System Science, Hawaii, 1999.
[152] A. Monk and S. Howard, "Methods & tools: the rich picture: a tool for reasoning about work context," *Interactions*, vol. 5, pp. 21-30, 1998.
[153] B. Wilson, *Soft Systems Methodology, Conceptual Model Building and its Contribution*: John Wiley & Sons Ltd., 2001.
[154] A. W. Scheer, *ARIS : business process modeling*, 2nd ed.: Springer, 1999.
[155] C. A. L. s. Dictionary: Cambridge University Press, 2003.
[156] D. Rosenberg and K. Scott, *Use Case Driven Object Modelling with UML: a practical approach*: Addison Wesley Longman, Inc., 1999.
[157] D. Hay, A. Kolber, and K. A. Healy, "Guide Business Rule Project: final report," 1995.
[158] H.-E. Eriksson and M. Penker, *Business Modelling with UML: business patterns at work*: John Wiley & Sons, Inc., 2000.
[159] U. o. N. U. Tyne, "NUInfo - University of Newcastle upon Tyne," vol. 2001. Newcastle Upon Tyne, 2001.
[160] U. o. N. U. Tyne, "Internal Website, Computing Science, University of Newcastle," , 2001.
[161] P. Mair, *The Strategic Procurement of CASE Tools*. Oxford, England: NCC Blackwell Ltd., 1993.
[162] AppWorx, "Workflow and Application Automation Tools: Choosing the Right Tool for the Job,", 2005, 2004.
[163] W. van der Aalst and K. van Hee, *Workflow Management: Models, Methods and Systems*: The MIT Press, 2004.
[164] E. Knottingham, *How to Do Everything with Microsoft Visio 2002*: Osborne/McGraw-Hill, 2001.
[165] I. Vessey, S. Jarvenpaa, and N. Tractinsky, "Evaluation of Vendor Products: CASE tools as methodology companions," *Communications of the ACM*, vol. 35, pp. 90-105, 1992.
[166] I. Sommerville, *Software Engineering*, 5th ed.: Addison-Wesley Publishing Company Inc., 1996.
[167] S. Holloway and T. Bidgood, *CASE Strategies Guide for Information Managers*: Ashgate Publishing Limited, 1993.
[168] R. Bussow, W. Grieskamp, W. Heicking, and S. Herrmann, "An Open Environment for the Integration of Heterogeneous Modelling Techniques and Tools," 1998.

[169] K.-U. Sattler, "A Framework for Component-Oriented Tool Integration," presented at Proc. of 4th Int. Conference on Object-Oriented Information Systems (OOIS'97), Brisbane, Australia, 1997.
[170] G. Karsai and J. Gray, "Component Generation Technology for Semantic Tool Integration," presented at IEEE Aerospace Conference, Big Sky, Montana, 2000.
[171] Microsoft, "Using Microsoft Tools for Business Process Management," vol. 2005, 2003.
[172] Scrittura, "The technology platform," 2005, 2004.
[173] I. B. M. C. (IBM), "IBM® WebSphere®: Business Integration Modeler Advanced EditionProduct Overview- Version 5.1.1.2," 2004,2005.
[174] P. Harmon, "Proforma's ProVision Suite version 4.4.1," 2005: Proforma Corp., 2004.
[175] C. GmbH, "COSA BPM," 2006. Pulheim, Germany, 2006.
[176] T. S. Inc., "TIBCO® Staffware Process Suite," 2006, 2006.
[177] S. AG., "SAP NetWeaver: Providing the Foundation to Enable and Manage Change," 2006. Walldorf, Germany, 2006.
[178] S. AG., "End-To-End Process Integration with SAP NetWeaver," 2006. Walldorf, Germany, 2005.
[179] S. AG., "SAP NetWeaver Platform Interoperability with IBM WebSphere and Microsoft .Net," 2006. Walldorf, Germany, 2003.
[180] S. Wade and J. A. Hopkins, "Framework for Incorporating Systems Thinking into Object Oriented Design," presented at 14th International Conference on Advanced Information Systems Engineering, Toronto, 2002.
[181] S. Patel, "Understanding Problems and Problem Situation- A Systems Approach," presented at First IEEE International Conference on Cognitive Informatics (ICCI'02), Calgary, Alberta, Canada, 2002.
[182] Y.-L. Chang, S. Chen, and C.-C. Chen, "Workflow Process Definition and Their Applications in e-Commerce," presented at 2000 International Symposium on Multimedia Software Engineering (ISMSE 2000), Taipei, Taiwan, 2000.
[183] R. Bastos, D. Dubugras, and A. Ruiz, "Extending UML Activity for Workflow Modeling in Production Systems," presented at the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Hawaii, USA, 2002.
[184] R. Bastos and D. Ruiz, "Towards an Approach to Model Business Processes using Workflow Modeling Techniques in Production Systems," presented at the 34th Annual Hawaii International Conference on System Sciences ( HICSS-34), Hawaii, USA, 2001.
[185] Y. Dong and Z. Shensheng, "Modeling Workflow Process Models with Statechart," presented at 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'03), Huntsville, Alabama, USA, 2003.
[186] W. M. P. van der Aalst, A. Kumar, and H. M. W. Verbeek, "Organizational Modeling in UML and XML in the Context of Workflow Systems.," presented at Proceedings of the 18th Annual ACM Symposium on Applied Computing (SAC 2003), 2003.
[187] C. A. Ellis and G. J. Nutt, "Modeling and Enactment of Workflow Systems," presented at the 14th International Conference on Application and Theory of Petri Nets, Chicago, Illinois, USA, 1993.

[188] OMG, "Unified Modeling Language: Superstructure version 2.0," vol. 2005, 2005.

[189] M. Chonoles and J. Schardt, *UML 2 for Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc., 2003.

[190] B. Potter, J. Sinclair, and D. Till, *An Introduction to Formal Specification and Z*, 2 nd ed.: Prentice Hall International, 1996.

[191] J. Woodcock and J. Davies, *Using Z: Specification, Refinement and Proof*: Prentice Hall International, 1996.

[192] J. M. Spivey, "The Z Notation: A Reference Manual," vol. 2005, 2 th ed. Oxford, England, 1998.

[193] J. C. M. Baeten and W. P. Weijland, *Process Algebra*: Cambridge University Press, 1990.

[194] J. A. Bergstra, A. Ponse, and S. A. Smolka, "Handbook of Process Algebra," Elsevier Science, 2001.

[195] J. B. Wordsworth, *Software Engineering with B*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1996.

[196] J. R. Abrial, *The B-Book: Assigning Programs to Meanings*: Cambridge University Press, 1996.

[197] J. J. Boyle, *The B-book: Assigning Programs to Meanings*: Cambridge University Press, 2005.

[198] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*: Prentice Hall, 1981.

[199] K. Jensen and G. Rozenberg, *High-Level Petri Nets: Theory and Application*: Springer-Verlag, 1991.

[200] H. Ehrig, *Unifying Petri Nets: Advances in Petri Nets*: Springer Verlag, 2001.

[201] C. Girault and R. Valk, *Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications*: Springer Verlag, 2003.

[202] D. Bustard and P. Lundy, "Enhancing Soft Systems Analysis with Formal Modelling," presented at IEEE Requirements Engineering Symposium, York, 1995.

[203] P. Krishnan, "Consistency Checks for UML," presented at Seventh Asia-Pacific Software Engineering Conference (APSEC'00), Singapore, 2000.

[204] R. Eshuis and R. Wieringa, "Verification Support for Workflow Design with UML Activity Graphs," presented at the 24th international conference on Software engineering, Orlando, Florida, 2002.

[205] K. Salimifard and M. Wright, "Petri net-based modelling of workflow systems: An overview," *European Journal of Operational Research*, vol. 134, pp. 664-676, 2001.

[206] Y. Yan and A. Bejan, "Modeling Workflow within Distributed Systems," presented at The Sixth International Conference on Computer Supported Cooperative Work (CSCW) in Design, London, Ontario, Canada, 2001.

[207] S. Cronholm, "Why CASE Tools in Information Systems Development? - an empirical study concerning motives for investing in CASE tools," presented at proceedings of the 18th Information Systems research In Scandinavia (IRIS 18), Gjern, Denmark, 1995.

[208] I. T. Hawryszkiewycz, *Introduction to Systems Analysis and Design*, 4 ed: Prentice Hall Australia Pty Ltd., 1998.

[209] J. Holt, *UML for Systems Engineering-Watching the Wheels*. London, United Kingdom: The Institution of Electrical Engineers, 2001.

[210] J. Iivari, "Why Are CASE Tools Not Used," *Communications of the ACM*, vol. 39, pp. 94-103, 1996.
[211] D. Software, "Canvas Software,", 2004.
[212] Corel, "CorelDraw Software,", 2004.
[213] Adobe, "Adobe Illustrator,", 2004.
[214] R. Sheppard, "Freehand 10 (Macromedia's draw software)(Evaluation)," June 2001.
[215] Macromedia, "Macromedia Freehand MX,", 2004.
[216] SmartDraw.com, "SmartDraw Software," 2004.

# Appendix A: Case Study

This appendix presents a full description of the work done on the case study, the course management.

## A.1 Introduction

The thesis case study is related to course management in a university. Due to the limitation of the research time and the difficulty to obtain information from the users or practitioners in the real world, the course management case study is considered as an illustrative one. The information of the course management is collected through interviewing my supervisor who is a member of the Board of Studies and reviewing the course development documents in the school and on the university's web site. My supervisor provides some ideas and documents about the course management.

The scenario of the case study as follows. The university offers several courses to the students. There are many aspects and procedures to plan, develop, run, and mange a course. Each course has its aims and objectives that the students are expected to acquire during their studies. The course planners organise these aims and objectives in different modules. The module is a unit of education, covering a particular subject area e.g. databases, system requirements and analysis, and programming ...etc. Each module has its aims and objectives which are parts of the course's overall aims and objectives. Also, a course requires various resources such as staff, building, different facilities and other resources which assist in achieving the course aims and objectives. In addition, it is important in the education process to measure the quality of providing the education service and the students' satisfaction or the users of this service through implementing a control sub-system that monitors and control the whole course aspects. In this study we attempt to apply this approach to the course management case study. There are many aspects and activities in both the course management and the module management that should be considered to achieve the required result from the course.

## A.2 The Approach Implementation

In this section, the SWfM approach, as described in section 4.3 will be implemented by using the course management case study. The approach has two stages: organisational business process investigation and workflow system modelling.

## i. The Organisational Business Process Investigation

This stage consists of the following steps.

**Step 1: Studying and investigating the organisation**

The course management case study is investigated to identify a future vision about the course management. It is analysed and depicted by using the rich picture.

The rich picture is a graphical representation about the course management situation that shows the different aspects of the situation. This is done through reading/consulting the course documents. Each course has different aims and objectives that can be used as the purpose of the course system to construct a root definition. The following figure (A-1) depicts the rich picture of the course management.

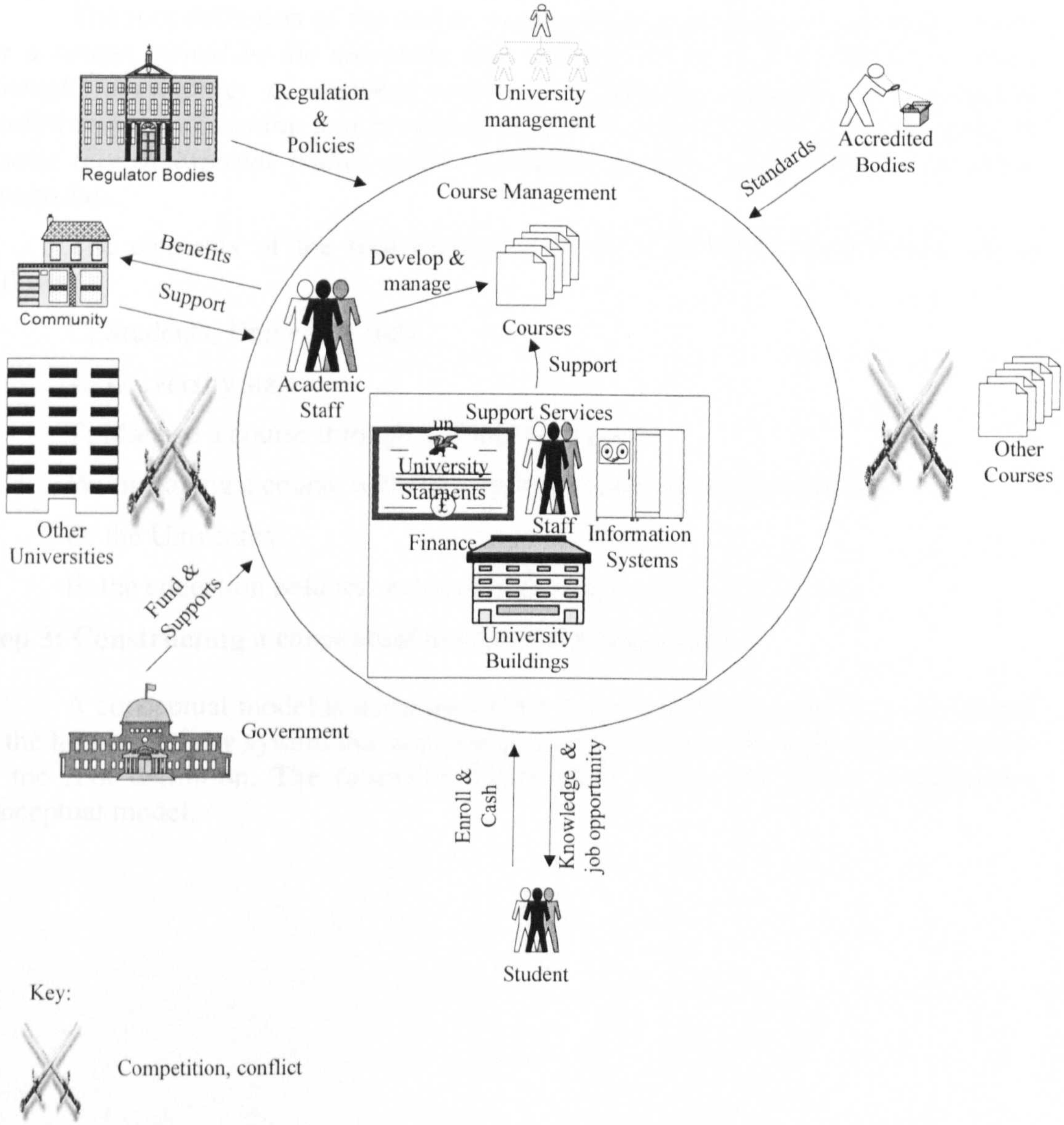



**Figure A- 1: The course management rich picture**

## Step 2: Identifying the organisation's mission or purpose

A root definition which is relevant to the problem situation is constructed and analysed using the CATWOE classification.

The root definition (RD) incorporates the purpose of the system and consists of different elements that are used to build a conceptual model of the situation. The root definition of the case study is developed from the mission statement of course management according to the first way to develop a consensus primary task model (CPTM) and by also considering the fourth way of developing CPTM which is an enterprise model assembly in which CPTM is developed from the models that describe the four aspects of the enterprise such as the enterprise core business or process, its support services, its relationship with environment and its managing process [153].

The root definition of the course management is *developing a workflow model for a system owned by the university and operated by its staff to manage a course through developing a required course, graduating students with required qualifications by teaching and providing them with required skills and assessing the course though external bodies within education policies and financial resources constraints.*

The elements of the root definition of the CATWOE classification are as follows:

C: Students, University staff

A: University staff

T: Manage a course through a workflow system

W: managing a course with the available resources and constraints.

O: the University

E: the education policies, external bodies and financial resources

**Step 3: Constructing a conceptual model of the organisation**

A conceptual model is developed from the root definition which is considered as the human activity system that consists of activities that achieve the aim or purpose of the root definition. The following figure (A-2) shows the course management conceptual model.

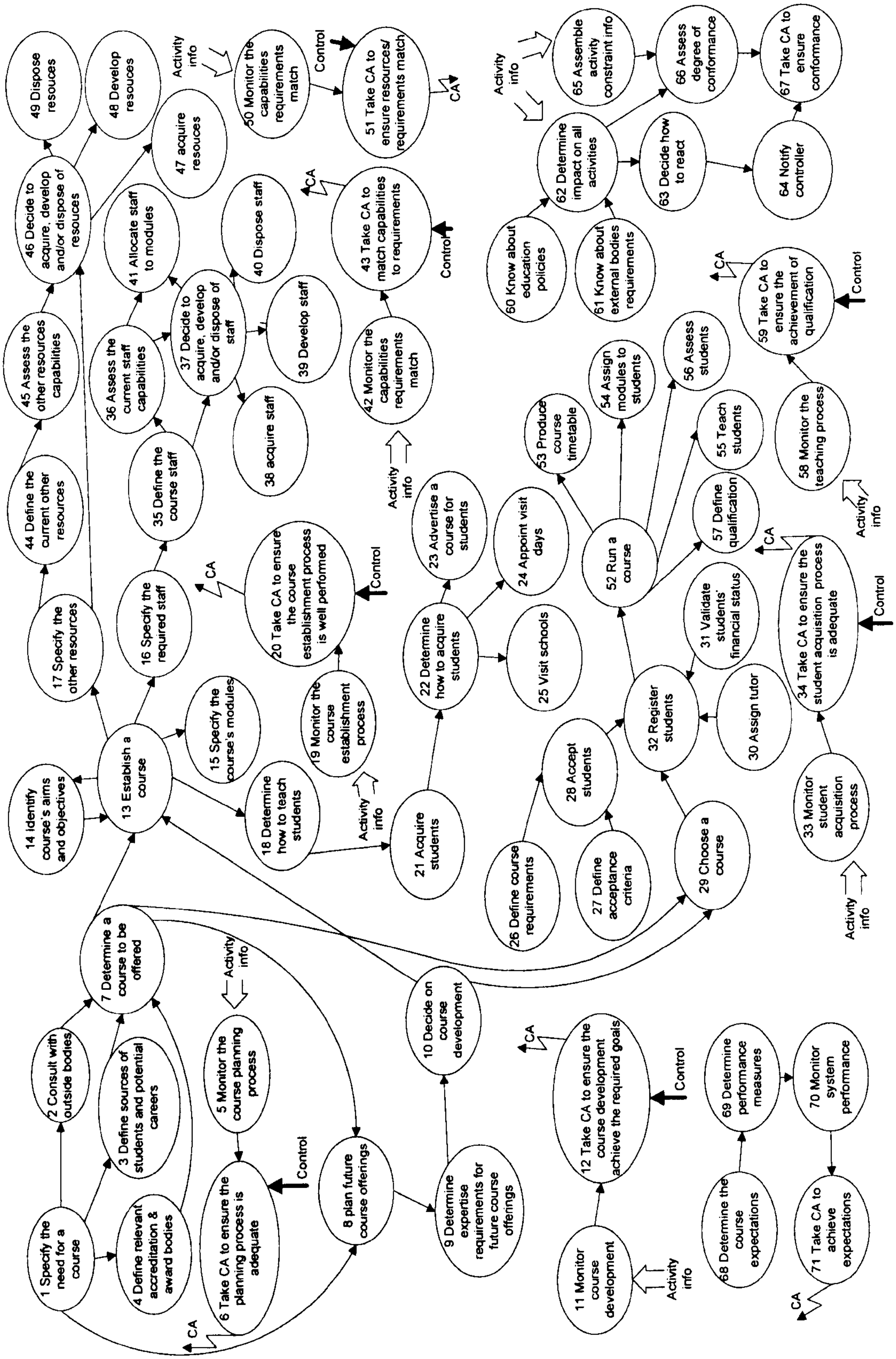



**Figure A- 2: The course management conceptual model**

The various activities of the conceptual model can be grouped into sub-systems. The following figure (A-3) shows the high-level conceptual model of the course management that contains the various sub-systems.

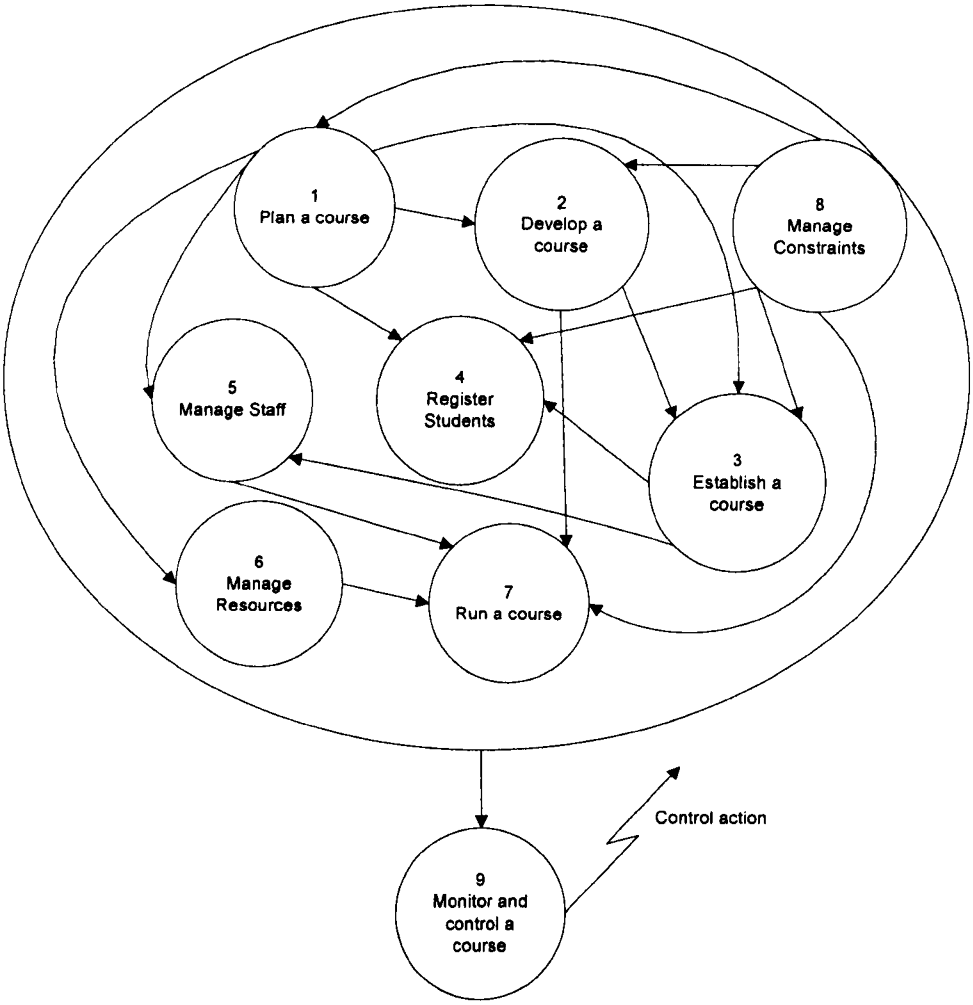



**Figure A- 3: The course management high-level conceptual model**

These high-level sub-systems are as follow:

1) The course planning sub-system
2) The course development sub-system
3) The course establishment sub-system
4) Student registration sub-system (It can be divided into acquisition subsystem and registration subsystem)
5) Staff management sub-system
6) Resources management sub-system
7) Course running sub-system
8) Constraints management sub-system
9) Overall control sub-system

The following figure (A-4) shows the grouping of the various activities of the conceptual model into high-level sub-systems after excluding the monitoring and control activities and grouping them into the control sub-system.

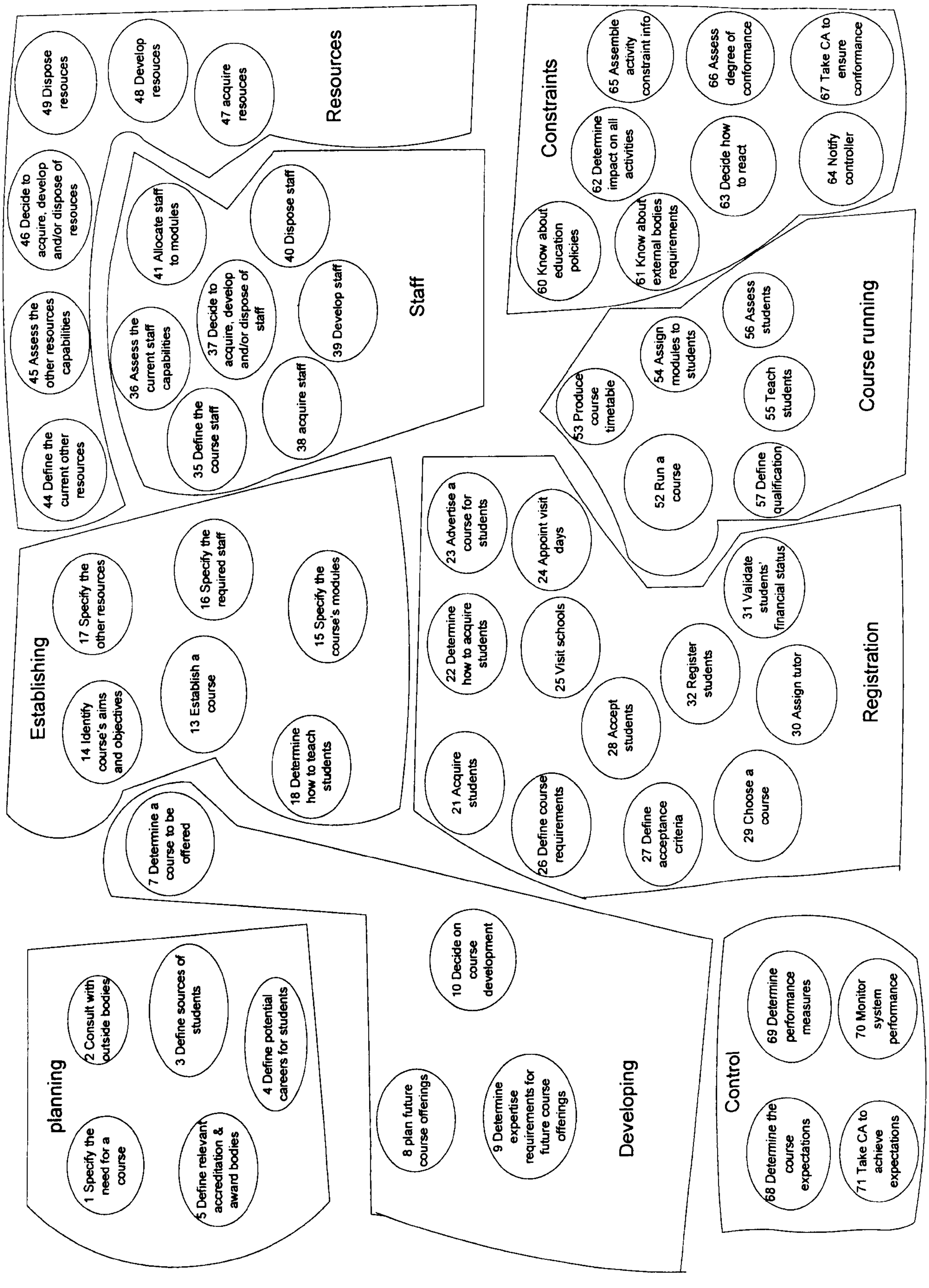



**Figure A- 4: The conceptual model activities grouped into sub-systems**

The course management conceptual model can be tested or validated against the formal system model rules. All the activities of the conceptual model reflect and satisfy the root definition. Also, the activities are connected to each other by using the logical dependency arrow which means that an activity at the head of the arrow depends on the activity of the other side of the arrow and will not be executed until the completion of that activity. For example the activity 'identify the course's aims and objectives' depends on the activity 'establish a course'. Finally, there is an 'overall control' subsystem which monitors the performance of each activity, collects information about it, assesses it with the setting measures for each activity and takes control actions to rectify any deficiency.

**Step 4: Comparing the conceptual model with the real world**

The developed conceptual model is compared with the current organisational business process (if it is available) to validate that the resulting conceptual model is fit for the organisation and to identify any missing activities or processes.

In this case study there is no existing model for an organisational business process. Thus the conceptual model is considered as the basis to model the organisational business process as a workflow system in the following stage.

## ii. The Workflow System Modelling

**Step 1: Constructing a Subsystem description table**

A subsystem description table is developed for each subsystem of Figure (A-3) to present its number, name and constituent activities. Table 5-1 shows the course planning subsystem.

| | |
|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning |
| Subsystem Head: (head of subsystem) | |
| Subsystem Activities:<br>1. Specify the need for a course<br>2. Consult with outside bodies<br>3. Define sources of students and potential careers<br>4. Define relevant accreditation and award bodies | |

**Table A- 1: 'The course planning' subsystem**

**Step 2: Constructing an activity description table**

An activity description table is developed for each activity of the subsystem to define in detail each activity within a subsystem. Table A-2 shows the activity 'Specify the need for a course'.

| | | |
|---|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning | |
| Activity Number: 1 | Activity Name: Specify the need for a course | |
| Preceding Activity: none | Following Activity: Consult with outside bodies | |
| Precondition: none | | |
| Input Data:<br>1. ICP | Activity Tasks:<br>1. A lecture discusses with Senior Staff (SS). | Output Data:<br>1. RICP |

| | | |
|---|---|---|
| | 2. A lecture presents ICP to Board of studies (BoS).<br>3. BoS discusses paper (ICP)<br>4. BoS makes recommendations (RICP or no course) possibly with changes. | |
| Business Rules and Constraints: an initial course proposal should be ready for discussion. | | |
| Postcondition: A decision to accept the course proposal or no is taken by BoS. | | |
| Required Skills and Capabilities:<br>Some experiences on developing a course proposal | Role Name: BoS | |
| Performance Criteria: - The required time to evaluate the course proposal.<br>- The course proposal standard. | | |

**Table A- 2: The 'Specify the need for a course' Activity**

| | | |
|---|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning | |
| Activity Number: 2 | Activity Name: Consult with outside bodies | |
| Preceding Activity: Specify the need for a course | Following Activity: Define sources of students and potential careers | |
| Precondition: Complete activity 1 | | |
| Input Data:<br>1. RICP | Activity Tasks:<br>1. BoS sends ICP to outside body.<br>2. Outside body returns comments.<br>3. Comments are attached to RICP or no course. | Output Data:<br>1. RICPA |
| Business Rules and Constraints: the course proposal should be complete to send it to the outside bodies for comments. | | |
| Postcondition: Comments from the outside bodies has been obtained. | | |
| Required Skills and Capabilities: some experiences and skills to communicate with outside bodies to get their comments about evaluating a course proposal. | Role Name: BoS | |
| Performance Criteria: - the necessary time to communicate with the outside bodies<br>- the usefulness of the obtained comments about the course proposal | | |

**Table A- 3: The 'Consult with outside bodies' Activity**

| | |
|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning |
| Activity Number: 3 | Activity Name: Define sources of students and potential careers |
| Preceding Activity: Consult with outside bodies | Following Activity: Define relevant accreditation and award bodies |
| Precondition: complete activity 1 | |

<table>
<tr><td colspan="2">Input Data:<br>1. RICP</td><td colspan="2">Activity Tasks:<br>1. BoS discusses ICP with student representative and potential careers people.<br>2. BoS provides views as addendum to recommendations (RICP or no course)</td><td colspan="2">Output Data:<br>1. RICPA</td></tr>
<tr><td colspan="6">Business Rules and Constraints: the course proposal should be completed to define the quality of the required student and their sources and to identify potential careers for them.</td></tr>
<tr><td colspan="6">Postcondition: - establishing the required students quality and their sources<br>- defining the potential careers for the graduated students</td></tr>
<tr><td colspan="3">Required Skills and Capabilities: the abilities to set a standard for the quality of the required students and their sources and the potential careers for them.</td><td colspan="3">Role Name: BoS</td></tr>
<tr><td colspan="6">Performance Criteria: - the time to set a standard for the quality of the required students<br>- specifying the potential careers for the graduated students</td></tr>
</table>

**Table A- 4: The 'Define sources of students and potential careers' Activity**

<table>
<tr><td colspan="3">Subsystem Number: 1</td><td colspan="3">Subsystem Name: Course planning</td></tr>
<tr><td colspan="3">Activity Number: 4</td><td colspan="3">Activity Name: Define relevant accreditation and award bodies</td></tr>
<tr><td colspan="3">Preceding Activity: Define sources of students and potential careers</td><td colspan="3">Following Activity: none</td></tr>
<tr><td colspan="6">Precondition: complete activity 1</td></tr>
<tr><td colspan="2">Input Data:<br>1. RICP</td><td colspan="2">Activity Tasks:<br>1. BoS identifies for ICP accreditation bodies<br>2. BoS appends accreditation bodies as addendum to recommendations (RICP or to course)</td><td colspan="2">Output Data:<br>1. RICPA</td></tr>
<tr><td colspan="6">Business Rules and Constraints: the course proposal should be complete to get an accreditation from the award bodies</td></tr>
<tr><td colspan="6">Postcondition: Comments or accreditation has been obtained from the award bodies</td></tr>
<tr><td colspan="3">Required Skills and Capabilities: the ability to manage and communicate with the award bodies to get an accreditation for the course proposal.</td><td colspan="3">Role Name: BoS</td></tr>
<tr><td colspan="6">Performance Criteria: - the necessary time to communicate with the award bodies<br>- the obtained result regarding the course proposal from the award bodies</td></tr>
</table>

**Table A- 5: The 'Define relevant accreditation and award bodies' Activity**

**Case study terms**

BoS: Board of Studies

SS: Senior Staff

UF: University Faculty

AT: Administration Tutor

CA: Course Administrator

T: Tutor

L: Lecturer

BoF: Board of Faculties

ICP: Initial Course Proposal

RICP: Refined Initial Course Proposal

RICPA: Refined Initial Course Proposal and Addendum

VCP: Viable Course proposal

AC: Active Course

**Step 3: Modelling the various perspectives of the workflow system**

In this section, the various workflow perspectives are reviewed to address the required technique or techniques to model each one. The conceptual model and the two tables above are used to define the various perspectives of the workflow system.

**Step 4: Unified Modelling Language (UML) and workflow perspectives**

**Step 4.1: Converting the activities of the conceptual model into use cases**

The conceptual model activities will be mapped into UML use cases through examining each activity and identifying a goal for each activity.

Next, these activities are mapped into activities with goals that have to be achieved. This step may involve decomposing an activity into two or more activities that have different goals or combining some activities into one activity that accomplishes one goal. Figure (A-5) shows the same conceptual model of figure (A-4) with including the goals of the activities of the first three sub-systems of the course management. Time constraints are prevented the consideration of further sub-systems.

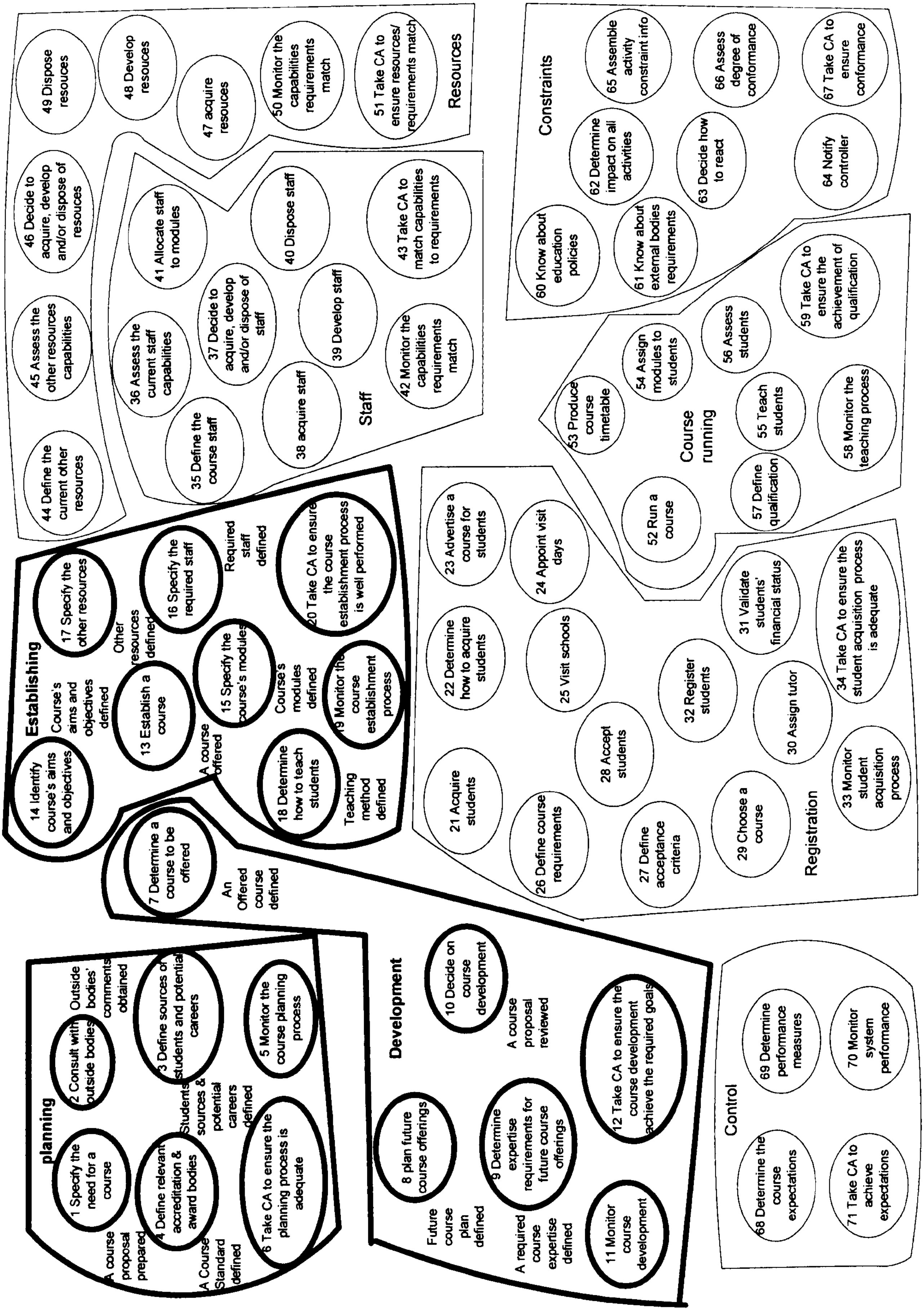



**Figure A- 5: The conceptual model activities with goals**

After that, the new activities with the required goals that must be accomplished are examined to map them as one to one onto a UML use case that achieves the same goal. Some activities can be decomposed into two or more activity-use-case maps. For example, the activity 'define student sources and potential careers' in the 'course planning' sub-system could have two different goals. Thus the activity is decomposed into two activity-use-case maps 'define sources of students' and 'define potential careers for students'. Therefore, it is important to update the sub-system description table through adding these two activities.

<table>
<tr><td>Subsystem Number: 1</td><td>Subsystem Name: Course planning</td></tr>
<tr><td>Subsystem Head: (head of subsystem)</td><td></td></tr>
<tr><td colspan="2">Subsystem Activities:<br>1. Specify the need for a course<br>2. Consult with outside bodies<br>3. Define sources of students<br>4. Define potential careers for students<br>5. Define relevant accreditation and award bodies</td></tr>
</table>

**Table A- 6: The modified 'course planning' subsystem**

Also, these two activities can be described either through constructing an activity description table for every activity and changing the sequence order of the activities (activity no. 4 'define relevant accreditation and award bodies' will be activity no. 5) or just defining their details in the use case template when they are mapped into use cases.

<table>
<tr><td colspan="2">Subsystem Number: 1</td><td colspan="2">Subsystem Name: Course planning</td></tr>
<tr><td colspan="2">Activity Number: 3</td><td colspan="2">Activity Name: Define sources of students</td></tr>
<tr><td colspan="2">Preceding Activity: Consult with outside bodies</td><td colspan="2">Following Activity: Define potential careers for students</td></tr>
<tr><td colspan="4">Precondition: complete activity 1</td></tr>
<tr><td>Input Data:<br>1. RICP</td><td colspan="2">Activity Tasks:<br>1. BoS discusses ICP with student representative.<br>2. BoS provides views as addendum to recommendations (RICP or no course)</td><td>Output Data:<br>1. RICPA</td></tr>
<tr><td colspan="4">Business Rules and Constraints: the course proposal should be completed to define the quality of the required students and their sources.</td></tr>
<tr><td colspan="4">Postcondition: - establishing the required students quality and their sources</td></tr>
<tr><td colspan="2">Required Skills and Capabilities: the abilities to set a standard for the quality of the required students and their sources.</td><td colspan="2">Role Name: BoS</td></tr>
<tr><td colspan="4">Performance Criteria: - the time to set a standard for the quality of the required students</td></tr>
</table>

**Table A- 7: The 'Define sources of students' new Activity**

<table>
<tr><td>Subsystem Number: 1</td><td>Subsystem Name: Course planning</td></tr>
<tr><td>Activity Number: 4</td><td>Activity Name: Define potential careers for students</td></tr>
<tr><td>Preceding Activity: Define sources of</td><td>Following Activity: Define relevant</td></tr>
</table>

| | | |
|---|---|---|
| students | accreditation and award bodies | |
| Precondition: complete activity 1 | | |
| Input Data:<br>1. RICP | Activity Tasks:<br>1. BoS discusses ICP with potential careers people.<br>2. BoS provides views as addendum to recommendations (RICP or no course) | Output Data:<br>1. RICPA |
| Business Rules and Constraints: the course proposal should be completed to identify potential careers for the graduated students. | | |
| Postcondition: - defining the potential careers for the graduated students | | |
| Required Skills and Capabilities: the abilities to set a standard for the potential careers for the graduated students. | Role Name: BoS | |
| Performance Criteria: - the time to define a standard for the potential careers for the graduated students | | |

**Table A- 8: The 'Define potential careers for students' Activity**

| | | |
|---|---|---|
| Subsystem Number: 1 | Subsystem Name: Course planning | |
| Activity Number: 5 | Activity Name: Define relevant accreditation and award bodies | |
| Preceding Activity: Define potential careers for student | Following Activity: none | |
| Precondition: complete activity 1 | | |
| Input Data:<br>1. RICP | Activity Tasks:<br>1. BoS identifies for ICP accreditation bodies<br>2. BoS appends accreditation bodies as addendum to recommendations (RICP or to course) | Output Data:<br>1. RICPA |
| Business Rules and Constraints: the course proposal should be complete to get an accreditation from the award bodies | | |
| Postcondition: Comments or accreditation has been obtained from the award bodies | | |
| Required Skills and Capabilities: the ability to manage and communicate with the award bodies to get an accreditation for the course proposal. | Role Name: BoS | |
| Performance Criteria: - the necessary time to communicate with the award bodies<br>- the obtained result regarding the course proposal from the award bodies | | |

**Table A- 9: The 'Define relevant accreditation and award bodies' Activity**

Then, each activity-use-case map in this model can be mapped into a use case. Further, the monitoring and controlling activities are combined into the control sub-system to be considered in one sub-system as a centralised control sub-system.

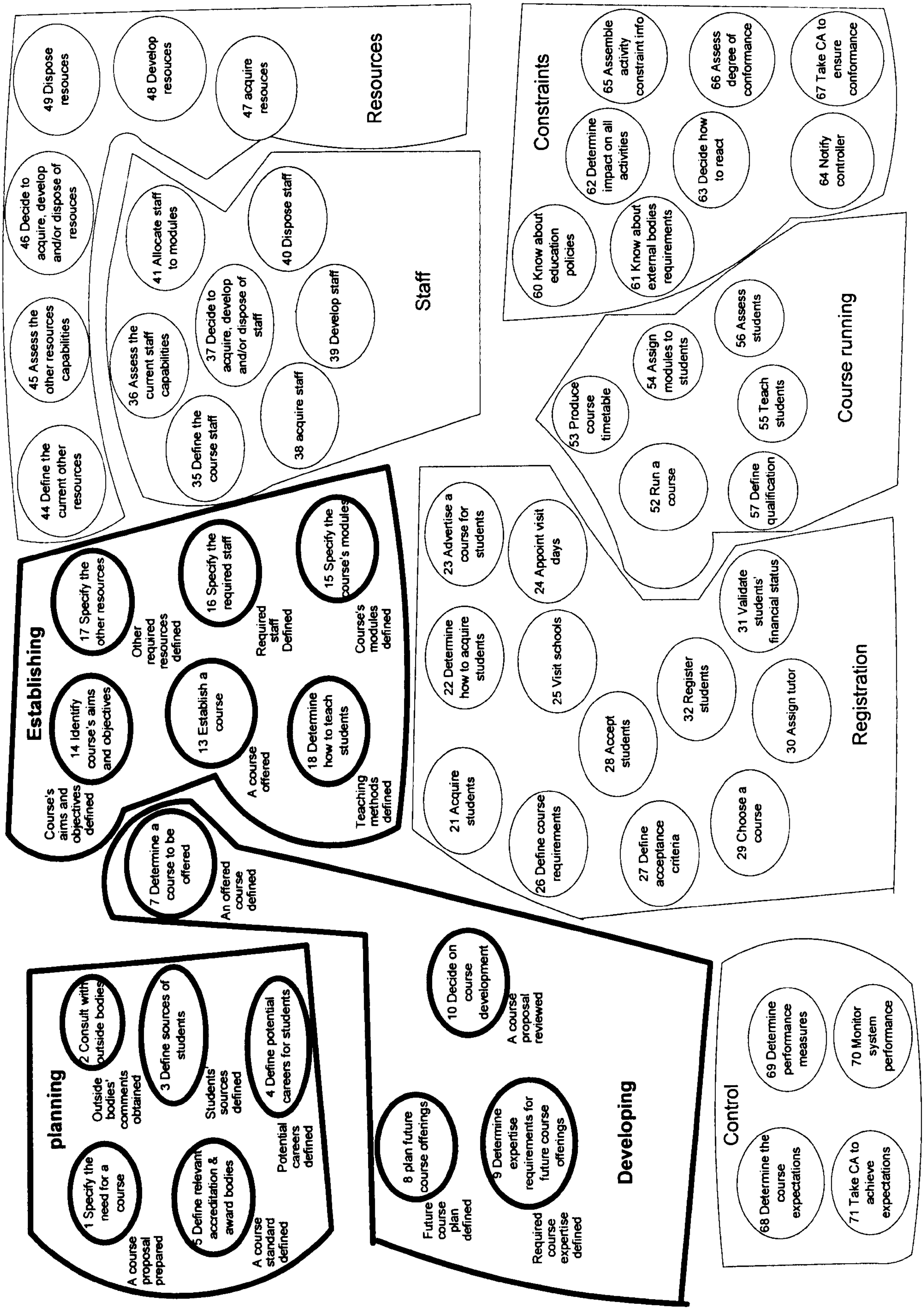



**Figure A- 6: Activities to use cases mapping**

In the next step, the use cases of the business process will be depicted in the use case diagram that consists of the use cases with their actors. The aim of the use case model, which is the collection of the use case diagrams, is to present and document the organisational business process. This model will be used as a basis for modelling the workflow perspectives through using the various UML diagrams. Finally it is possible to design and implement the workflow system. Figure (A-6) shows the use case diagram for the course planning sub-system of the course management.

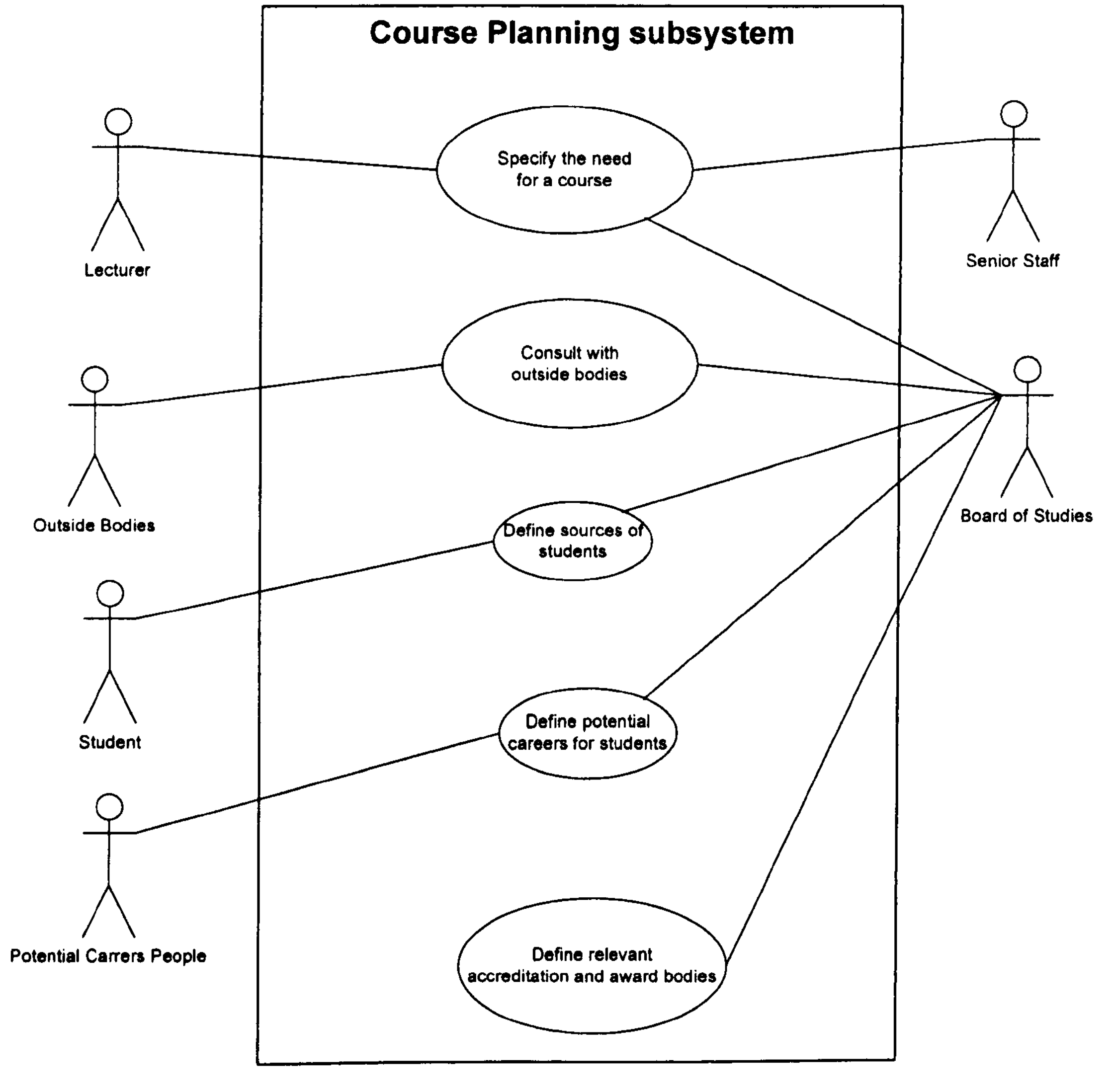



**Figure A- 7: The use cases of a course planning sub-system**

## Step 4.2: Analysing the use cases of the business process

These use cases are considered as the base to model, analyse the required system (workflow system) by using UML and the object-oriented approach and later design and implement it.

For each use case, it is essential to describe it in detail by using textual format of the use case template that defined in chapter five. The following use case template represents 'specify the need for a course' use case of the course planning sub-system. The course planning sub-system includes four use cases as follow:

A) The 'specify the need for a course' use case

i. Name: specify the need for a course

ii. Brief description

This use case describes how to specify the need for a course through the following steps:

a. A lecture discusses with Senior Staff (SS).
b. A lecture presents ICP to Board of Studies (BoS).
c. BoS discusses paper (ICP)
d. BoS makes recommendations (RICP or no course) possibly with changes.

iii. Performance goals

iv. Benefit/value

v. Workflow/ flow of events

vi. Special requirements

vii. Extension points

viii. Relationships

Some of these features will be added in an iterative way by adding some detail in each iteration when the detail is available and through the progression of the project. It is possible to prototype some of the use cases by showing the interfaces that the users will use when they are carrying out their tasks.

**Step 4.3: Developing an activity diagram for each use case**

Uses cases are modelled by using the activity diagrams. This diagram represents and documents the functionality of the use case. It shows an overview picture of the use case by depicting its various components. It is used to model the functional, informational, behavioural, and organisational views of the workflow system. However, the activity diagram needs to be improved to include some aspects of the business process such as the resources and some other control symbols. The activity diagram in figure (A-8) models the workflow of the 'specify a need for a course' use case. It shows the roles that are involved in this use case such as lecturer, senior staff and Board of Studies which represent the organisational perspective of the workflow system and the activities. The latter represent the functional perspective that are carried out, for example *discuss the need for a course, prepare a course's proposal, submit a course's proposal etc.* A swimlane is defined for each role which includes the activities that are performed inside the boundary of this role. The activity diagram starts with the initial state (black circle) followed by the activities (a rectangle with round sides). The control flow is specified by using transitions (directed line) that define the path from one activity to another which indicates sequential activities. When an activity completes, the control flow passes to the next activity. It is possible to use the other control symbols such as decision (a diamond with a guard condition to define various paths), fork (a synchronisation bar with one input and several outputs to represent the splitting of a single control flow into two or more parallel flows of control) and join (a synchronisation bar with several inputs and one output to represent the joining of one or more control flows into one control flow) to model the flow of control. The control flow of the use case represents the behaviour perspective. The informational perspective is modelled by using the object flow that represents the object that is used within the activity as an input or output parameter and it is possible to depict its state if the activity modifies it. The object is connected to the activity through a dependency relationship (a dotted line) and its state is depicted in brackets below the object's name. The activity diagram concludes with the final state (a black circle within a white circle within a black border).

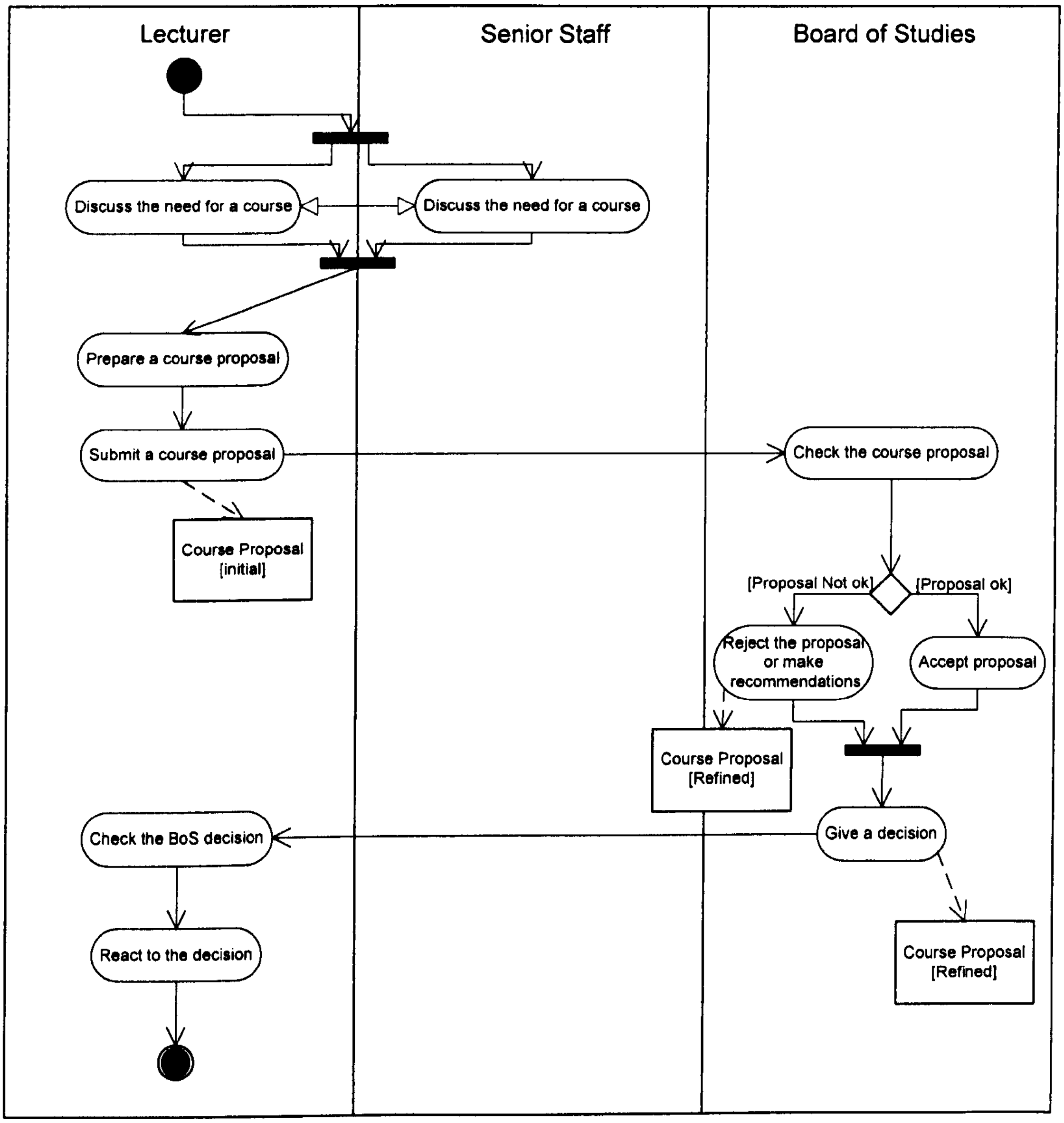



**Figure A- 8: An activity diagram for a 'specify the need for a course' use case**

## Step 4.4: Developing a sequence diagram for each use case

The interactions among the objects of the use case are modelled by using the sequence diagram. It is used to provide more detail about the workflow system through depicting the various objects and the exchangeable messages between them. This will assist the designer in designing the workflow system and implementing it in the implementation stage. This diagram is used to model the dynamic aspect of the workflow system through showing the interaction between the system objects to carry out the various tasks or activities. The interaction messages that are received by the objects are considered as operations in these objects. Figure (A-9) shows the sequence diagram for the 'specify a need for a course' use case. It models the interactions between the various objects to carry out the use case behaviour through the various messages. The various objects (with a dotted line to represent their lifelines) are modelled at the top of the diagram such as lecturer, senior staff and Board of Studies. The diagram flows from top to bottom to depict the passing of time. The interaction between objects occurs by sending messages (a directed line). When a message is sent to an object, it calls an operation of that object which will be executed. The period of time during which an operation executes is known as an activation (a rectangle block covers the object lifeline). It is possible to model the construction and the destruction of objects on the sequence diagram. The created object is depicted directly to the message arrowhead such as the creation of a course's proposal object while the destruction of an object is modelled by a large X on the object lifeline. An object can send a message to itself. This is called a reflexive message (a message arrow that starts and ends at the same object lifetime). The 'check the decision' message in the figure represents a reflexive message. A return message is optional and can be included in the diagram to show the return of control to the object that initiates the message that starts the activation. The iteration is represented by an asterisk before the message name and the condition that specifies the iteration limit can be added before the asterisk symbol and the message.

A branching can be added to the sequence diagram if the use case has more than one execution path. Each path is depicted by a condition guard inside square brackets and a message name that starts from the same object lifeline. Each path is followed only if the path condition is true.

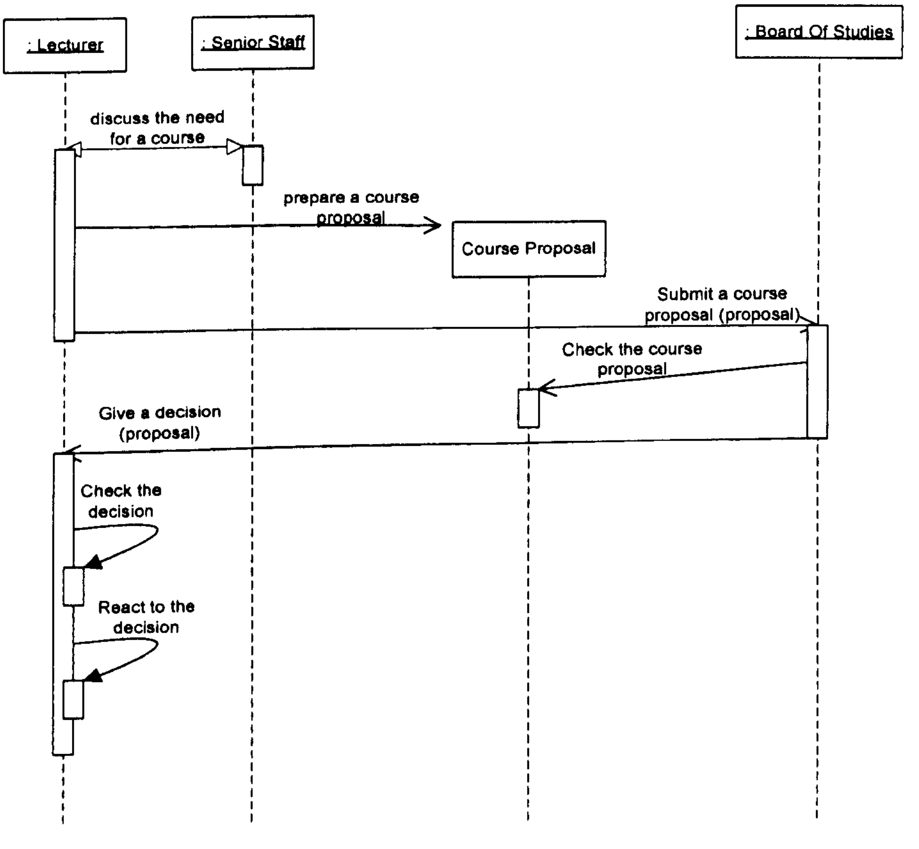



**Figure A- 9: A sequence diagram for a 'specify the need for a course' use case**

### Step 4.5: Developing a class diagram for the static structure of each use case

The classes of the use case are modelled by using the class diagram. This diagram is used to model the static structure of the use case by representing its classes, and the relationships between them, which collaborate to carry out its behaviour. The class modelling includes specifying its attributes, operations and its relationships with the other classes. Figure (A-10) shows the class diagram for a 'specify the need for a course' use case. This diagram includes four classes with their attributes, operations and the relationships among them.

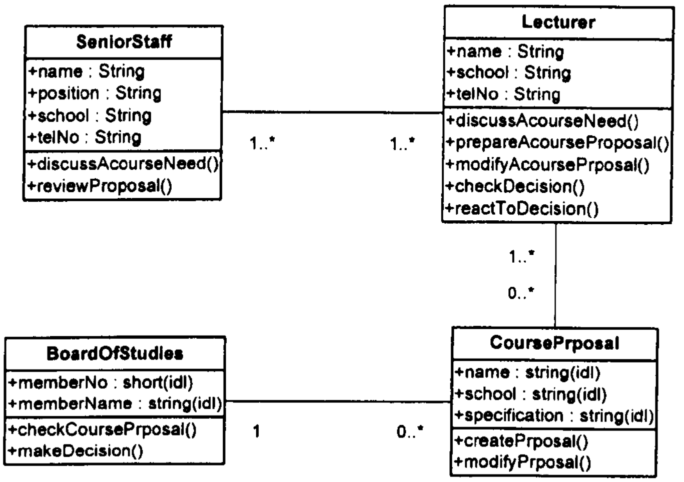



**Figure A- 10: A class diagram for a 'specify the need for a course' use case**

### Step 4.6: Developing a class diagram for the organisation structure of each use case

The different roles of the organisation perspective are modelled by using the class diagram. This diagram is used to model the different roles of the organisation that interact with each other to perform the use case behaviour by representing each role as a class with a stereotype <<role>>. Also, the attributes, operations (responsibilities) and relationships with the other classes are added to each role. Figure (A-11) depicts the class diagram of the organisation perspective of a 'specify the need for a course' use case. This diagram includes three classes that represent the three roles involved in carrying out the use case behaviour and their relationships.

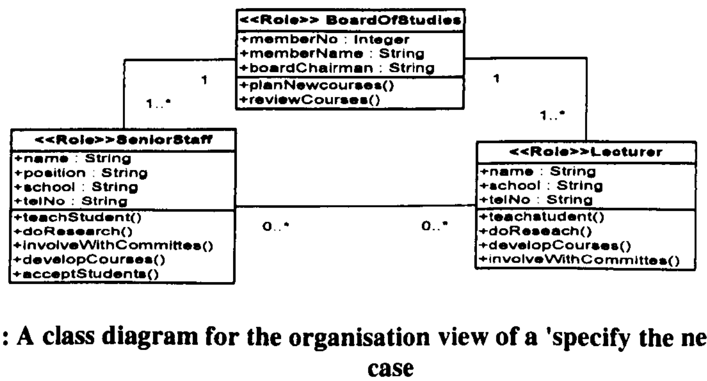
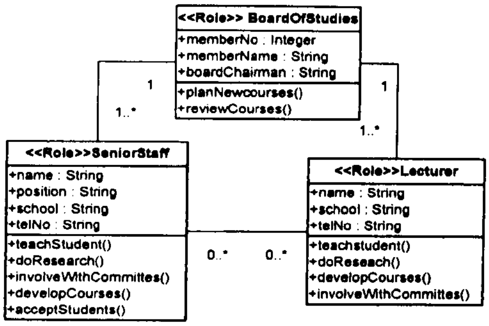
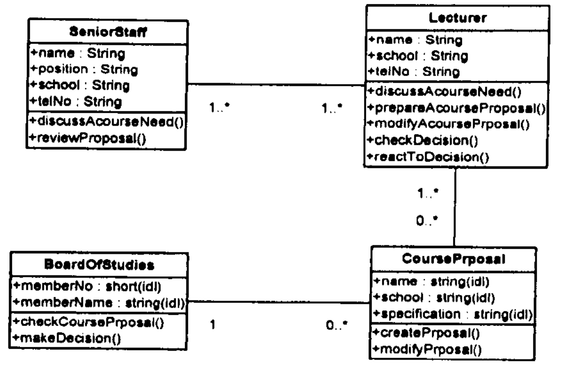



**Figure A- 11: A class diagram for the organisation view of a 'specify the need for a course' use case**

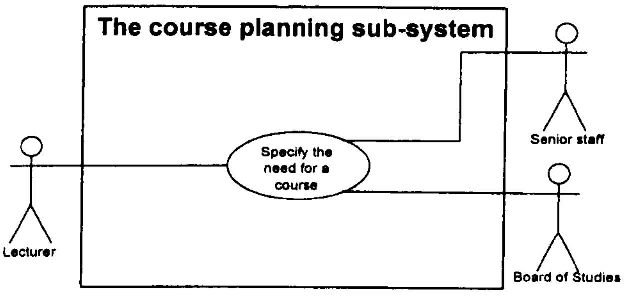
All the diagrams of the 'specify the need for a course' use case are combined into one model that shows the different views or perspectives of the workflow system. Figure (A-12) shows the collections of the views of a use case. In the middle of the figure is the use case that is analysed and modelled. Also, there is an activity diagram which shows the different perspectives of the use case such as the organisational perspective (the organisation roles), the functional perspective (activities), the behaviour perspective (the flow of control) and the informational perspective (object flow). The sequence diagram depicts the dynamic aspect of the workflow system by presenting the interactions among the objects to perform the use case behaviour. Finally, the two class diagrams model the static structure of organisational perspective and the workflow system.

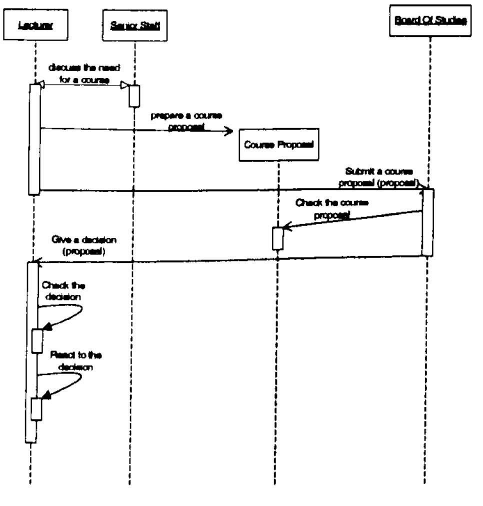



A class diagram for the organisation view of a 'specify the need for a course' use case

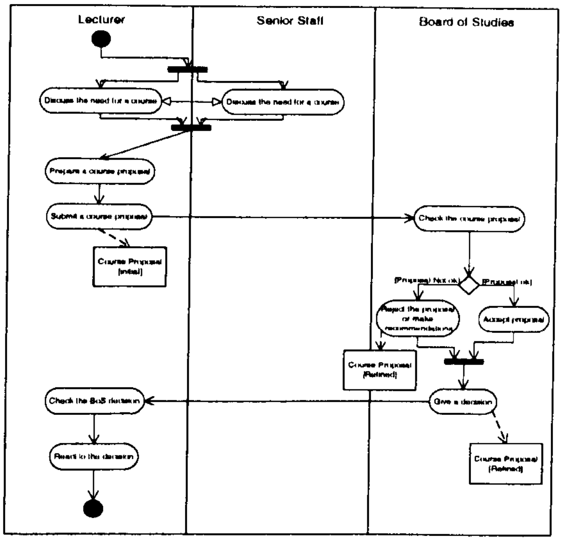



A class diagram for a 'specify the need for a course' use case




A sequence diagram for a 'specify the need for a course' use case




An activity diagram for a 'specify the need for a course' use case

**Figure A- 12: All diagrams for a 'specify the need for a course' use case**

**(larger versions of the components are given in pervious figures)**

The steps from one to five are repeated for each use case in the sub-system.

The next use case of the 'course planning' sub-system is a 'consult with outside bodies' use case. The steps from one to five are implemented on this use case as follow:

B) The 'consult with outside bodies' use case

The use case is analysed by specifying its detail.

i. Name: consult with outside bodies

ii. Brief description

This use case describes how to consult the outside bodies about the course and its contents through the following steps:

a. Board of Studies sends Initial Course Proposal (ICP) to outside body.
b. Outside body returns comments.
c. Comments are attached to Refined Initial Course Proposal (RICP).

An activity diagram is developed for the use case.

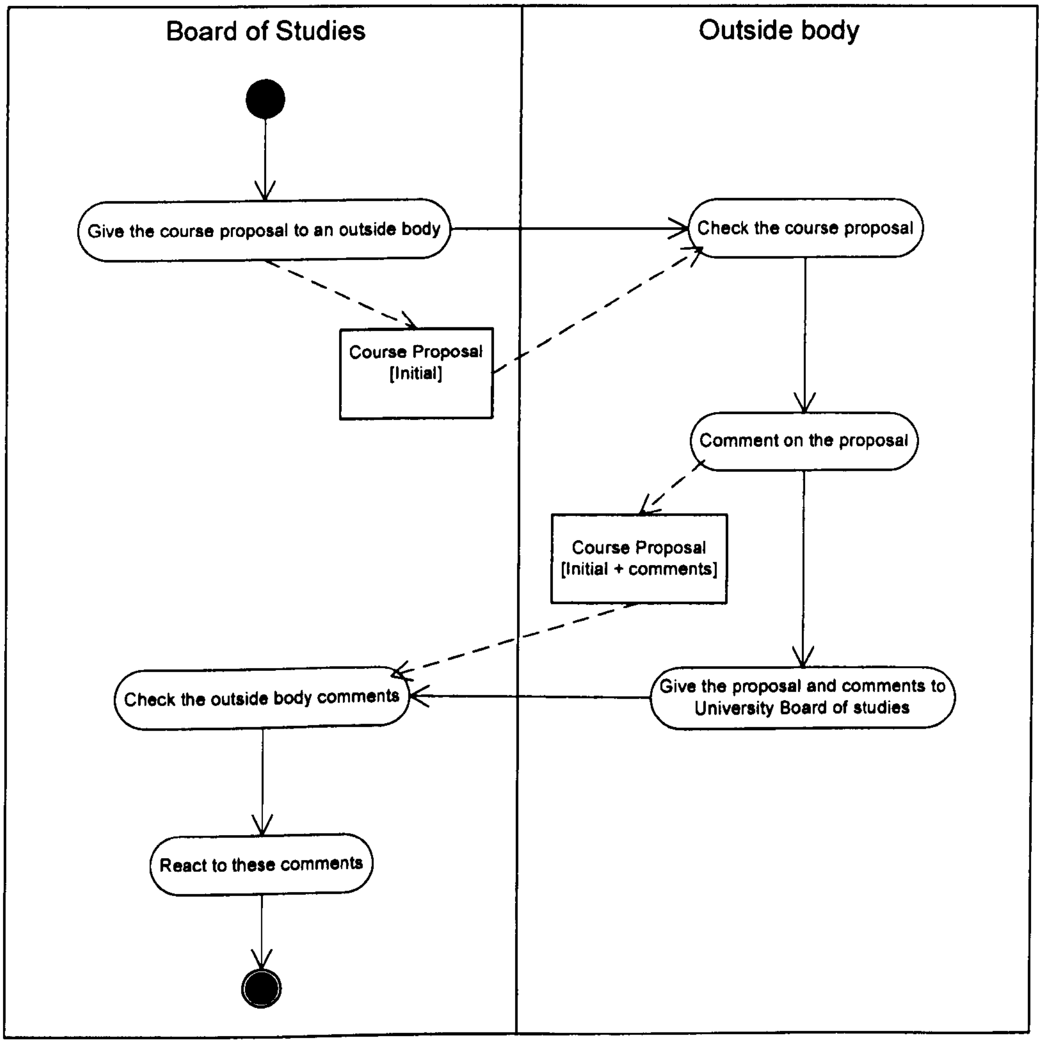



**Figure A- 13: An activity diagram for a 'consult outside bodies' use case**

A sequence diagram is constructed for the use case.

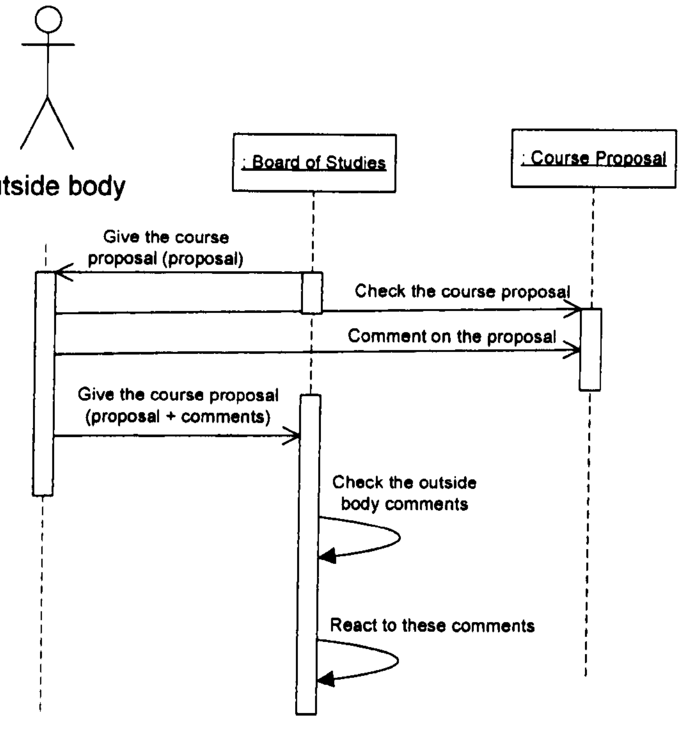



**Figure A- 14: A sequence diagram for a 'consult outside body' use case**

A class diagram is formed to present the use case classes that are interacted to perform its functionality.

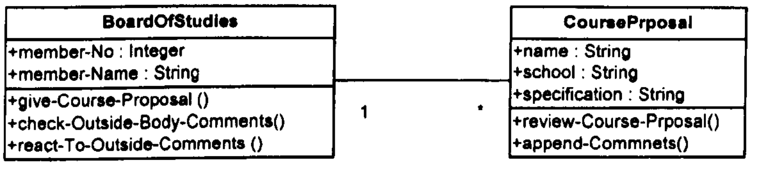



**Figure A- 15: A class diagram for a 'consult outside body' use case**

A class diagram is built to present the various roles that are involved in carrying out the functionality of the use case.

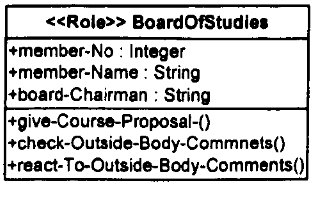
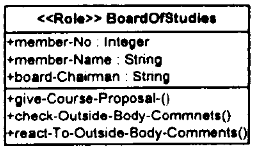
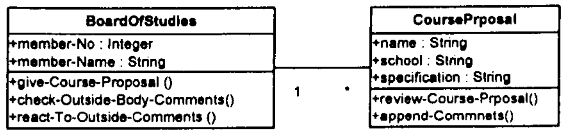



**Figure A- 16: A class diagram for the organisation view of a 'consult outside body' use case**

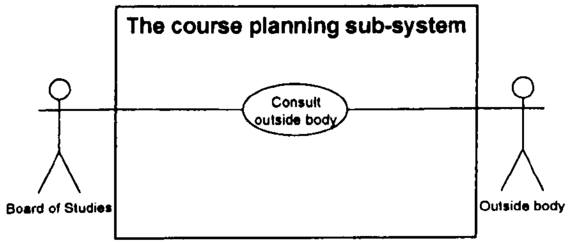
All these diagrams can be combined on one diagram to show all the perspectives of the modelled use case.

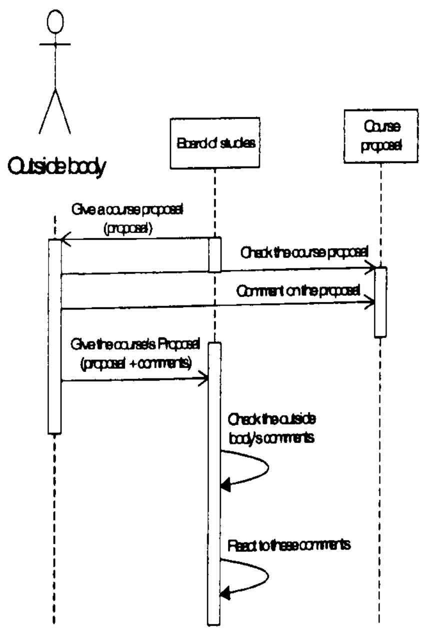



A class diagram for the organisation view of a 'consult outside body' use case

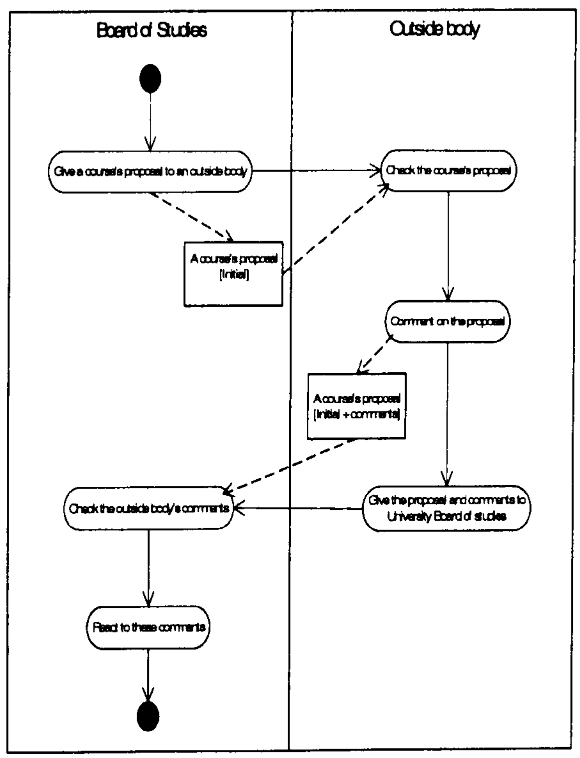



A class diagram for a 'consult outside body' use case




A sequence diagram for a 'consult outside body' use case




An activity diagram for a 'consult outside bodies' use case

**Figure A- 17: All diagrams for a ' consult outside body ' use case**

C) The 'define sources of students' use case

The use case is analysed by specifying its detail.

i. Name: define sources of students

ii. Brief description

This use case describes how to acquire students and define their sources through the following steps:

a. Board of Studies (BoS) discusses Initial Course Proposal (ICP) with student representative.
b. BoS provides views as addendum to recommendations (RICP).

An activity diagram is developed for the use case.

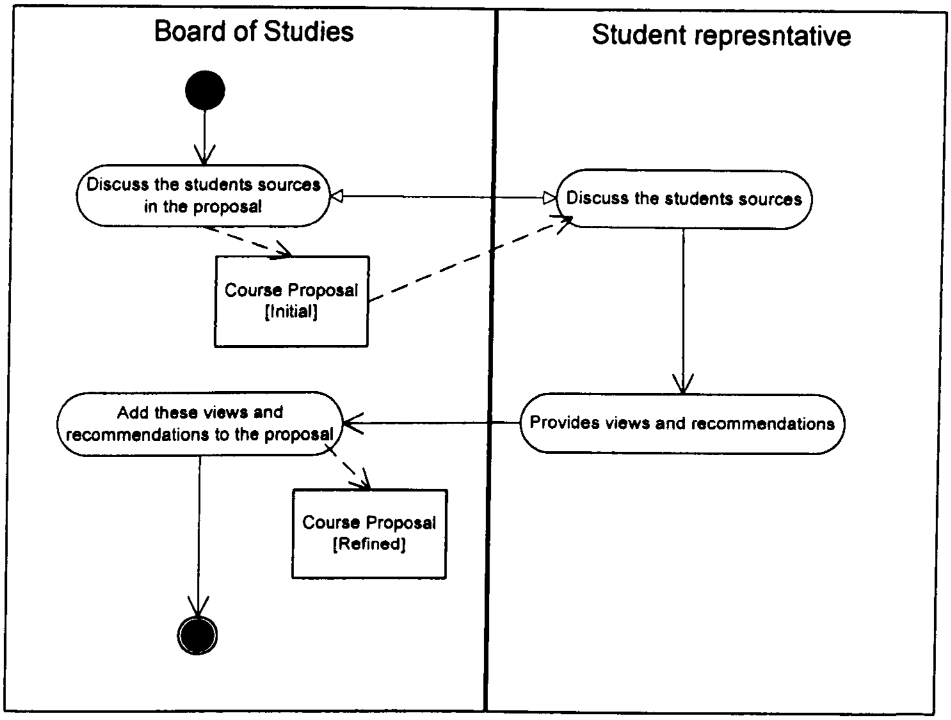



**Figure A- 18: An activity diagram for a 'define sources of students' use case**

A sequence diagram is constructed for the use case.

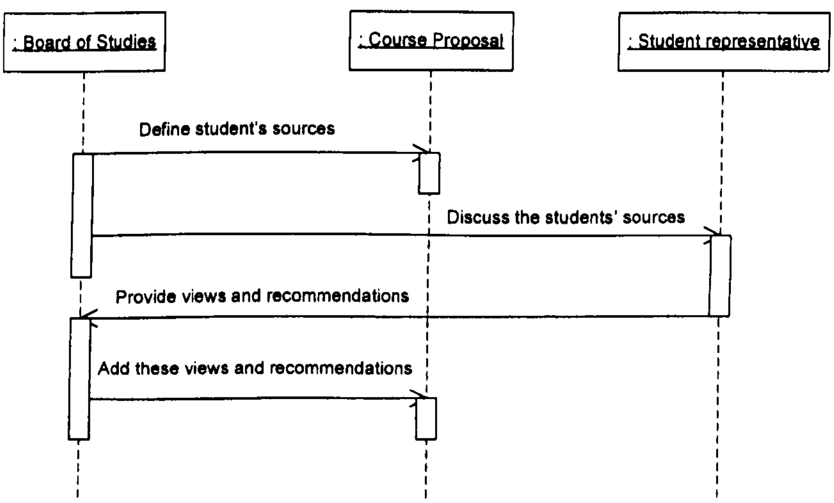



**Figure A- 19: A sequence diagram for a 'define sources of students' use case**

A class diagram is formed to present the use case classes that are interacted to perform its functionality.

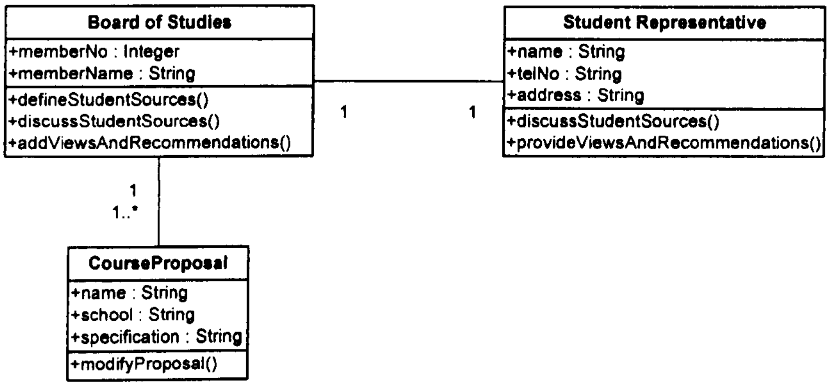



**Figure A- 20: A class diagram for a 'define sources of students' use case**

A class diagram is built to present the various roles that are involved in carrying out the functionality of the use case.

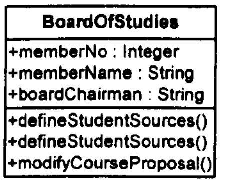
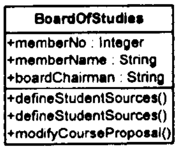
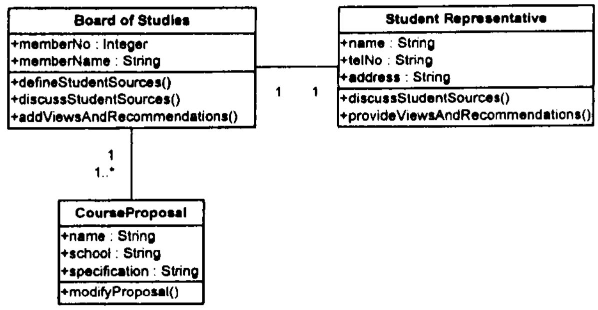



**Figure A- 21: A class diagram for the organisation view of a 'define sources of students' use case**

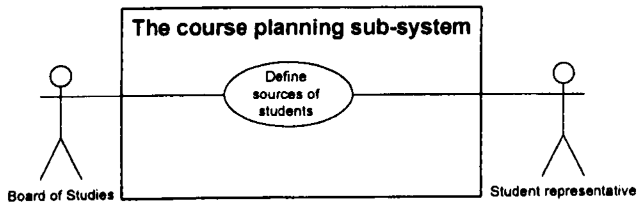
All these diagrams can be combined on one diagram to show all the perspectives of the modelled use case.

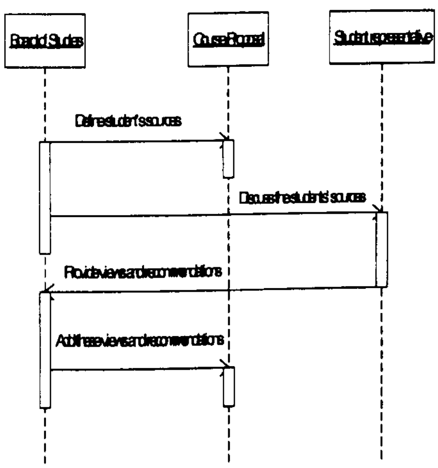



A class diagram for the organisation view of a 'define sources of students ' use case

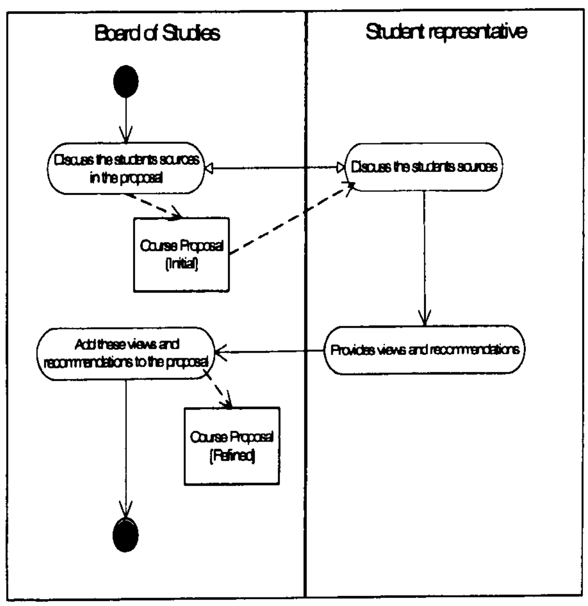



A class diagram for a 'define sources of students' use case



A sequence diagram for a 'define sources of students' use case




An activity diagram for a 'define sources of students' use case

**Figure A- 22: All diagrams of a 'define sources of students' use case**

D) The ' define potential careers for students' use case

The use case is analysed by specifying its detail.

i. Name: define potential careers for students

ii. Brief description

This use case defines the potential careers for student when they graduate from the university through the following steps:

a. Board of Studies (BoS) discusses Initial Course Proposal (ICP) with potential careers.

b. BoS provides views as addendum to recommendations (RICP).

An activity diagram is developed for the use case.

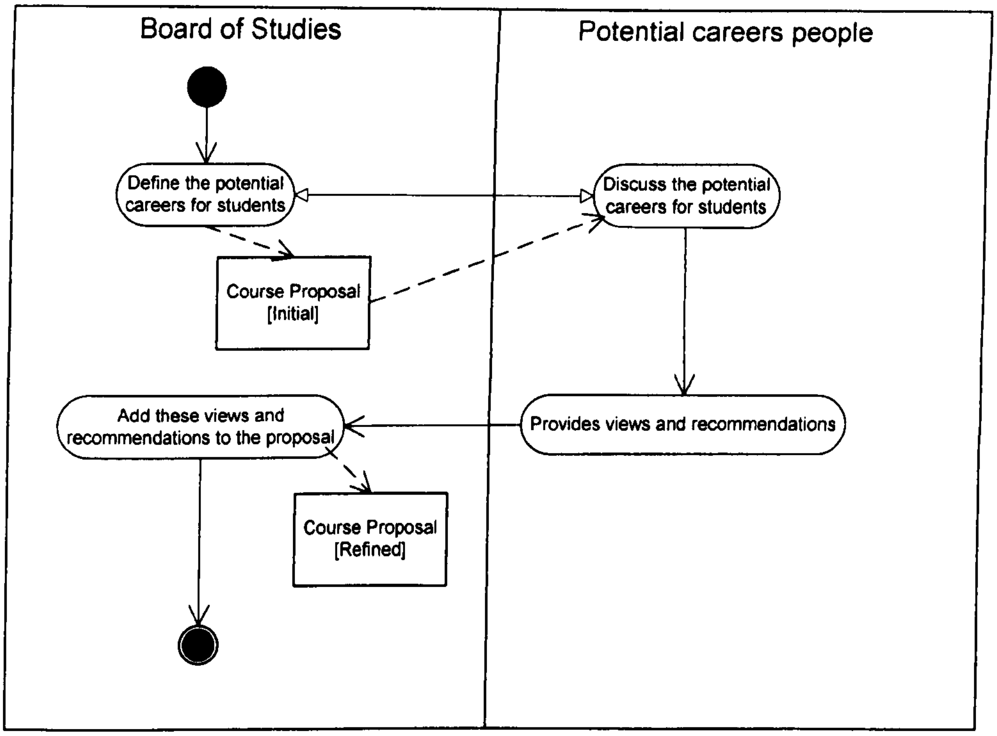



**Figure A- 23: An activity diagram for a 'define potential careers for students' use case**

A sequence diagram is constructed for the use case.

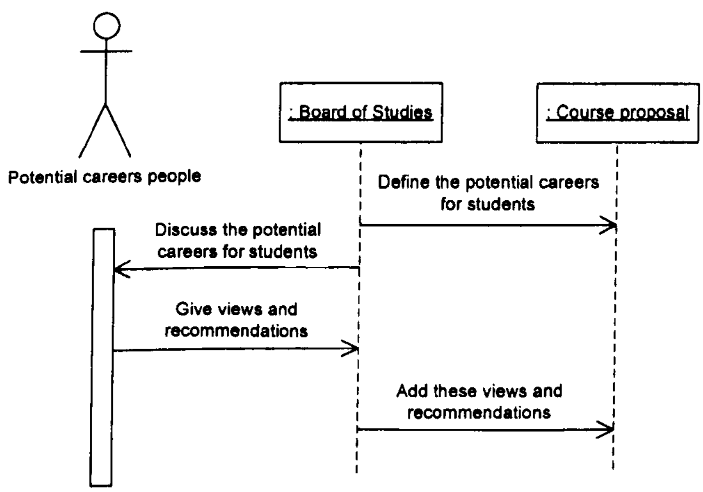



**Figure A- 24: A sequence diagram for a 'define potential careers of students' use case**

A class diagram is formed to present the use case classes that are interacted to perform its functionality.

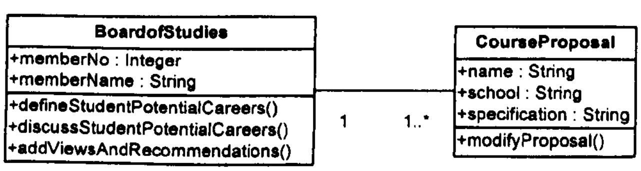



**Figure A- 25: A class diagram for a 'define potential careers for students' use case**

A class diagram is built to present the various roles that are involved in carrying out the functionality of the use case.

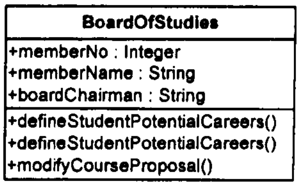
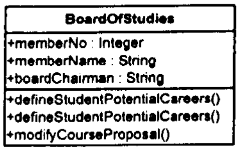
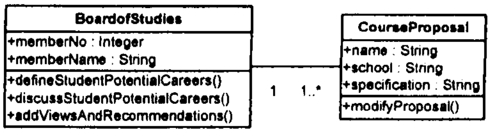



**Figure A- 26: A class diagram for the organisation view of a 'define potential careers for students' use case**

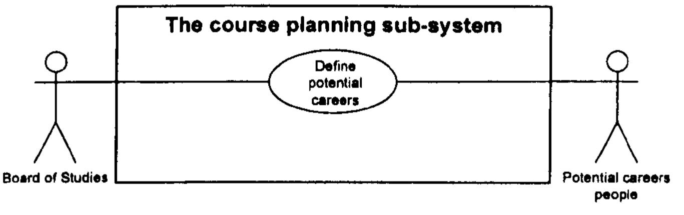
All these diagrams can be combined on one diagram to show all the perspectives of the modelled use case.

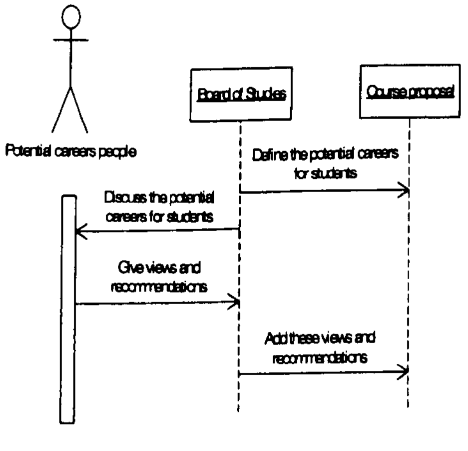



A class diagram for the organisation view of a 'define potential careers for students' use case

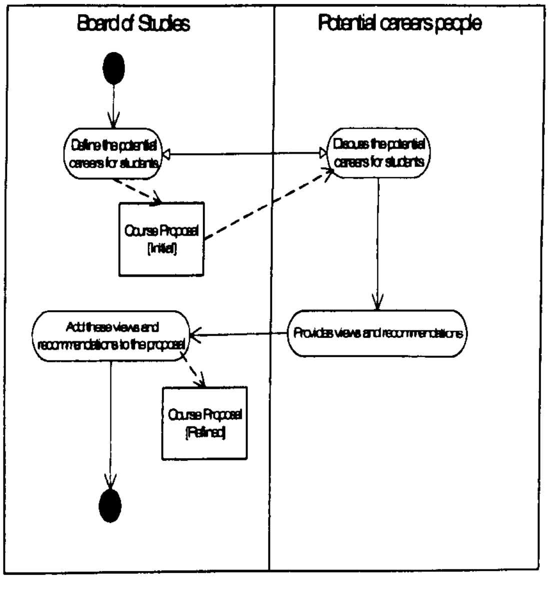



A class diagram for a 'define potential careers for students' use case




A sequence diagram for a 'define potential careers of students' use case




An activity diagram for a 'define potential careers for students' use case

**Figure A- 27: All diagrams for a 'define potential careers for students ' use case**

E) The 'define relevant accreditation and award bodies' use case

The use case is analysed by specifying its detail.

i. Name: define relevant accreditation and award bodies

ii. Brief description

This use case defines the relevant accreditation and award bodies for the course through the following steps:

a. Board of Studies (BoS) identifies for Initial Course Proposal (ICP) accreditation and award bodies.
b. BoS appends provides accreditation and award bodies as addendum to recommendations (RICP).

An activity diagram is developed for the use case.

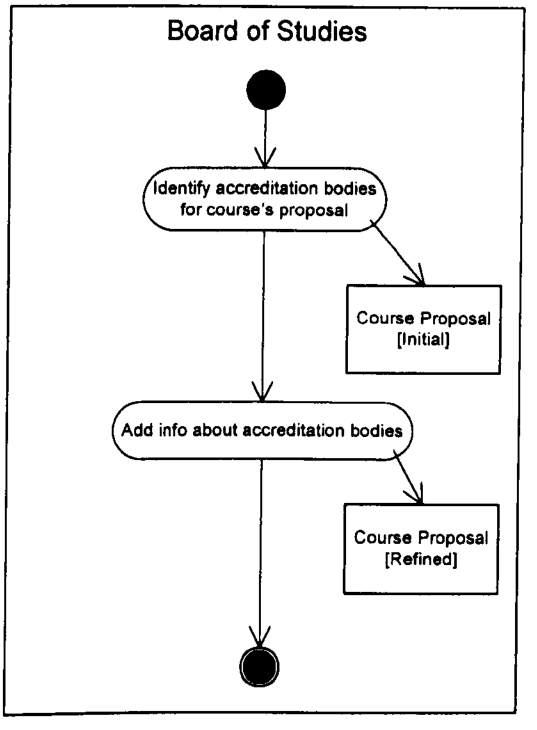



**Figure A- 28: An activity diagram for a 'define accreditation and award bodies' use case**

A sequence diagram is constructed for the use case.

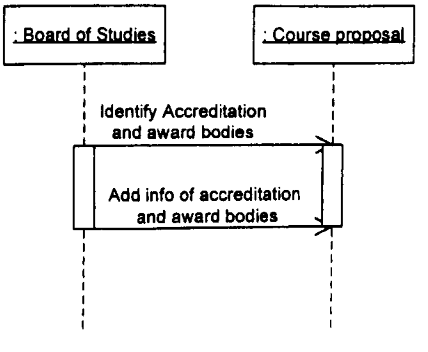



**Figure A- 29: A sequence diagram for a 'define accreditation and award bodies' use case**

A class diagram is formed to present the use case classes that are interacted to perform its functionality.

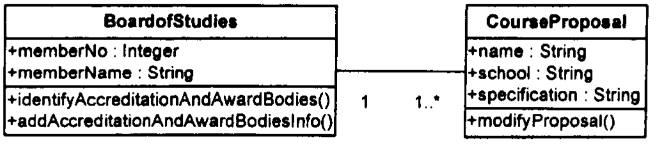



**Figure A- 30: A class diagram for a 'define accreditation and award bodies' use case**

A class diagram is built to present the various roles that are involved in carrying out the functionality of the use case.

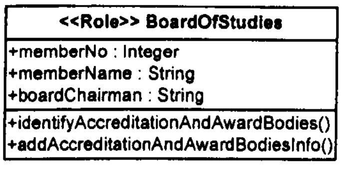
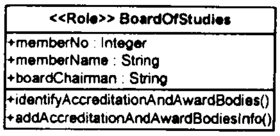
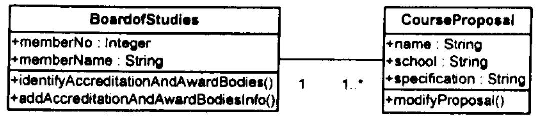



**Figure A- 31: A class diagram for the organisation view of a 'define accreditation and award bodies' use case**

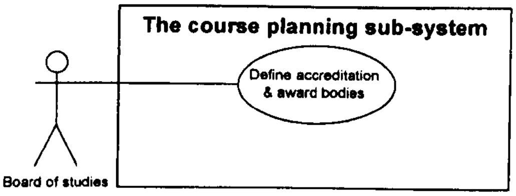
All these diagrams can be combined on one diagram to show all the perspectives of the modelled use case.

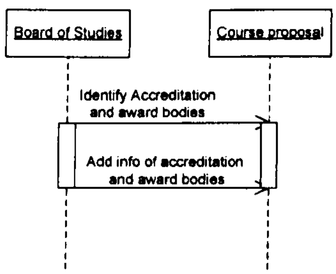



A class diagram for the organisation view of a 'define accreditation and award bodies' use case

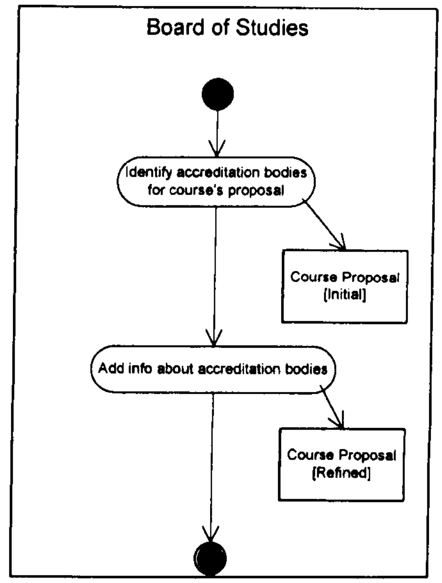



A class diagram for a 'define accreditation and award bodies' use case




A sequence diagram for a 'define accreditation and award bodies' use case




An activity diagram for a 'define accreditation and award bodies' use case

**Figure A- 32: All diagrams for a 'define accreditation and award bodies' use case**

## Step 4.7: Constructing a class model for the business process

An analysis class diagram is constructed from the different class diagrams of the use cases. After modelling each use case and identifying the required classes that collaborate to carry out the behaviour of each use case, these classes of all use cases are combined into one class diagram which is called an analysis model. This model will be used to design the workflow system by adding the design aspects to produce the design model. The diagram (A-33) shows the classes of the 'course planning sub-system'. The classes are arranged by using the generalisation, aggregation and composition relationships. These relationships structure the classes of the model to overcome the model complexity and make them easy to reuse in the later stages of the development. There is a generalisation relationship between the staff class as a superclass with its subclasses lecturer and senior staff. Also, there are two composition relationships between the staff and the Board of Studies and the Board of Studies and the student representative.

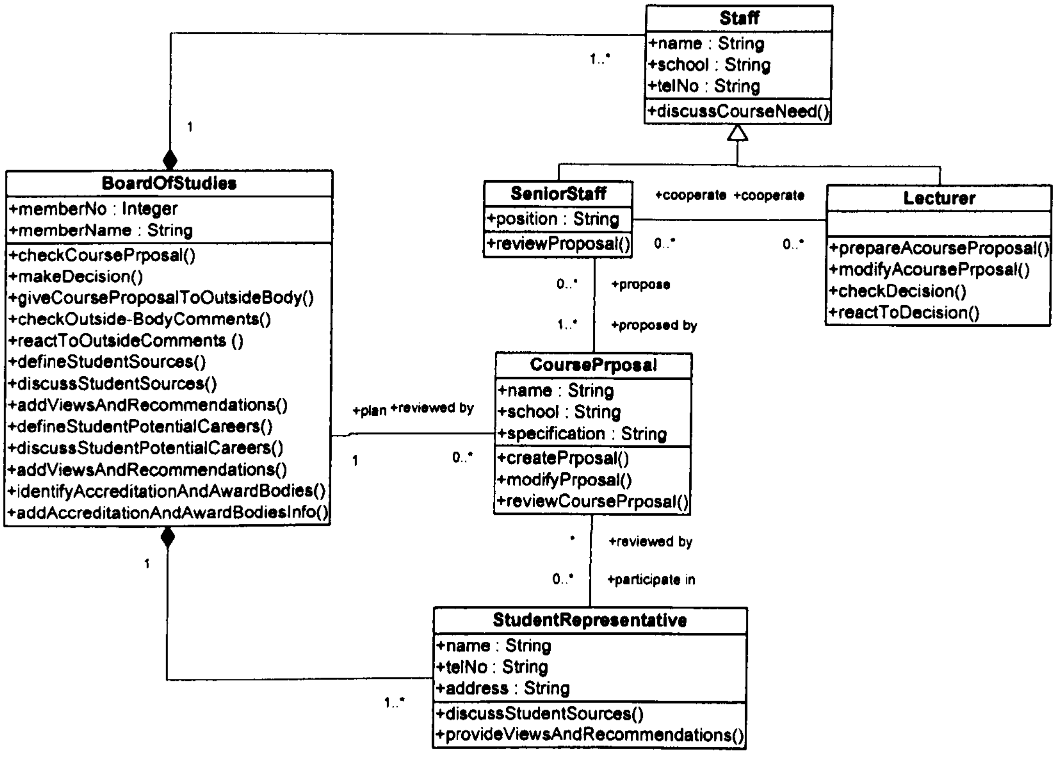



**Figure A- 33: A class diagram for a 'course planning' sub-system**

In addition, it is possible to use the package mechanism to arrange and structure the class diagram of the course management analysis model. A package represents a subsystem so each subsystem of the system is depicted by a package with stereotype <<subsystem>>. Figure (A-34) depicts the course management system with its subsystems that are modelled as packages. The relationship between the course management system and its subsystems is aggregation.

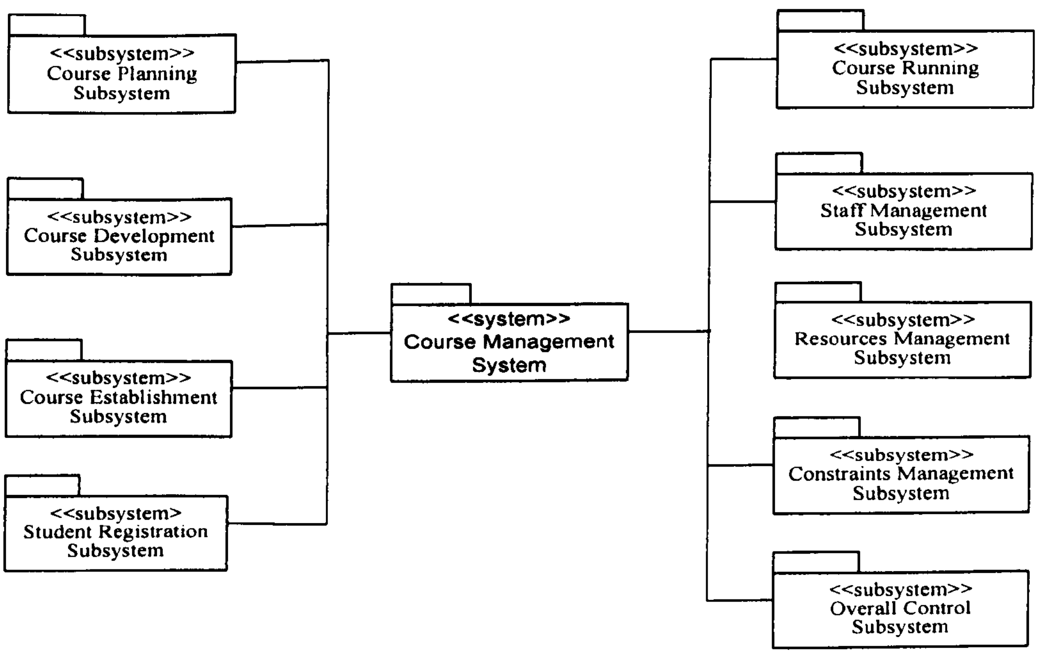



**Figure A- 34: A package diagram for the course management systems and its subsystems**

## Step 4.8: Constructing a class model for the organisation structure

A class diagram for the organisation view is developed from the various class diagrams that represent the organisation view of each use case. The diagram depicts the class diagram for the organisation view of the 'course planning sub-system'. Figure (A-35) shows the overall structure of the organisation perspective of the developed system. The different groups of the organisation are modelled as a class with stereotype <<group>>, and the roles within these groups are modelled as a class with stereotype <<role>>. The relationships among these groups and these groups with their roles are modelled by using the different class diagram relationships such as generalisation and aggregation.

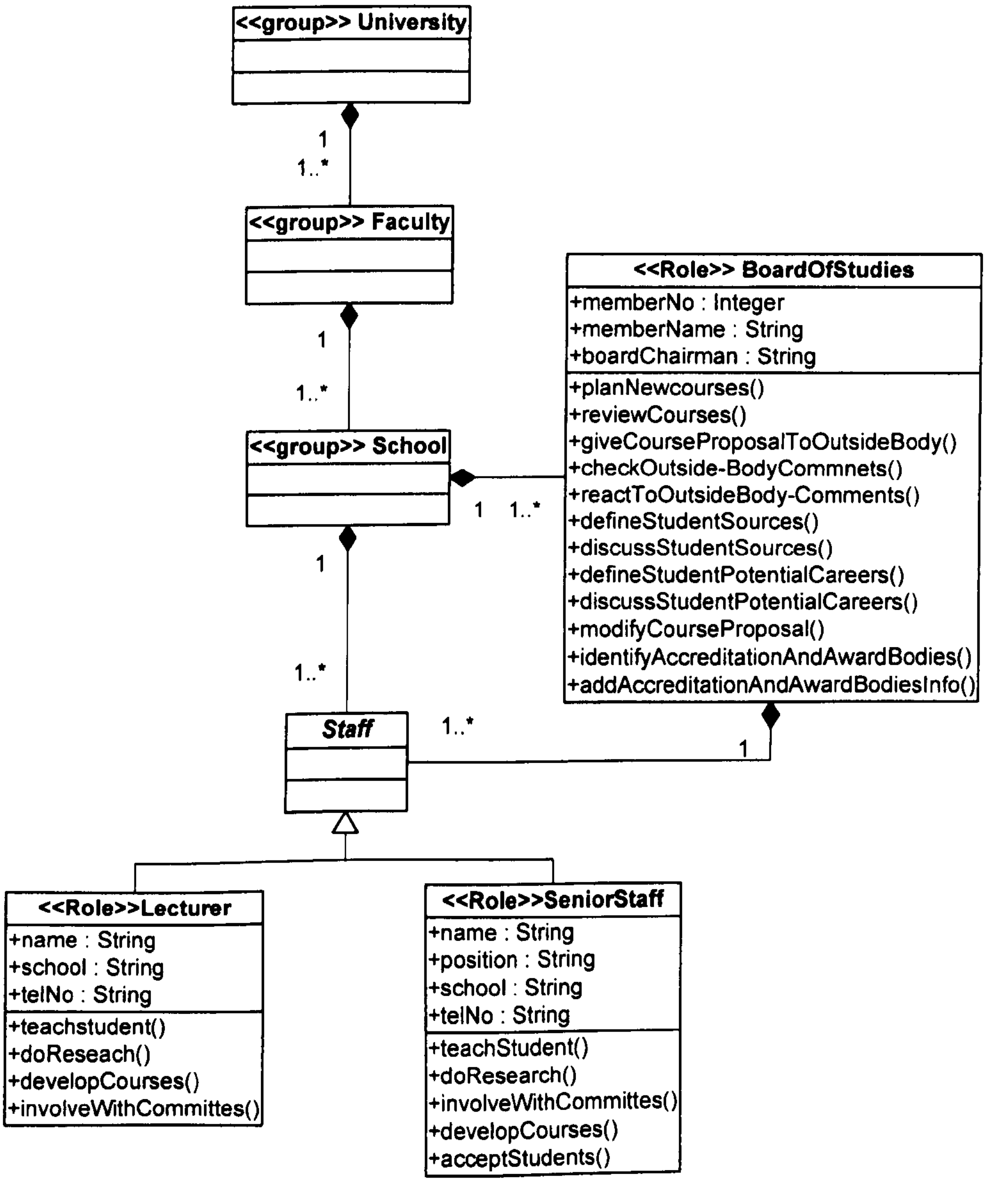



**Figure A- 35: A class diagram for the organisation view of a 'course planning' sub-system**

## Step 4.9: Developing a deployment diagram for the operational perspective

The operational perspective is considered as the implementation of the workflow system operations by using different application programs. The application programs are modelled as components in the nodes of the system. These nodes are connected to each other to show the distribution of the business process operations among the different groups of the organisation. A deployment diagram is developed to depict this perspective. It is not possible to construct the deployment diagram at this stage of the workflow modelling but it should be built after the workflow design stage in which the required application programs are specified.

For the other two sub-systems, the course development and the course establishment sub-systems, the steps of stage two of the SWfM approach can be repeated for every sub-system. The following tables show the sub-system description tables for the two sub-systems.

| Subsystem Number: 2 | Subsystem Name: Course development |
|---|---|
| Subsystem Head: (head of subsystem) | |
| Subsystem Activities:<br>1. Determine a course to be offered<br>2. Plan future course offerings<br>3. Determine expertise requirements for future course offerings<br>4. Decide on a course development | |

**Table A- 10: 'The course development' subsystem**

| Subsystem Number: 3 | Subsystem Name: Course establishment |
|---|---|
| Subsystem Head: (head of subsystem) | |
| Subsystem Activities:<br>1. Establish a course<br>2. Identify course's aims and objectives<br>3. Specify a course's modules<br>4. Specify the required staff<br>5. Specify the other resources<br>6. Determine how to teach students | |

**Table A- 11: 'The course establishment' subsystem**

The following table represents the activity description for every activity in the other two sub-systems.

| No. | Sub-system Name | Activity | Input & its source | Procedure steps | Output & its destination or target | User or agent | Customer or beneficiary |
|---|---|---|---|---|---|---|---|
| II) | The course development sub-system | | | | | | |
| 1) | | Determine a course to be offered | RICPA | a) BoS at second reviews RICPA with addendum.<br>b) BoS decides | VCP | BoS | |

| No. | Sub-system Name | Activity | Input & its source | Procedure steps | Output & its destination or target | User or agent | Customer or beneficiary |
|---|---|---|---|---|---|---|---|
| | | | | whether course should run.<br>c) BoS issues final decision in VCP | | | |
| 2) | | Plan future course offerings | | | | BoS | |
| 3) | | Determine expertise requirements for future course offerings | VCP | a) SS review skills needed<br>b) SS compiles skills needed for future courses<br>c) SS adds compilation to VCP | VCPA | SS | |
| 4) | | Decide on a course development | VCP | a) BoS receives details of course<br>b) BoS reviews proposal<br>c) BoS appends review to VCP | VCPA | BoS | |
| III) | The course establishment sub-system | | | | | | |
| 1) | | Establish a course | VCPA | a) BoS receives final specification<br>b) BoS decision to | AC | BoS | |

| No. | Sub-system Name | Activity | Input & its source | Procedure steps | Output & its destination or target | User or agent | Customer or beneficiary |
|---|---|---|---|---|---|---|---|
| | | | | proceed is recorded | | | |
| 2) | | Identify course's aims and objectives | VCP | a) Lecturer (Appointed Course Leader) proposes aims and objectives in paper<br>b) BoS discusses proposals<br>c) BoS appends aims and objectives to VCP | VCPA | BoS | |
| 3) | | Specify a course's modules | VCP | a) Lecturer (Appointed Course Leader) proposes modules in paper<br>b) BoS discusses proposals<br>c) BoS appends modules to VCP | VCPA | BoS | |
| 4) | | Specify the required staff | VCP | a) SS examines VCP to see staff requirement s.<br>b) SS discusses with staff their loads and interests.<br>c) SS adds staff notes | VPCA | SS | |

| No. | Sub-system Name | Activity | Input & its source | Procedure steps | Output & its destination or target | User or agent | Customer or beneficiary |
|---|---|---|---|---|---|---|---|
| | | | | to VCP as addendum | | | |
| 5) | | Specify the other resources | VCP | a) SS discusses needs with bursar and technicians<br>b) SS attaches views and options to VCP | VCPA | SS | |
| 6) | | Determine how to teach students | VCP | a) BoS asks Lecturer (Appointed Course Leader) for document on options<br>b) BoS reviews this document<br>c) BoS attaches preferred options to VCP | VCPA | BoS | |

**Table A- 12: The two sub-systems activities description**

# Appendix B: Tool Support

In this appendix the CASE tool history and its business view are discussed. Also, many drawing packages are reviewed in the context of choosing the suitable software for modelling the organisational business process.

## B.1 Introduction

A CASE tool can play a central role in an organisation through assisting it to achieve strategic objectives and competitive advantage over other organisations by faster developing and improving the quality of the information system that contribute to these objectives through the planning parts of the integrated CASE tool [207]. The CASE tool can be used to develop large information systems that have complex structures and consist of several parts and elements by implementing an information system development process that involves a lot of rules and tasks which are carried out by techniques. A tool should support the performing of these tasks by offering the required techniques, managing and controlling these tasks, and storing and organising their deliverables and documentations in its repository.

In addition, the CASE tool can aid in improving the quality of the system through providing diagrammatic presentation to model the system requirements, validating the developed diagrams automatically by following the used methodology and techniques rules and maintaining the consistency among the different stages of the development process by facilitating the output of each stage to the following stage. This will help the development team to focus on the early stages of the development process to capture the requirements of the system correctly for eliminating the problems that may be occurred in the late stages where the modification cost is more expensive than earlier stages and enforcing the developers or the users to adhere to the methodology conventions and standards which ensure consistency in the system development [43].

## B.2 The CASE Tool Structure

The CASE tool consists of several layers that communicate with each other to carry out their functionality. They are as follow:

### B.2.1 Operating Systems

The operating system enables the computer to run the various applications including the CASE tools. It controls the interaction between the applications and the system hardware. The operating system should be efficient, reliable and maintainable, and it uses less computer resource. It is important to consider the operating system under which the CASE tool will run. An operating system is linked to hardware that controls the type of operating system to be used. UNIX operating system is considered as open system environment that can be used to run the various CASE tools on mainframe, mini-computer and workstations. On the other hand, for the personal computer (PC) there are several operating systems such as Microsoft Windows, IBM-OS/2 [161].

### B.2.2 Database

An underlying database is considered as a key part of a CASE tool. Workbenches have their proprietary databases while the ICASE environments include several tools which share information tend to have more sophisticated databases. Open environments that may integrate third party tools should include sophisticated flexible databases. The CASE tool database is used to store information that is produced by the tool to assist the system development method and that is produced as application database which is the data that the application program will operate on in everyday use. The stored data in the database can be manipulated by using a database query language (SQL) that comes as part of the database [161].

Also, database is called as a repository that contains all information about the developed system and enables the integration of all models, definitions and mapping of stages. The repository offers a central catalogue for all aspects of an information system development project. An active repository contains information about the rules of the techniques or the development methodology that can be applied which allows analysis, validation, consistency and completeness checking. These rules can be embedded into the repository manager, hard-coded into the repository, or defined in expert systems language which makes them more flexible and easily modified [43].

### B.2.3 Tool Interface

Tool interfaces are the key components that enable third-party tools to be integrated to open environments. There are some initiative standards to define the Public Tools Interfaces (PTIs). For example, the Portable Common Tools Environment (PCTE) is a European-funded activity, defines the architecture for the development, integration and executing of the tools. The aim of PTI is to shield the tools from dealing with the operating system which promotes portability [161].

### B.2.4 Toolsset

The toolsset consists of various tools, each dealing with a specific function or role of the CASE tool functionality such as:

#### B.2.4.1 Project Management Support

It is used to manage and control the various resources of the system development project. This is done by storing all project-related information in a database that is shared by all the project members.

#### B.2.4.2 Models:

They help to model the required system. One of the CASE tool functionalities is to draw the diagrams of the developed system. It is practical to use the tool due to the constant changes to modify the various diagrams. The tool should support the different diagrams and the syntax of the used methodology.

#### B.2.4.3 Editors

It is used to develop the various models of the required method and attach that with a textual description that explains each model.

#### B.2.4.4 Reports

It helps to document the developed system and to produce any report about its deliverables from the tool repository.

#### B.2.4.5 Analysis

Analysis involves two techniques: static and dynamic analysis. Static analysis aims to analyse the various models by applying the technique rules to find any error in the model before proceeding to the next step or stage of the system development and to analyse the source code by testing the flow of data through a system and produce statistics on complexity and code usage. On the other hand, dynamic analysis is used to test the system under controlled situations by using predefined testing data [161].

#### B.2.4.6 Consistency Checkers

It helps to verify and validate the model by applying the syntax checking on the diagrams, consistency checking between diagram and the semantics of the used methodology. It also should contain syntax checking for the models, explanations of possible errors and suggestions for their correctness [208].

#### B.2.4.7 Code and Database Generators

It helps to generate code through converting the models into code stubs for a specific language or to do reverse engineering through converting the code into models.

#### B.2.4.8 Prototype Tools

These tools are used to build a small working system from the specifications to validate the system requirements with the users.

#### B.2.4.9 Configuration Tools

These tools are used to construct a working system from the various developed parts.

#### B.2.4.10 Version Control

It is used to keep track about the different versions of the developed element in the project. It should be able to coordinate the deliverables of the different developers who work in parallel way and make them consistent. It should also provide version management in which a large numbers of different versions of the system are managed [43].

### B.2.5 Graphic Screen Interfaces

The graphical interface is considered as a central part of the CASE tool. Most of the simple tools and workbenches use the WIMP (Window, Icons, Mice, Pull down menus) technology to develop their interfaces. If the tool will be integrated to an open environment, it is important to assess the integration capabilities of the tool interface

[161]. It should include a user-friendly interface that helps the developer to create the models and validate them [208].

## B.3 The Business View of the CASE Tool

### B.3.1 CASE Tool Acquisition

The acquisition of the CASE tool is a difficult and complicated decision that involves investment of a part of the organisation resources so it is important to consider several issues.

The most important aspect is to define the requirements of the tool to specify the required investment to buy a drawing package or a more sophisticated CASE tool that supports all the project stages [209].

Another important aspect is identifying the supported methodology that the tool supports. It is important to check the methodology rules that are enforced in the tool because they may differ from the ones that are used in the organisation or they may be interpreted differently by the tool vendor. Vessey [165] uses the methodology companion term to describe the support of a CASE tool to structured methodology by following its steps and rules. They define three types of philosophies for a methodology support that a CASE tool can provide namely, restrictive, guided and flexible philosophies. The restrictive philosophy means that the CASE tool forces the users to follow the methodology steps and rules. In the guided philosophy, the user may be encouraged to use the CASE tool as a guide in implementing the methodology steps and rules. Finally, the user in the flexible philosophy has a complete freedom to use the CASE tool and he may use his ad-hoc process.

A further aspect is the learning curve for the tool which means the period that the developer or the user takes to use the tool in a productive manner. This period is estimated between six months to two years [43].

The final aspect is the total cost of the tool. This includes the initial outlay of the tool and its maintenance. Also, it is important to consider the required software packages that integrate with the tool. In addition, it is essential to consider the support that the vendor of the tool can provide in terms of upgrades to versions, documentation and on-line or telephone-help [209].

### B.3.2 CASE Tool Characteristics

The CASE tool should offer several characteristics that assist in developing and managing an information system. One of these characteristics is the use of the CASE tool. The CASE tool should be easy to use through its interface that enables the user to interact with it. With the widespread use of the PC, the graphical interface is the common interface that assists the user to carry out his task by interacting through the use of WIMP (Windows, Icons, Mice, Pull-down menus) technology and 'look and feel' approach. There are two types of further interface environments: open CASE environment with well-defined interface that assist other third part tools to be integrated in them and closed CASE environment with interface functions that are embedded in the code that is available to their developers only, which make the integration with third party tools difficult if not impossible. The analyst uses the graphical interface of the tool to develop diagrams that depict his thinking about the system specialisations and requirements [167]. Various diagrams are produced depending on the method used.

The second characteristic is the method or methodology which is supported by the CASE tool. Some CASE tools support one or several system development methods or methodologies. They include support for the various techniques of the methods by providing a diagramming facilities to develop these techniques and enforcing the method rules to use this technique and ensure consistency between the various techniques. It is essential that the CASE tools have knowledge-base system that contains all the required information to support the various stages of an information system method or methodology [167]. Sometimes, the CASE tool developers interpret the method rules differently resulting in different tools for supporting the same method. As Mair [161] stated that 'the CASE developers have worked through the text books and references manuals and built their rule base'. So it is important to check if the CASE tool has adhered to the method that is being used by the enterprise through implementing it on a pilot project.

Another essential aspect of the CASE tool is its repository which is a key component of the CASE tool architecture. The repository is a database system that is used to store all data and information relating to a project such as the requirements specification, models, design documentation, code, test results, project management information and so on. The repository can be rudimentary with pop-up screen formats to complete which are linked to the diagram [167].

Training is an important issue in acquiring a CASE tool. The vendor of a CASE tool should provide training to the users. If the tool implements a specific method, the vendor should offer training on both the tool and the method. The user training assists to avoid a lot of problems that are associated with introducing the tool in the organisation such as the resistance to use because the users do not like to change the way that they develop new systems due to inadequate knowledge of the new tool capabilities.

### B.3.3 The Introduction of a CASE Tool into an Organisation

The CASE tool should be chosen depending on the organisation requirements. There are several approaches to introduce the CASE tool into an organisation. First, the tool is introduced in the organisation without any consideration of the organisation requirements and role of the tool. In the second approach, the tool is presented in a slow but deliberate way. Finally there is the active way in which the tool is acquired and put for use in a short time. This approach requires organisational and managerial support. Iivari [210] in his study about the CASE tool usage reported that management can positively influence the use of the tool by factors such as participation, training and expectation realism.

There are three possible effects of introducing a tool into an organisation. First, there is no effect or change due to the tool being compatible with the methodology used in the organisation. Second there is incremental change where the tool introduces small change to current working practices. Finally, the change is radical if the tool brings major change and new experiences into the organisation.

Sommerville [166] mentions that the CASE tool goes through six stages from procurement of the CASE tool until it becomes obsolete. These stages are procurement, tailoring, introduction, operation, evolution and obsolescence.

In the procurement stage, the CASE tool is selected depending on several factors such as the method used in the organisation, the existing hardware, the required developing systems, and the required security facility. It is important to consult the CASE tool users about their requirements and ask them to participate in selection of the suitable tool.

The CASE tool in the tailoring stage is customised to the working practices and the specific method that are used in the organisation. The CASE tool customisation includes installing the CASE tool and testing it on the organisation's hardware, defining a process model for the CASE tool, integrating the CASE tool with the existing tools and documenting the customisation information.

The introduction and operation of the CASE tool in the organisation involve changing the working practices of the organisation, and solving the problems associated with introducing the CASE tool such as user resistance, lack of training and management resistance. Regarding the CASE tool users, it may affect the developers who will use it to develop a system and the users of the developed system. For developers, there are two cases. The first one is when the CASE tool will be introduced with a new methodology which requires learning both the tool and the methodology. This will result in some difficulties to adopt and use the CASE tool. The other case is that the CASE tool will be considered as a control tool to monitor the work of the developers and enforce certain rules. This will result in rejection of the tool. For the users, their perceptions of the tool depend on the benefits that they will obtain by using the tool such as better quality systems, corrected identification of their requirements or faster production of systems. If the users get more advantages by using the tool, they will accept and use it. So it is important to consider the human factor in introducing the CASE tool in an organisation [43]. These problems can be solved by involving the user in the tool selection decision, providing an adequate training to the users through attending some courses and workshops which are offered by the tool vendor and using the tool in a prototype project to discover its capabilities. It should be a supported strategic plan that is defined by the top management to introduce the tool into the organisation.

In the fifth stage, the CASE tool may need to evolve as a result of new hardware and software platforms. The new version of the CASE tool should be compatible with the old versions otherwise the problems that arise due to incompatibility should be resolved by introducing a new hardware for the new version or supporting the various machines with different operating systems by using a heterogeneous network.

The last stage, that is the time when the CASE tool becomes obsolete due to the lack of support offered by the vendor or the changing of the hardware or software platforms. The movement to another CASE tool should go through a transition period in which the software systems that were developed by the old CASE tool are transferred into the new CASE tool.

## B.4 Drawing software

There are several drawing packages or software that can be used to develop a notation of any method or methodology such as Canvas, CorelDraw, Illustrator, Freehand, SmartDraw or Visio.

### B.4.1 Canvas Software

Canvas is a graphical package that is developed by Deneba Software which is a leading developer and publisher of graphics software. It offers vector drawing, diagrammatic facilities, technical illustration, creative drawing, image editing, web graphics and page layout features in one powerful application. It includes the following features such as greater drawing precision, new user interface, new technical drawing features, an expanded imaging editing architecture, more Scripting Options, and more file formats support. Canvas comes in three editions: Canvas professional edition, Canvas GIS (Geographic Information System) mapping editions and Canvas scientific imaging edition. It provides a smart toolbox that tracks and displays all drawing tools that the user can use to develop a drawing. The required tool sets can be resized and locked so they are always available. Also, the tool has a properties bar feature that keeps track of what objects and tools are selected and displays the most relevant options at all times. This will help to save the users' time [211].

### B.4.2 CorelDraw Software

CorelDraw is a powerful, easy-to-use graphics program that is created by Corel Corporation. CorelDraw comes with a suite of programs that serve many functions, but the heart of the program is the Draw module. It delivers graphic design and vector animation software for print and the Web. It has interactive tools that attempt to save time and make the design process easier. It supports the Microsoft's user interface and visual styles. It includes workplace features that assist the user to customise its interface and settings, object handling that allow the user to define objects such as symbols or shapes and control them, toolbox that allows the user to develop a drawing and file import/export capabilities to exchange file with other applications [212].

### B.4.3 Adobe Illustrator

Adobe Illustrator is graphical software that is designed by Adobe Company, is a developer of world-leading digital imaging, design, and document technology platforms for consumers and enterprises. Illustrator has several features that allow the user to design a professional drawing: a graphical user interface assists users in developing a drawing and communicating with other Adobe applications. Designing and drawing tools construct drawings such as toolbox and professionally designed templates. Finally, Illustrator can be integrated with other applications such as other Adobe applications or Microsoft products, web page support, script support for Java script and Visual Basic for application to control its features and file import/export capabilities to exchange file with other software [213].

### B.4.4 Freehand Software

Freehand is a drawing package that is created by Macromedia Company. Freehand includes several features as follow: A common Macromedia user interface allows the user to switch to other Macromedia products. Several tools such as freeform tool, connector line tool, extrude tool, eraser tool, blend tool, and pen tool, are provided to assist the user in developing drawing and editing tools to control the behaviour of the symbols, over spacing, angle offset and scaling. Freehand also is equipped by a library feature which allows transformation of objects into symbols that can be stored and used across the whole project. When editing a library item, the changes will cascade through the whole project. Freehand can be integrated with other applications, for example Dreamweaver to improve its working deficiency with web, Flash software to create movies and Fireworks to enhance the bitmap files editability. It supports file import/export facility [214, 215].

### B.4.5 SmartDraw Software

SmartDraw is a Windows based software drawing package that is developed by SmartDraw.com Company which offers three products: the flagship SmartDraw program for creating business diagrams; SmartDraw Photo for managing digital images; and Visual Script XML for visually developing XML.

SmartDraw software comes in three editions namely, standard, professional and plus professional. The standard version is used for creating basic business charts, diagrams, and presentations. The Standard Collection comes with every version of SmartDraw, containing over 1000 symbols and templates selected from the eleven SmartDraw Symbol Collections which are the basic symbols and templates for most types of drawing and charting. SmartDraw symbol collections are:

i) Business and Charting

ii) Clip Art

iii) Electrical Engineering

iv) Floor Plans and Facilities

v) Life and Leisure

vi) Maps and Geography

vii) Mechanical Engineering

viii) Medical and Anatomy

ix) Network Design

x) Science and Math

xi) Software Design

SmartDraw Professional contains all the standard features and symbols, plus these useful extras, Spelling checker (in 9 languages), advanced import and export filters, full integration with MS Office, plus the Business & Charting collection - with over 1100 additional symbols and templates for flowcharts, org charts, forms and more.

SmartDraw Professional Plus includes all the features of the professional version in addition to more than 500,000 symbols and examples in the eleven different specialized collections [216].