A Co-ordinated Business Object Approach for Supporting Tactical Level Management Decisions

STEVEN ANDREW JOHN BROCKIE

Doctor of Philosophy

ASTON UNIVERSITY

March 2003

ASTON UNIVERSITY

A Co-ordinated Business Object Approach for Supporting Tactical Level Management Decisions

STEVEN ANDREW JOHN BROCKIE

Doctor of Philosophy

2003

The proliferation of data throughout the strategic, tactical and operational areas within many organisations, has provided a need for the decision maker to be presented with structured information that is appropriate for achieving allocated tasks. However, despite this abundance of data, managers at all levels in the organisation commonly encounter a condition of 'information overload', that results in a paucity of the correct information. Specifically, this thesis will focus upon the tactical domain within the organisation and the information needs of management who reside at this level. In doing so, it will argue that the link between decision making at the tactical level in the organisation, and low-level transaction processing data, should be through a common object model that uses a framework based upon knowledge leveraged from co-ordination theory.

In order to achieve this, the Co-ordinated Business Object Model (CBOM) was created. Detailing a two-tier framework, the first tier models data based upon four interactive object models, namely, processes, activities, resources and actors. The second tier analyses the data captured by the four object models, and returns information that can be used to support tactical decision making. In addition, the Co-ordinated Business Object Support System (CBOSS), is a prototype tool that has been developed in order to both support the CBOM implementation, and to also demonstrate the functionality of the CBOM as a modeling approach for supporting tactical management decision making. Containing a graphical user interface, the system's functionality allows the user to create and explore alternative implementations of an identified tactical level process. In order to validate the CBOM, three verification tests have been completed. The results provide evidence that the CBOM framework helps bridge the gap between low level transaction data, and the information that is used to support tactical level decision making.

Keywords
Business Objects; Object Modeling; Co-ordination Theory; Tactical Management; Tactical Management Information; Decision Support.

Aknowledgements

I would like to thank the following people for their help throughout this study.

Firstly, I would like to thank my supervisor Dr Paul Golder for his help, invaluable advice and support throughout.

I should also like to thank John Quimby at the Center for Co-ordination Science (CCS) at The Massachusetts Institute of Technology (MIT), Dr Fred Hewitt from Aston Business School and the MIT Sloan school of management, and Brendan McCarthy from BMW.

Finally, I would like to thank my parents for their help, encouragement and support.

List Of Contents

List Of Figures

List Of Figures

Appendices

List Of Tables

CHAPTER 1

Introduction

1.1. MOTIVATION

A wide range of issues provided the impetus which led to the undertaking of this research. Before starting this study, my days were spent engaged in the development of report-type management information systems. Such systems, were primarily concerned with the searching and sorting of data, and the subsequent returning of that data in the form of meaningful information. This information, generally reflected the information need of the decision maker. However, as is the case with many information systems, and as many authors have observed (Ackoff, 1967; Flowers, 1996; Lyytinen and Hirschein, 1987; Sauer, 1993), these systems, for a variety of reasons fail to be effective, and hence do not support the decision making requirements of management. One such reason maybe owing to their rigidity, as they do not lend themselves to accurately mapping or describing the outside or real world organisation with which they represent. Secondly, a tool that simply returns a report based upon low-level data, does not always produce sufficient information for supporting management decisions, as it does not present sufficiently, a broad or deep enough image of an organisation's operations. It is also clear that such limitations with these systems, cause a distinct lack of 'information integration' within the information supply chain, between an organisation's functional areas. Consequently, managing these different areas becomes a much harder task as the necessary information, in some cases, is not always available.

In order to overcome this, managers use a variety of different types of information systems for supporting their decision making needs. For example, these can be broadly defined as; Transaction Processing Systems (TPS) and Office Automation Systems (OAS), which are typically used for operational level tasks, and whilst Executive Information Systems (EIS) support strategic level information, Decision Support Systems (DSS) and Management Information Systems (MIS), typically support the formulation of tactical or middle management level decisions. At this point it is necessary to outline that the terms 'middle', 'control' and 'tactical' are all used to identify the same level of management, and are therefore interchangeable. In order to avoid ambiguity, this thesis from this point forward will use the term 'tactical', unless otherwise stated.

Finally, throughout my initial readings, it became clear that there is a distinct lack of research that focuses solely upon the tactical level and its corresponding information needs. Therefore, before considering the problem that this thesis addresses in more detail, it is necessary to outline that this thesis is primarily concerned with supporting the information needs of managers who reside at the tactical or middle level within the organisation. Therefore, this research is broadly motivated by the question; "Is there a better way of generating information for supporting tactical management decision making?"

1.2. THE PROBLEM

Ideas posited by authors such as Anthony (1965) and Jaques (1989), suggest that the organisation is comprised of a series of layers. Anthony proposes that the organisation has three definitive layers, namely, operational, tactical and strategic. This is shown in figure 1.1 below.

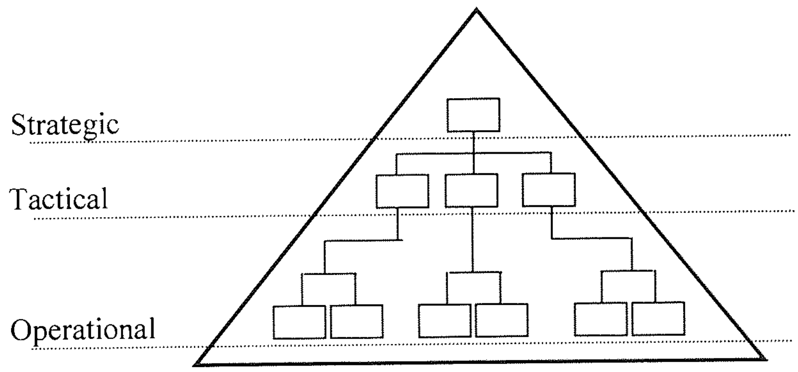

Figure 1.1. The organisational hierarchy, adapted from Anthony (1965)

The information needs of managers who reside and operate at each of these levels are different. Despite, or perhaps because of the wide range of information systems, many organisations suffer from a condition of 'information overload' (Swash, 1998).

In short, this term details an abundant supply of information where only a small amount is considered useful. For managers who operate at the tactical level, this problem is all the more relevant, especially if their control spans the management of many functional areas within the organisation. Therefore, the need to have accurate information about allocated activities at the right time is crucial, as many activities that are assigned to the tactical level manager may execute simultaneously.

Information systems, especially those that make use of object oriented technologies, represent the organisation as comprising of a series of objects, that encapsulate functionality and business logic relating to the organisation's functional areas. The term 'business object' has been used to describe this (Sutherland, 1995). However, although the information system possesses 'knowledge' as to the structure and functionality of the organisation, the problem of deriving the 'correct' or 'right' information is still prevalent. Many managers who have tactical level responsibility, still rely upon report based systems in order to obtain information. More often than not, this information is aggregated in order to meet tactical level decision making requirements. For example, from an operational managers point of view, the following statement could typically describe a workplace issue; "25 parts were not produced in the last hour because employee 235 injured himself". From a tactical management perspective, this same problem may be interpreted as; "Production in the parts department is under its hourly target". Such tactical level information, generally represents the type of information with which tactical management would use to make decisions. In itself, it is static and uncoordinated in nature. For instance, simple facts such as those gathered in the above workplace problem, do not expose to the manager the failure of the despatch department to meet their allocated despatch time. In this case they have to rely upon operational level information. It is this seeming lack of providing the 'relevant' tactical level information, which causes a blinkered view when formulating tactical level decisions.

Such operations rely upon the tactical level manager to interpret this low-level data in order to support the formulation of tactical level decisions. Therefore, although the organisational layers presented by Anthony are specifically separated, the information used by management who reside at the tactical level, arguably is not. In short, the layers loose their opaqueness as the tactical level manager is not supplied with tactical type information, but has to formulate tactical level decisions based upon aggregated operational information, which is generally in the form of management reports.

This study will argue that tactical decision making should be supported by tactical level information. In doing so, it will suggest that this information should be produced through a common object model, that is specifically centred around the modelling of tactical level processes.

1.3. CURRENT APPROACHES TO THE PROBLEM

Throughout the course of this research a fairly extensive search of the research literature was carried out. It was important to conduct such a survey in a number of areas, in order to ascertain what approaches if any have been made for addressing the problem outlined in section 1.2. Throughout the data gathering stage, a number of references pointing to advances in three main areas were uncovered. Namely;

- Management structures.
- Data modelling approaches.
- Integrative technologies.

This section discusses briefly, recognised approaches that may be considered useful for providing a solution to the problem area stated above. These opinions derive not only from different areas of computer science, but also from within the area of business and management studies. This thesis will investigate these areas for two main reasons. Firstly, owing to their close relation to the problem area stated above, and secondly because these areas figure prominently in the academic literature. It should be understood that on occasions throughout this research period, it may be necessary to investigate other areas that are closely related to the three main areas listed above. References to these closely related sub-branches will be made where necessary.

1.3.1. Management Structures

Various embellishments to pre-existing management structures, have in the past been used as a basis for improving the information flow throughout the various management levels. Most notably, flatter management structures that reduce the amount of hierarchical layers within the organisation, place the emphasis on using semiautonomous workgroups for managing the organisation's functional areas (Kettley, 1995; Jaques, 1989). The use of flatter structures in aiding the tactical decision making process, can be largely attributed to the apparent ease with which information flows throughout the various levels. Theoretically, this is as a result of a significant reduction in the amount of hierarchical layers.

1.3.2. Data Modelling

Advances in the field of data modelling suggest that the design of an information system should accurately reflect the real world domain that it represents. Various approaches for detailing this have been outlined in conventional data models such as the 'Relational Model' (Codd, 1970), and the 'Entity Relationship Model' (Chen, 1976), as well as various object oriented approaches (Rumbaugh et al, 1991; Sutherland, 1995). In addition to this, the notion of 'business objects' as opposed to conventional objects, for explaining a modelling approach for reflecting the organisation's business functionality, is becoming increasingly popular as a concept used in the developing of information systems (Sutherland, 1995). Consequently, by designing the information system so that it more closely reflects the real world business paradigm, will theoretically realise an information system that delivers more accurate information to the decision making process. Furthermore, the business oriented approach to modelling information systems, reflects both the necessity to capture static data regarding an organisation's operations, as well as the more dynamic aspects such as organisational processes.

1.3.3. Integrative Technologies

Where the business object approach is primarily concerned with the design of the information system, collaborative technologies are more concerned with the overall communication and functioning of information systems. Most notably, the Distributed Component Object Model (DCOM) by Microsoft, and the Common Object Request Broker Architecture (CORBA) by the Object Management Group

(OMG), are amongst the leading technologies that allow for the creating, distributing and managing of distributed program objects in a network. In essence, they allow different systems that are placed at different parts of the network to communicate with each other via an interface broker. Such techniques are designed to integrate various information systems throughout the various platforms, hence provide all levels of management with more detailed information.

These three subject areas described above, detail relevant approaches from three distinct fields. These fields can be identified as the main areas where the relevant investigations will be conducted, in order to gather evidence in support of identifying a solution to the problem stated in section 1.2 above. The specific areas that are outlined as comprising the research domain with which this thesis will target, are listed below.

1.4. THE RESEARCH DOMAIN

This thesis will examine academic literature from two research domains, namely 'Tactical Management' and 'Information Systems'. Figure 1.2 provides an initial snapshot of these research areas along with subject areas that form possible sub-branches to the main themes. These domains have been chosen for various reasons. Firstly, this thesis as the abstract outlined, is concerned with supporting the decision making capabilities of tactical level management. Such support is generally delivered by information that is fit for the specific management need. Therefore, by focusing upon the information requirements of tactical level management, will provide the reader with a thorough understanding of what tactical level information is and how it can be used to support the tactical decision making process. Secondly, as the method used to support the decision making capabilities of not only tactical level management, but all management at all levels in the organisation is through computer based information systems, it seems appropriate to conduct a period of research in these areas. The author recognises however, that some managers nowadays may be supported with traditional paper based system, however, this study is not concerned with such methods and for clarity will narrow its specific focus to concentrating of those that are computer based in nature.

It would perhaps be more accurate to suggest that these domains are starting points from which the research process will naturally flow. In essence, it is assumed that throughout the research phase it will be necessary to branch out and investigate closely related areas where they can provide appropriate supporting material or evidence when required. Thus, providing enough material in order to develop an appropriate solution to the problem area outlined in section 1.2.

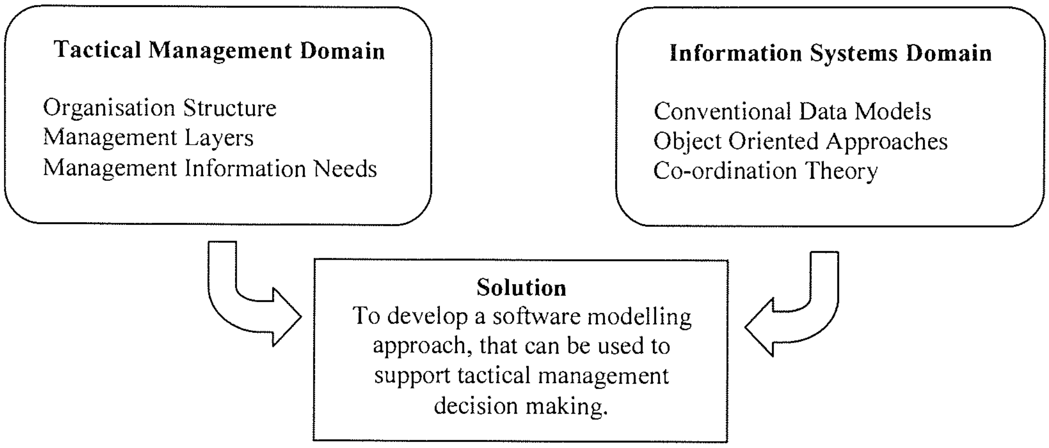

Figure 1.2. The solution and the relevant research areas

The two main areas listed in the diagram above, will be considered in more detail throughout the remainder of this section.

1.4.1 Tactical Management Domain

The three areas for analysis and investigation in this domain are:

- Organisational Structure
- Management Layers
- Management Information Needs

It should be noted that the list above is by no means exhausted. Indeed, it is expected that as the research progresses, so additions to the list may become apparent. However, it can be recognised that these are the salient parts of the tactical management domain that this research will focus upon.

Many types of organisations tend to operate using either a vertical (hierarchical) or horizontal (flat) structure of management. Other structures such as peer groups or matrix structures fall into this depiction, as residing somewhere between that of a vertical or horizontal structure. From the perspective of this thesis, the aim is not to simply assess all known permutations of organisational structures, along with the relevant behaviour of their corresponding managers, but to recognise the existence of tactical management tasks, regardless of the structure. Current literature in the field of management studies, abounds with ideas of flatter styles of management (Kettley, 1995). In contrast, evidence that a hierarchical style of leadership is still very much a common place in the modern organisation can also be recognised (Jaques, 1989). Despite these contrasting views, it has become clear that these different areas must be taken into account, if an accurate investigation into the production of tactical management information is to be achieved.

In order to fully understand the information needs of tactical management, this thesis shall investigate the nature and role of tactical level information. In addition, a brief investigation into the role and nature of both strategic and operational level information will be conducted. The purpose of this is two-fold. Firstly, tactical management are required to implement strategic level objectives. Secondly, management who reside at the tactical level, traditionally set operational level objectives. Therefore, objectives pass from the strategic to the tactical level, and information regarding the effectiveness of these objectives pass from the tactical level to the strategic level. From this perspective, in order to provide a sufficient understanding of tactical level information, it is deemed necessary to investigate all of the management levels that interact with the tactical level.

1.4.2 Information Systems Domain

Three main subject areas have been identified for investigation under this domain. These are:

- Conventional Data Models
- Object Oriented Approaches
- Co-ordination Theory

Ideas taken from *data modelling,* suggest that the architecture of an information system should reflect the real world domain (Codd, 1970; Chen, 1976; Rumbaugh et al, 1991). From this point of view, it is necessary to investigate the various data modelling techniques that are used to model information systems. Therefore, conventional data models such as the 'E-R model' and the 'Relational model', as well as established object oriented approaches, shall be considered throughout the course of this study.

By themselves, these approaches are primarily concerned with modelling entities / objects and their behaviour. In order to provide a more robust modelling method that meets the needs of providing tactical level information, this thesis will consider the role of co-ordination theory. Suggested by Malone and Crowston (1994), the notion of co-ordinating organisational tasks by identifying activities and the associated dependencies that link them (Malone et al, 1999), provides managers with the opportunity to not only focus upon their span of control within the organisation, but also to monitor resources as they are either consumed or produced. In addition, by using a system of co-ordination that allows tactical management to focus upon organisational processes, will provide a richer data model for returning appropriate information concerning their functional responsibilities. Although many processes belong to higher level or more abstract processes, they can also in contrast be decomposed into smaller sub-processes or activities. For example, a tactical type process 'Make Product' may consist of many sub-processes at the operational level, i.e. 'Order Stock', 'Assemble Product' then 'Despatch Product'. The application of the co-ordination framework to tactical level processes, highlights the relationships one activity may have with another, as well as the resources that are required for use by these activities. In short, co-ordination theory provides a framework for analysing organisational processes. As will be made clear throughout this study, the constructs contained within the co-ordination approach, can be used within an object model for supporting tactical management information.

For clarity, it should be noted that various approaches have been proposed that outline the mapping of organisational processes and that the co-ordination approach is not the only method. For example, the use of Business Process Modelling Tools (BPMT), allow for the representation and design of organisational processes. However, by themselves these tools provide a description of a process and hence do not provide any information regarding the operational capacity of the process, i.e.

when the process is live. Consequently, such tools do not provide support for the facility to track live changes as they occur. Workflow systems are also based around allowing the user to develop a particular organisational process that can be implemented into the organisation's operational procedures. However, these operate at a very low-level, i.e. at the operational level within the organisation and therefore, the level of abstraction with which they represent organisational processes can be considered to be to low for tactical level management. For example, workflow system may record employee movements between machines, or the unproductive repetitive nature of some inefficient tasks, all of which would tend to provide information that would be more use to a operational level manger than a tactical level manager. The Business Object approach on the other hand uses the notion of modelling processes within an information systems domain. Such systems in contrast to BPMT's do provide information as to changes in the process. Therefore, they represent dynamic information as to the changing state of the process over a period of time.

In concluding this section, it can be regarded that the co-ordination perspective provides a framework for analysing organisational processes. In short, it is an approach that in the context of this study, can be applied to an information systems model in an attempt to provide the user with co-ordinated information based upon tactical level processes, by further extending the constructs associated to the business object model. The co-ordination approach will therefore be analysed in much further detail throughout this section. This is for two reasons. Firstly, other approaches outlined above do not possess the functionality to address the problem area stated in section 1.2. Secondly, as this thesis will focus upon the supporting of tactical decision making within an information systems framework, it will be necessary to explore the possibility of a modelling approach that is new and provides a novel contribution to the academic literature.

1.5. THE RESEARCH OBJECTIVES

The research objectives that have been defined in order to address the problem area stated in section 1.2 above, will be considered in further detail in this section. The overarching aim of this thesis, acts as the yardstick by which the research objectives are formed. This aim was broadly described in section 1.1 above as;

> ***"Is there a better way of generating information for supporting tactical management decision making?"***

Perhaps at this point in the chapter, and before considering the objectives, it is necessary to examine the aim of this thesis in greater detail. In essence, the aim acts as a broad guideline as to the overarching question the thesis aims to answer. Here the aim outlines a 'better way' of generating information that can be used for supporting tactical level decision making. In making this assertion, the word 'better' is used to highlight two principal meanings. Firstly, it is used to describe a more suitable way of supporting tactical level decision making as opposed to other approaches outlined in section 1.4 above. Secondly, it is used to underline more efficient ways of generating tactical level information, based upon tactical level processes. Therefore, in order to provide an accurate assessment of tactical level tasks and information needs, the objectives listed below will aim to provide a broad review of the role of tactical management as well as the environment in which tactical management operate. These objectives will specifically be covered in objective one and two below. By outlining these areas, will provide the reader with an understanding as to the parameters of tactical management as well as providing a platform with which to build further objectives upon. In order to achieve the aim stated above, the following smaller objectives can be recognised as:

OBJECTIVE 1 - ***To explore the tactical management role in the organisational structure. This will be achieved by;***

OBJECTIVE 1.1 - ***Investigating the hierarchical structure of organisations.***

The popular view of the organisation details a structure that is split into hierarchical levels (Anthony, 1965; Jaques, 1989). In modern times however, more 'flatter' or 'horizontal' systems of organisational structure, have began to emerge (Kettley, 1995). This shifts the emphasis from a rigid hierarchical form, to one that has less organisational levels. In short, the flatter structure is diametrically opposed to the hierarchical structure. Therefore, the aim of this research objective is to provide an investigation into the structure of organisations in which a tactical level manager would operate.

OBJECTIVE 1.2 - ***Exploring the tactical management role in the organisational structure.***

In order to explore this in detail, it will be necessary to outline not only the role of the tactical manager, but to also highlight the tactical managers relationship with both the strategic and operational levels within an organisation. Therefore, as the first objective will investigate the dimensions of the hierarchical structure, the purpose of this research objective will be to assess the role of the tactical manager in the organisation.

OBJECTIVE 2 - ***To investigate the information needs of tactical management.***

Specifically, this research objective will focus upon the information needs of tactical management. This will be carried out in order to make it clear to the user what exactly constitutes tactical level information. In outlining this, it is important to outline what distinguishes tactical level information from both strategic and operational level information (Brockie & Golder, 2000, 2001). Moreover, as these levels interact with each other, in so much as they pass information and objectives between themselves (Ahituv & Neumann, 1990), (Mclean & Soden, 1977), (Wetherbe, 1994), an investigation of specific information needs of tactical level management will not only outline the scope of such information, but also its characteristics, uses corresponding level of abstraction, and its relevant applications.

OBJECTIVE 3 - ***To investigate the role of information systems that are used to support tactical management tasks.***

Information systems such as Management Information Systems (MIS) and Decision Support Systems (DSS), are used to provide tactical management information. Therefore, the aim of this objective is to investigate various approaches, including information systems architectures and technologies, that are used to deliver tactical management information. Furthermore, this objective will also provide an investigation into the role that information systems provide, in the production and dissemination of information when supporting tactical management information needs.

OBJECTIVE 4 - *To investigate modelling approaches for supporting tactical management decision making.*

Various conventional data modelling approaches are available for modelling the dynamic and static properties of data. Such approaches are used to ensure that the architecture of an information system closely reflects the real world domain with which it represents. The principle aim of this research objective, is to provide a review of the various static and dynamic data modelling methods that are used to accurately model the information system, in accordance with the requirements of tactical management.

OBJECTIVE 5 - *To develop a modelling framework for supporting tactical management decision making.*

The aim of this research objective will be to develop a modelling framework that encompasses the necessary functionality for supporting tactical management processes. This objective will be achieved by drawing upon the research presented throughout both sections A and B of this study. Specifically, the modelling framework that will be suggested will incorporate a business object model that makes use of co-ordination theory constructs.

OBJECTIVE 6 - *To develop a software tool for demonstrating the modelling framework suggested in objective 5.*

In order to implement the proposed modelling framework suggested in objective five above, this research objective, if possible, will develop a prototype application that will support an implementation of the proposed modelling framework.

OBJECTIVE 7 - *To test the modelling framework suggested in objective 5, in order to derive results as to the efficacy of the framework for supporting tactical management decision making.*

In order to validate the theory development of this study, this research objective will outline an environment that is suitable for testing the proposed modelling framework.

1.6. RESEARCH METHODOLOGY

Various approaches to conducting a period of research fall into two main categories. Namely, quantitative and qualitative. Quantitative research provides a method for arriving at predictions through statistical procedures or other means of quantification. Qualitative methods on the other hand details a kind of research that produces findings that are not derived from statistical or other quantification procedures. In essence, qualitative research results in a different type of knowledge than quantitative inquiry. From the perspective of this study, it can be argued that because that will be a lack of numerical quantification in this study when deriving results from the research presented, the research method adopted falls into the qualitative category. This research is therefore not concerned with establishing a right or wrong or even a true or false answer. In contrast, its main thrust is to establish whether or not the approach set out in objective 1.5 does indeed support the tactical management decision making process. The approach adopted by this study therefore, outlines a more hermeneutic approach to research, in so much as it is interpretivist (Galliers 1985, 1992; Galliers & Land 1987) in nature.

The objectives outlined above detail the development of a modelling approach that can be applied to the supporting of tactical management decision making. Moreover, they also outline the evaluation of this in a real world environment. From this point of view the research method used in this thesis from diagnosing a problem to taking action and then evaluating it, is similar to that described by Susman (1983). Displayed in figure 1.3 below, it shows a five stage approach to starting and completing a period of research.

Figure 1.3. Five stage research approach, (Susman, 1983)

These are; Diagnosing, Action Planning, Action Taking, Evaluating and Specifying Learning. In relation to this thesis, the areas that have been established will correspond to; problem recognition, research identification, software/ideas planning, software construction/implementation, testing and evaluation and specification of the overall contribution of the study. This method is consistent and acceptable with the cyclical approach expounded by Checkland (1991).

Although the diagram above states a particular research approach, figure 1.4 aims to delineate the nature of a research study. The areas within the diagram are references which are intended to draw comparisons with empirical reality. Over time a possible prediction, based upon the relevant modes of enquiry will prevail. The three research praxes: intervention, reduction and interpretation are all co-present, regardless of the research practice adopted. Their prominence is prevalent in respect of some elements being stronger than others. Therefore, it is possible for a research study to have a mix of the suggested elements. Although the approach suggested by Baskerville and Wood-Harper (1996) would not in itself provide a systematic approach to conducting a period of research, the model, provides guiding elements that can be used to shape various ideas produced from the research. For example, the research approach carried out in this study, will be based around reviewing the research literature, and then subsequently forming new ideas that are finally refined to a point whereby they can be tested. It is recognised however, that throughout the ideas forming stage, a certain amount of intervention into the research approach will be required in the way of revising and changing ideas, as learning in relation to interpreting the problem domain will lead to a greater understanding of the activities carried out by tactical level management. From this perspective, this research study is not primarily concerned with prediction through the reduction of facts, but more through a cycle of understanding and change. The figure below points to where this research study should belong in the 'Research Elements' model suggested by Baskerville and Wood-Harper (1996) in the opinion of the author.

Figure 1.4. Research Elements, (Baskerville and Wood-Harper, 1996)

1.6.1. Evaluating the Thesis

When testing an information system it is not always necessary to determine whether something is true or false (Walsham, 1993). Indeed, the overarching aim of this research is to suggest a more appropriate method for producing tactical management information. Therefore, when devising a suitable testing strategy, it becomes clear that there is a distinct need to test the research, in order to provide evidence as to its validity. In this case, it is essential to provide evidence regarding:

- The feasibility of co-ordinating tactical level management processes using the modelling approach suggested in section 1.5, objective five.
- The usefulness of such a representation in facilitating both the creation and modification of the modelling framework, as well as providing an insight into alternative approaches for producing tactical level information.

Specifically, the feasibility and usefulness of this thesis hinges around the formulation of a testing plan that will in essence, point to the usefulness of the modelling approach suggested in section 1.5 objective five. It can be ascertained that this will measured in the following ways:

- The creation and implementation of a testing plan that will act as a benchmark that will point to the appropriateness of the modelling approach for representing tactical level processes.
- An evaluation of such a testing plan, that will outline if the modelling approach adopted in section 1.5 has proven useful for supporting the tactical management decision making process.
- An outline of future directions that this research could be extended into, based upon the testing plan outlined as part of objective seven will, be presented in the final chapter.

1.7. STRUCTURE OF THE THESIS

In order to fully address the research objectives listed in section 1.5 above, the structure of this thesis will be divided into three sections. Contained in these sections will be a series of chapters. This is shown in figure 1.5.

Section A, will be comprised of three chapters. These chapters detail the role of the tactical level manager, the environment with which management at this level operates, information systems used for making tactical level decisions and the information needs that tactical level management have. In short, section A is designed to introduce the problem area.

Section B, also comprises three chapters. This section will provide a survey of various approaches to static and dynamic data modelling, along with an introduction to the co-ordination theory approach and its applicability and usefulness for analysing organisational processes.

Section C, will detail the role of the modelling framework suggested in section 1.5, objective five, as an approach for supporting tactical level decision making. Furthermore, it will detail the functionality of the modelling tool suggested in section 1.5, objective six, whilst presenting evidence of three verification tests that will be used to assess the design constructs of the proposed modelling approach, for supporting tactical management decision making. This section will conclude by readdressing the CBOM requirements outlined in chapter ten, and assess whether or not they have been met.

Finally, chapter eleven will provide an assessment of the thesis delivered, including relevant contributions to knowledge.

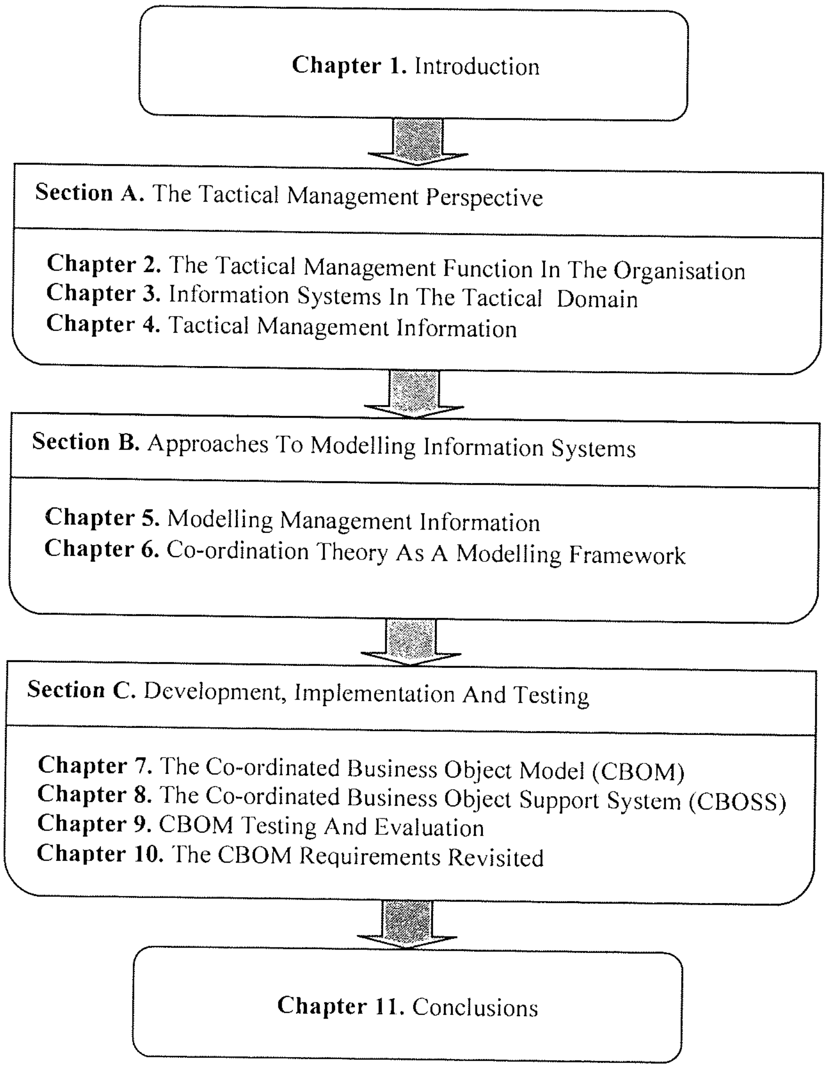

Figure 1.5 Structure of the thesis

SECTION A

The Tactical Management Perspective

A.1. INTRODUCTION

Section A introduces the first part of this thesis. In short, this section is concerned with introducing the role of tactical management. In doing so, it will delineate the wider perspectives of the tactical management function, detailing both its operational capacity and required information needs.

A.2. OVERVIEW OF SECTION A

This section consists of three chapters. They are summarised as:

- Chapter 2 - The Tactical Management Function In The Organisation
- Chapter 3 - Information Systems In The Tactical Domain
- Chapter 4 - Tactical Management Information

The contents of these chapters are outlined below.

A.2.1. Chapter 2 - The Tactical Management Function In The Organisation

Chapter two is the first chapter in section A. The primary aim of this chapter is to introduce the role of tactical management. The landscape of the chapter consists of two areas. Firstly, it considers the organisational context within which the tactical level manager operates. Secondly, by considering the stratification of the management hierarchy, it highlights the relationship between the tactical, strategic and operational levels.

A.2.2. Chapter 3 - Information Systems In The Tactical Domain

Where the aim of chapter two is to introduce the role of the tactical manager and the environment in which he/she resides. The aim of chapter three is to introduce the role of information systems, and their usefulness for supplying the tactical level manager with information that is appropriate to his/her information needs.

A.2.3. Chapter 4 - Tactical Management Information

The final chapter in this section will focus upon what constitutes tactical management information. Specifically, it will consider two areas:

The Scope Of Tactical Management Information. This will detail the scope and characteristics of tactical level information, whilst generally providing a review of the type of information that is required by management at this level.

Various Approaches and Suggested Architectures. This will consider various approaches and technologies used for delivering tactical level information.

CHAPTER 2

The Tactical Management Function In The Organisation

2.1. INTRODUCTION

Tactical management in the view of Anthony (1965), resides between the strategic and operational levels within an organisation. Other authors such as Gorry & Scott-Morton (1989), suggest that tactical management resides between the extremes of operational control and strategic planning. From this point of view, strategic level management make decisions, tactical level management transmit and co-ordinate, whilst operational level managers implement (Dichter, 1991). The organisational type in which tactical level managers operate however, can belong to various structures. Distinctions between these structures, generally rely to some extent, in the degree of hierarchy an organisation reflects. Some organisations are hierarchical, whereby there is a distinguished chain of command. Alternatively, some prefer what has been termed a 'flatter' or more 'horizontal' approach, which is team based in its style. However, regardless of the organisation's structure, operational, tactical and strategic level decisions still have to be made. Therefore, this study does not argue in favour of a hierarchical or flatter structure, it merely recognises the existence of the tactical level function within the organisation, regardless of its degree of hierarchy.

Therefore, the aim of this chapter is to introduce both the role of the tactical level manager and to describe the environment in which management at this level resides. The distinguishing characteristics of this environment will focus upon the traditional vertical structure, and the more modern horizontal structure.

2.2. OVERVIEW OF THE VERTICAL STRUCTURE

Many authors over the past forty years have posited views in regards to the existence of various organisational structures (Burns and Stalker, 1961; Drucker, 1988; 1991; Handy, 1993; 1995; Mintzberg, 1979; Mintzberg & Quinn, 1988; Porter, 1985). Furthermore, as the information age becomes more embedded into the organisations business practices, so new systems of organisational structure, such as virtual organisations or matrix structures, are proposed. However, the traditional hierarchical structure of the organisation is still prevalent within many companies. Described by Anthony (1965), figure 2.1. below shows the organisation to be divided into vertical domains. These domains are very often recognised as departments or divisions in many organisations. Opponents to the hierarchical structure however, note that the stifling of creativity and initiative, as being inherent in the structure (Kettley, 1995). On the other hand, it has been noted that the hierarchical structure can release energy and creativity, rationalise productivity and improve morale (Jaques, 1989).

Figure 2.1. The organisational hierarchy, adapted from Anthony (1965)

Max Weber (1922), in his renowned text 'Wirtschaft und Gesellschaft' (Economics and Society), suggested a definition of organisational hierarchy, which describes levels of graded authority along with a super and subordinate relationship between vertical levels. From this perspective, the definition below draws parallels with Anthony's model presented in figure 2.1.

> "The principles of office hierarchy and of levels of graded authority mean a firmly ordered system of super and subordination in which there is a supervision of the lower offices by the higher ones... The principle of hierarchical office authority is found in all bureaucratic structures: in state and ecclesiastical structures as well as in large party organisations and private enterprises". Weber (1922).

Both Weber and Anthony suggest a model of the organisation that demonstrates operational and tactical levels as being subordinate to the strategic level. In addition, tasks that are performed by management at various levels in the organisation, will undoubtedly differ in levels of complexity. For example, a typical operational level task of signing an invoice for a particular client, will not have the same permutations and requirements in complexity of thought as a strategic level task, such as deciding upon the diversification strategies for new markets. In essence, hierarchy provides a command structure for all staff. In this case, employees know who they report to and who takes responsibility for delivering the organisation's objectives.

This section will consider two further forms of hierarchy, notably, functional hierarchies, where the organisation is centred around the achievement of activities and tasks, and hierarchies which are based around authority and responsibility.

2.2.1. Functional Hierarchies

The functional hierarchy is also referred to as the 'unitary' form or 'U-form' (Williamson. 1975). Within this type of hierarchy, activities that are carried out by tactical level managers cross departmental and organisational boundaries. Therefore in the case of a virtual organisation, the functional hierarchy would be built upon a virtual collaboration concerning two or more organisations. Viewing the organisation from an activity or functional perspective, entails a hierarchy that is driven by the need to fulfil its functional objectives. This particular structure is very similar to the matrix organisation. It is task driven in contrast to authority driven, and therefore it comes about as a by product of the need to achieve task completion.

2.2.2. Authority and Responsibility Hierarchies

Authority is the power assigned to certain individuals. It also reflects the employees' position within the organisation. It affects areas such as controlling, co-ordinating and the implementation of functions. The concept of authority, determines a single line of command. Authority however, may come from various quarters. Peabody (1957) (taken from Alshawi (1991)), noted four different areas where this might be so.

- **Authority of Legitimacy.** This details the authority often found between the employer and the employee.
- **Authority of Position.** This type of authority is attached to the positon and not the person who holds it.
- **Authority of Competance.** This concept is largely exhibited by technical knowledge and experience.
- **Authority of Person.** This relates to some authority belonging to the person, usually by some personal attributes.

The approach listed by Peabody is broad in context, and covers many of the areas where sources of authority may occur. It does not however, mention factors such as authority derived from competition or influence, or through perhaps more deviant measures such as blackmail.

2.2.3. Relationships Between Vertical Levels

Tactical level managers belong to recognised functional areas such as departments or divisions. The model proposed by Anthony (1965), recognises a structure that has three main layers. However, a more realistic stratification would recognise a structure that had multiples of layers (Jaques, 1989). Furthermore, the same author notes that task complexity becomes more prominent the higher up the hierarchy a manager resides. As Jaques (1989) suggests;

> "The complexity of the problems encountered in a particular task, project, or strategy is a function of the variables involved – their number, their clarity or ambiguity, the rate at which they change, and, overall, the extent to which they are distinct or tangled. Obviously, as you move higher in a managerial hierarchy, the most difficult problems you have to contend with become increasingly complex". (Jaques, 1989).

Taking the above into account, Jaques also argues that weaknesses may occur if there are too many levels in the vertical structure. For example, exacerbating communication and creating ambiguity in the command hierarchy, as well as diluting the structure to such an extent that the superior-subordinate relationship becomes blurred.

2.3. OVERVIEW OF THE FLATTER STRUCTURE

In the past, many organisations have undergone a period of de-layering, in order to attain a more efficient organisational structure. The flatter structure (Kettley, 1995; Coulson-Thomas, & Coe, 1991), functions by means of primus groups or matrix overlap groups. In short, the hierarchical structure is broken down to facilitate semiautonomous working teams, that make decisions as a group. Reductions in running costs and changes to the management structure, in order to align and improve the information supply between tactical level managers and line workers, are amongst the primary reasons for undergoing the move to a flatter structure (Kettley, 1995). However, despite various claims, the popularity and use of such a structure has gone largely unchallenged in the literature. The delayering process has been used in recent times by various organisations to remove unnecessary vertical levels in the organisation. Indeed, it has been observed that organisations who do not alter their structure when conditions dictate, are prone to a 'Structural Darwinism' (Zeffane, 1992). The flatter management structure however does not require a large or complex chain of managers, and the supposed unnecessary layers of the hierarchical system are therefore stripped away, theoretically allowing information to pass more freely from manager to worker.

The flat structure however, adopts a democratic method of making decisions, generally based upon employee specialisation. However, from a logistical standpoint, this determines a degree of hierarchy. Indeed, Jacques (1989) argues that where there is a leader, there is a hierarchy. The hierarchical system places decision making authority in the hands of a chosen few. However, detailing the group as being accountable, assumes that the power does not rest in the hands of a manager. This view was highlighted by Jaques (1989);

> "None of the group-oriented panaceas face this issue of accountability. All the theorists refer to group decisions, and group consensus, none of them to group accountability. Indeed, they avoid the issue of accountability all together, for to hold a group accountable, the employment contract would have to be with the group, not with the individuals, and companies simply do not employ groups as such. ....specific tasks within that given work are assigned to you by a person called your manager (or boss or supervisor), who ought to be held accountable for the work you do".

Jaques (1989), points out that group accountability cannot be enforced as groups cannot be made accountable. He strengthens this argument by observing that employees are individually accountable, owing to the nature of the employment system that hired them.

In contrast, the flatter structure theoretically enhances the information flow as all employees are party to the same information. However, both the flatter and hierarchical approaches, still determine that tactical type decisions still have to be administered, regardless as to how horizontally or vertically aligned the management structure. Therefore, any information system that models tactical level approaches would still be user (tactical management) driven, and would still need to embody the tactical type functionality needed to produce tactical type information.

2.3.1. Relationships Between Horizontal Groups

Whereas the hierarchical structure represents a superior-subordinate relationship, Dawson (1996) proposes two different types of relationship within the flatter structure. These are;

- **Functional Structure**, this is where horizontal divisions represent different functions and are driven by these functions.
- **Functional Product Groups**, whereby the horizontal boundaries are between product groups.

Where functional and product groups occur, it is feasible that within virtual collaborations and departmental divisions, there might be a combination of different groups working towards a common organisational goal. Tactical level decisions however, are also administered in organisational structures that arguably present characteristics of both the vertical and horizontal structure. The next section will examine these in further detail.

2.4. ENHANCEMENTS TO HIERARCHICAL AND FLATTER ORGANISATIONAL STRUCTURES

A will to implement new working methods, as well as a drive to remove the levels of hierarchy within the organisation, have resulted in enhancements to hierarchical and flatter structures. This section will consider two organisational forms, that maintain an individual identity from the traditional vertical and flatter structures.

2.4.1. Matrix Structures

The matrix structure (Kingdon, 1973), in contrast to the aforementioned vertical (hierarchical) and horizontal (flatter) structure, is neither a product or a functional organisation. In short, a product organisation overlaps a functional organisation. Implicit in the various functional areas stated in figure 2.2 below, is the capacity for each unit tactical manager (engine, bodywork and testing managers), to have product hierarchies below them. Moreover, each specialist is assigned to one product type. The functionality of the matrix structure distinguishes between employee specialists working in a defined product group, which would incorporate a designated project manager. This style of organisation means that the specialists answer to two managers. The emergence of the matrix structure usually comes to fruition, in response to the need for organisations to ensure the quality and extent of lateral relationships of employees working in turbulent market conditions.

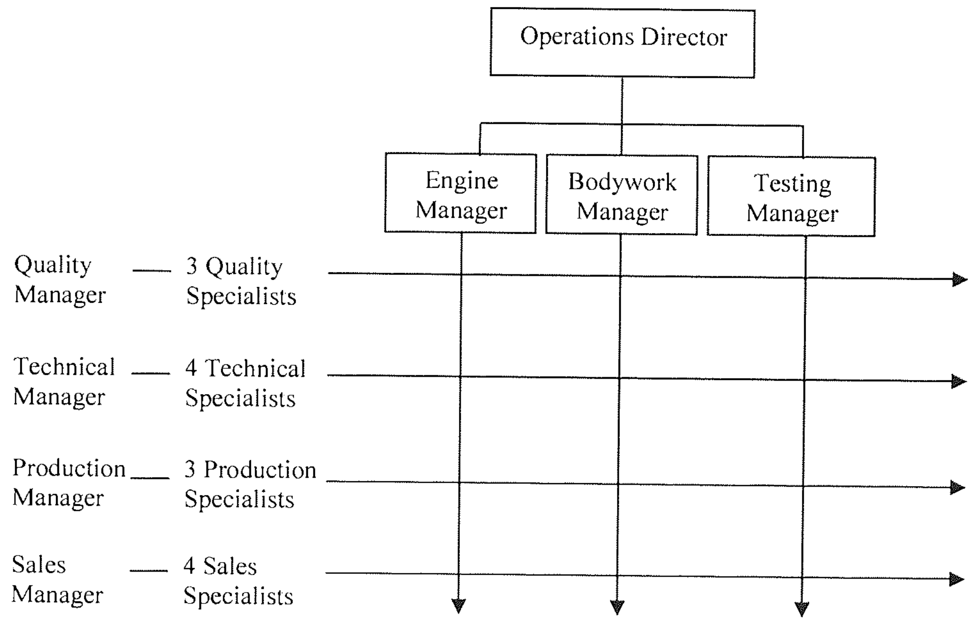

Figure 2.2. The matrix organisational structure

The matrix structure is highly bureaucratised and formalised. In practice, the flow of information throughout this structure should flow effortlessly. However, this is not always the case (Dawson. 1996). Many organisations have adopted the matrix structure, most notably NASA. Some organisations however have changed the matrix structure to fit their specific organisations needs. Taking an example of the Shell petroleum organisation, they have installed what as been termed a "multi-layer matrix structure that combined functional expertise, product-related knowledge, site-specific know-how and experience" (van den Bogaard and Speklé, 2003).

2.4.2. Project Teams

Just as the matrix structure presents a rigid organisation, the project team details a more flexible and adaptive structure (Bryan, 2000). The project based organisation is shown in figure 2.3 below. The figure describes four temporary project groups (Group 1, Group 2, Group 3 and Group 4), that are made up from workers out of four different departments (Sales, Technical, Engineering and Management).

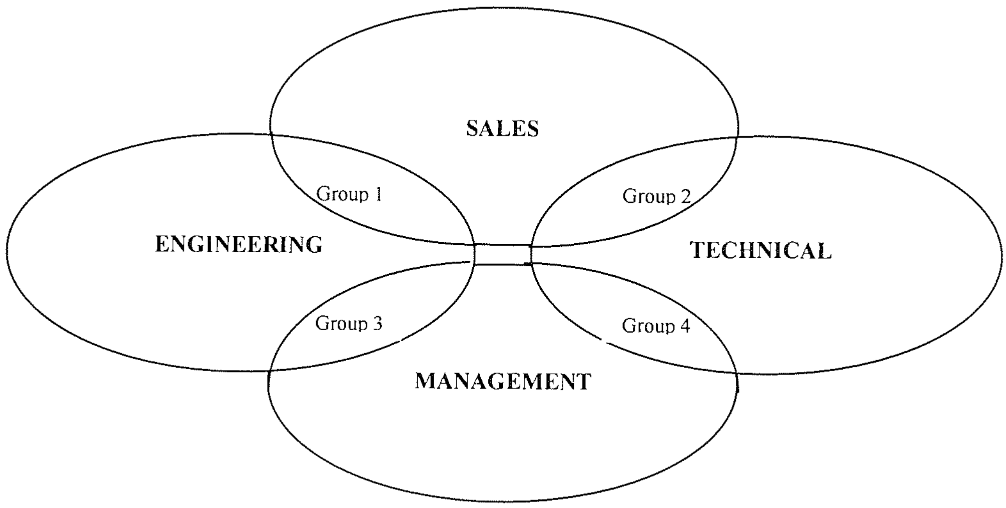

Figure 2.3. The project group

High in functional specialisation with a project focus, these teams are typically small and are formed on a temporary basis, when projects arise that require a certain mix of skills and specialisation (Malone and Laubacher, 2003). From this perspective, decision making is decentralised and devolved by team managers, who in the context of this study, represent the tactical level. From a critical perspective, the project team approach to management draws similarities with the management structure found in virtual collaborations. Such ventures usually detail organisations forming temporary team based partnerships, that harness the skills of the associated organisations, in an attempt to raise revenue by trading over the internet. Management decisions at all levels are also found in such ventures (Brockie & Golder, 2001).

2.4.3. Virtual Enterprise Teams

With the advent of the internet, organisations are no longer restricted by physical boundaries when searching for opportunities to raise revenue. However, as virtual organisations embark upon virtual collaborations, the resulting effect usually realises a virtual enterprise that has its own structure with strategic, tactical and operational levels. The Virtual Organisation (VO) exists in cyberspace with its members geographically dispersed. Organisations that participate and trade as VO's, have greater leverage for flexibility than in a corresponding static organisation. The attributes that underpin the functionality of the VO, rely primarily upon intangible elements such as trust and relationships. Furthermore, as the VO has developed, many recognisable forms have also become evident. Burn, Marshall and Wild (1999), define six different forms of (VO) known as 'Models of Virtuality'. These depict the diversity of forms that a business operating in the VO may undergo. Namely;

Virtual Faces - The cyberspace element of a non-virtual organisation.

Co-Alliance Models - The collection of partnerships that combines to serve the virtual organisation.

Star-Alliance-Models - A co-ordinated network of interconnected members reflecting a core surrounded by satellite organisations.

Value-Alliance-Models - Based on the value or supply chain model, the contractor co-ordinates various participants on a project basis.

Market Alliance Models - Market-Alliance models exist primarily in cyberspace and depend on trading from this domain of the business. Amazon.com provides a good example of this.

Virtual Brokers - A virtual broker is a designer of a dynamic network. Brokers provide additional strategic opportunities either as third party, value added suppliers, and information brokers, providing a virtual structure around specific business information services.

As part of their analysis, Burn, Marshal & Wild (1999), also point out that the culture of a virtual organisation is dependent upon the structural alliances, strategic positioning, knowledge management and 'Information Communication Technology' (ICT) surrounding that organisation.

2.5. THE MANAGEMENT LEVELS

So far this chapter has highlighted various organisational forms in which the tactical level manager operates. The remainder of this chapter will consider in further detail, the management levels that form the organisational hierarchy. Using the model suggested by Anthony (1965), the remainder of this section will consider the three layers, *strategic, tactical* and *operational* in more detail.

2.5.1. Stratification Of The Organisation

The stratification of the organisation generally centres around a particular hierarchical structure where managerial staff hold a position. These positions can generally be equated to *strategic*, *tactical* and *operational.* Many organisations however, may have a vertical structure where there are many levels between operational control and the strategic level. Others may be more horizontal, and hence the levels of hierarchy between the operational and strategic layers are much fewer. The higher up the management hierarchy a manager resides however, the more responsibility he/she will have. As Brown (1971) observes.

> "In an employment hierarchy, each ascending stratum includes in its degree of accountability for work, the work of the subordinate stratum below it, until finally the chief executive is accountable for the entire work of the hierarchy. This leads to the notion of considering an executive hierarchy as a series of ascending orders abstraction concerned with work." (Brown, 1971)

Brown's analysis is consistent with the traditional view of the hierarchical organisation. Indeed, increased responsibility will become more prevalent, owing to the span of control widening. Furthermore, as responsibility increases, so an increase in accountability will also be realised.

Central to the management process is planning and supervision. Anthony (1965), as discussed previously, suggested that the organisation is made up of three distinctive levels, which all represent different functional objectives that require different information needs. Figure 2.1 above provides a representation of where the tactical level stands in relation to the strategic and operational levels. All levels of the structure are connected, in so much as the strategic level passes objectives to the tactical level, which in turn deploys these objectives throughout the operational level. Subsequently, information as to the effectiveness of these objectives is then passed back to the tactical and strategic levels (Brockie & Golder, 2000).

As a consequence of the different demands placed upon the various levels within the management hierarchy, the information need throughout the structure will undoubtedly change. For example, strategic level information will typically be more abstract, whilst the information need further down the structure will become more detailed and specific. In relation to managerial levels of activity, it is not so obvious what the managers level is in accordance to the activity(s) performed. The reasoning for this can be found in the difficulty in typifying an activity (Alshawi, 1991). Therefore, it is easier to determine the level of management a manager belongs to in accordance to status, authority or position. Such an assertion is also supported by Wu, Chen and Lin (2003) who noted that from an IS perspective management activities vary between different levels of management. The following sections below, consider this by outlining the strategic, operational and tactical management levels in more detail.

2.5.2. Strategic Level

The strategic level is primarily concerned with the direction of the organisation (Morden, 1993). It is at this level that the corporate mission of the organisation is set. Here the organisation's place in its environment, strategy for the coming year, diversification and core business assessment, as well as the formulation of overall enterprise direction, are all laid out at this level. Internal structuring, as well as the location of operations and product markets, are also considered at this level. Consequently, as a result of the corporate mission statement, organisational objectives are formulated.

The strategic level also lays down the purpose and ethos of the organisation. Purpose was defined by Argenti (1974, p.39) as;

> 'the reason why the organisation was formed or why it now exists. All organisations are originally formed to provide a specific benefit for specific groups of beneficiaries'

The purpose is the distillation of both mission and objectives. The same author also proposes an explanation of ethos as;

> 'How an organisation behaves towards its employees and all other people or groups... with whom it interacts. These include... the state, the local community, employees, suppliers and so on... The way in which it has decided to behave constrains and modifies the means it uses to achieve its purpose'. Argenti (1974 p.39).

Such definitions prove useful when establishing the strategic managers role in the organisation. From this perspective, the strategic level is concerned with the bigger picture. Decisions at this level are of an abstract nature. Similarly, the same ramifications also stand in relation to the information perspective regarding strategic level management. Here, aggregated data produced by an information system such as a 'Knowledge Based Systems' (KBS) or more notably an Executive Information Systems (EIS), typically provide strategic management with strategic level information.

2.5.3. Operational Level

The operational level is responsible for the everyday running of the organisation. Examples of operational level objectives include data processing and customer interfacing. In essence, it is at this level that strategic level objectives are implemented into workable and realistic objectives. For instance, a strategic level objective may be to 'increase production', at the operational level this may be deployed as 'provide more overtime' or 'take on more staff'. The operational level therefore, follows a devised programme of operations that allow for the accomplishment of organisational objectives.

The operational level is primarily concerned with the detailed quantification, implementation, and planning at the functional level of the organisation. Various information systems used by operational management comprise, 'Office Automation Systems' (OAS) or 'Transaction Processing Systems' (TPS). Information derived from these systems, is used to return information to the tactical level. Subsequently, information can then be passed back to the strategic level regarding the effectiveness of strategic level objectives.

2.5.4. Tactical Level

Tactical management resides between the extremes of operational control and strategic planning within the organisation (Gorry & Scott-Morton, 1989; Dopson & Stewart, 1993; Hattrup, 1993). The tactical level is responsible for implementing strategic level objectives. Moreover, management tasks at this level also focus upon reporting the effectiveness of existing strategic level objectives back to strategic level management, as well as acquiring new approaches for conducting organisational processes, based upon previous and past performance. In short, the primary function of a tactical manager is to guide, standardise and control the various operations that are considered to be crucial for the everyday execution of the organisation's tasks (Wheatley, 1992). There are many definitions as to the exact meaning of the term 'tactical management'. Harris (2000), outlines that tactical management is, "the link between executive strategy and the employees who will carry out goals." Hicks (1995) in contrast, proposes that tactical management comprises; "The process of making decisions at the middle or co-ordinating level of the organisation. The decisions are made primarily to reach the present goals of the organisation. A common decision on this level involves resource allocation for the present needs of the organisation". This definition to a degree assumes that the 'middle or co-ordinating level' points to the tactical level within the organisation. Secondly, it indicates that 'present goals' points towards current goals of the organisation, that have been defined by strategic level management.

It is the primary function of the tactical level manager to extrapolate and exploit resources at the organisation's operational level. This view is consistent with Anthony (1965); "The process by which managers assure that resources are obtained and used effectively and efficiently in the accomplishment of the organisation's objectives". If it is assumed that the organisation's objectives are set by strategic management, then this definition clearly relies upon the tactical level manager holding a high degree of autonomy with regards to planning, organising, co-ordinating, deciding and controlling organisational resources. This point is recognised by Reynolds (1995), "Tactical decisions involve allocation and control of the firms' resources to meet the objectives that support the strategic goals of the business".

As strategic decision making is crucial to the overall welfare of the organisation, the tactical component must include an evaluation of the detailed alternative courses of action, regarding the organisation's strategic context. As well as providing objectives, criteria and standards for performance evaluation, so that effective monitoring and review can be administered (Morden, 1993). The tactical component also attempts to achieve co-operation between potentially conflicting functions within the organisation (Shapiro, 1977), whilst supporting the long range planning activities of the organisation. These may include market position, expected employment and productivity levels.

2.6. CONCLUSIONS

The aim of this chapter has been to introduce both the role of the tactical level manager, and to describe the environment in which management at this level resides. The distinguishing characteristics of this environment have focused upon the traditional hierarchical structure, whilst taking into account more modern structures, such as horizontal systems and autonomous work units. The purpose of examining organisations with varying degrees of hierarchy, was to arrive at a point upon which the identification of hierarchical stratification can be ascertained.

The tactical management role however, as mentioned in chapter one, is traditionally supported by a computer based information system. The next chapter will look at the use of such systems in more detail, and will specifically highlight the use of information systems in relation to the tactical domain.

CHAPTER 3

Information Systems In The Tactical Domain

3.1. INTRODUCTION

The previous chapter introduced the role of the tactical level manager. Specifically, it outlined the existence of the tactical management function within various organisational forms. Regardless of the organisation's structure however, most managers nowadays use computer based information systems to support not only the operational aspects of the organisations everyday tasks, but also the management decision making process. Indeed, as Hicks (1995) observes "Information systems have become crucial to the functioning of modern organisations and businesses. Firms are using information systems technology to gain competitive advantages over their rivals".

This chapter will further broaden the characteristics of the tactical management function, by examining in more detail, the role information systems play in the production and dissemination of information for supporting the tactical management information needs.

3.2. WHAT IS AN INFORMATION SYSTEM?

Research carried out by Stafford Beer (1979) in the field of cybernetics, has presented a usefull way of looking at the role of information systems in organisations. However, describing an information systems in this day and age, would undoubtedly recognise a computer based system that can process and co-ordinate information. Furthermore, it would support decision making at all levels throughout the organisation, and in the context of this study, provide information for the tactical level manager that is relevant for both the implementation of strategic objectives and the deployment of operational level objectives. Figure 3.1. below, adapted from Laudon & Laudon (1998), describes the relative functions of an information system. It portrays the information system as processing information relative to the organisation and its environment. The diagram underlines three basic activities that are executed by the information system. These are *Input*, *Processing* and *Output*. Here, information that is required for the organisation, is inputted into the system, processed, and then outputted for decision making purposes. Feedback is then given, in order to provide possible refinement and evaluation of the returned information.

Figure 3.1. Functions of an information system, (Laudon & Laudon, 1996)

The diagram above, in displaying the characteristic functionality of an information system, does not provide any indication as to the paradigm that the system should represent. Information in this context flows in a cyclical manner.

Finally, arguments that support the failure of information systems to realise their potential have focused upon deficiencies in their design. Ackoff (1967) in his paper 'Management Misinformation Systems', outlined five common and erroneous assumptions that compromised the success of information systems. These are;

- The critical deficiency under which most managers operate is the lack of relevant information.
- The manager needs the information he wants.
- If a manager has the information he needs his decision making will improve.

- Better communication between managers improves organisational performance.
- A manager does not have to understand how his information system works, only how to use it.

An information system must be supported by a modelling architecture that meets not only the problem domain, but also specifically targets the management information need. Although Ackoff made this point over thirty years ago, his critique still holds valid to this day.

3.3. FROM DATA TO INFORMATION

The notion of a data rich but information poor organisation, has received much attention in the literature (Golder, 1997). Designing the information system to more closely match the real world paradigm, has provided in part, a solution to such a problem. The modelling constructs of many information systems however, generally tend to be focused around the transformation of low level data into meaningful information. In 1954, Norbert Weiner described information as being; "a name for the content of what is exchanged with the outer world as we adjust to it, and make our adjustment felt upon it". From this perspective, information exchange is evolutionary. However, there is an exchange point, generally at the information systems design level, whereby base level data that is held in a database for example, becomes information that the tactical level manager can use to make decisions.

Data in its purest form, comprises raw facts that are derived from calculations such as counting, measuring, weighing or by collecting factual data from reading and observation. Information on the other hand, is derived from data, and is vital for generating change and providing the tactical level manager with the correct understanding of their particular business issues (West & Courtney, 1993). In essence, information is the building block of all knowledge, and the main attribute that underpins the concept of an effective information system. Used correctly, it will result in effective decision making and control (Ardell, 1993). Moreover, information in its simplest form, ties together all the components of the organisation. For example, personnel, machines, money and materials. It also defines structure and organisation, unlike data which has no structure and is uninformed (Schoderbek et al, 1990). The quality and accuracy of data however, ultimately effects the quality of information. For instance, data entered by users may be incorrect, or it might have aged or have been transcended. The implications for accurate information based upon such data, will undoubtedly be compromised. Consequently, an organisation must use the correct or best data it has at its disposal. Such issues are controlled by the effective managing of data.

3.4. MANAGING DATA

From one perspective, the management of data is governed by statutory law. The Data Protection Act 2002, provides specification for the control and use of data. However, in the context of this study, the concept of managing data will be viewed from an information systems perspective, and analysis of the data protection act will not be examined.

Information has for some time been viewed as an organisational resource (Burk & Horton, 1988; Best, 1996; Braman, 1989). The quality of this information, is to an extent dependent upon the quality of the underlying low-level data. Subsequently, this will ultimately affect the quality of management decisions (Norfolk, 1997). In short, the term 'garbage in, garbage out', holds valid for the information quality delivered by an information system, if the data pool is out of date, or generally inaccurate. Therefore, just like information, data can also be recognised as a resource that requires a certain amount of management. In practice this is generally achieved via the use of a database system.

Understanding the role of data management within the organisational context, is necessary for the efficient execution of the organisations operational capacity (Guynes & Vanceek, 1996). However, managing information is as equally important, as it is used to support the tactical management decision making process.

3.5. MANAGING INFORMATION

Information that is used as a resource, supports decision making. Furthermore, it has also been observed that good decision making throughout the organisation is not only central to the management role, but also directly effects the financial health of the organisation (Wilson, 1993). The dissemination of information throughout the organisational hierarchy however, is necessary if it is to be of any use. Figure 3.2 below, clearly demonstrates the flow of information throughout the organisational layers. Portrayed in the diagram are the three main hierarchical levels, 'Top Management (strategic)', 'Middle Management (tactical)' and 'Operating Management (supervisory)'. Accompanying both sides of this structure are two domains. These comprise 'data requirements', which detail if they are poorly defined or well defined and 'kinds of problems', which are any one of either unstructured and structured.

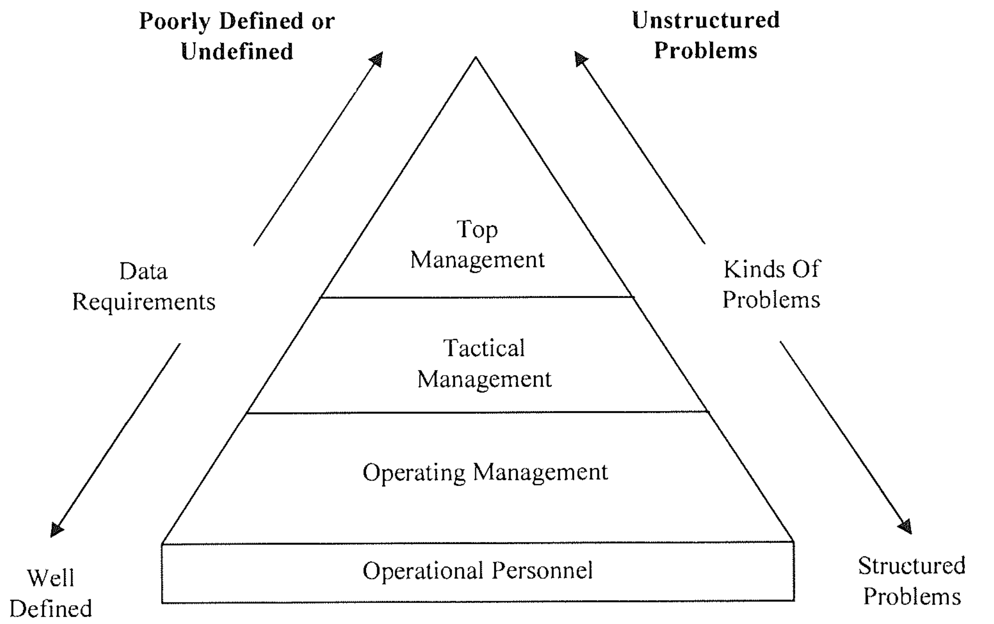

Figure 3.2. Flow of information, adapted from Senn (1990)

The diagram above shows that the further up the management hierarchy, the more unstructured the data requirements and the everyday problems become. For example, a typical strategic management problem may require the formulation of competitive policy, or a decision as to which market diversification strategy to pursue. In contrast, the operational level has defined data and seemingly structured problems. Examples, may include, refunding a customer, or amending ordered details. In the diagram above, Senn (1990) is not only describing the flow of information in relation to the management level, but also the change in complexity that accompanies management decisions at these different levels. Furthermore, it is also noted that as this level of complexity arises, so the structure of the data required to carry out the decision making process, becomes less structured.

3.6. THE VALUE OF INFORMATION

Well managed information will not necessarily provide valuable information. Lucey (1995) provided various characteristics as to what constitutes good information. Notably, *Relevant for its purpose*, *Accurate for its purpose*, *Complete enough for the problem*, *Confidence source*, *Communicated to the right person*, *Communicated in time for purpose*, *Right level of detail*, *Appropriate channel of communication*, *Understandable by the user.* Lucey provides a fairly comprehensive list that underlines the communicative and quality aspects of information. In contrast however, the list does not mention the time aspect of receiving and disseminating information within a particular allocated length of time. Furthermore, it does not point to the cyclical manner in which information travels as a two way communicative process (Glazer, 1993). From this perspective, it would be perceivable to further extend this list by adding *Communicated and available at the right time* and *Action required based on this information.*

Information at the tactical level travels in two directions. For instance, information travels vertically to the strategic and operational levels, as well as horizontally, to other functional areas within the organisation. The time this information takes to travel to its designated areas, results in cost implications. An attempt to assess the value of information from a cost perspective, has been posited by Lucey (1995). This is shown in diagram 3.3. below.

Figure 3.3. The value of information, Lucey (1995)

Here, the Y axis represents the cost, the X axis, the amount / quality of the information. The idea portrayed in the diagram, suggests that an increase in the amount / quality of the information, will result in an increase in the cost of that information. In essence, as the cost increases, so does the value of that item or set of information, owing to the time and effort that was spent on securing it. The diagram also outlines a correlation between 'Amount/Value of Information' and the value derived. Here the author recognises that the greater the volume of information, the lower it's value becomes.

Further work highlighting the value of information has been produced by Glazer (1993). Adapted from his work, figure 3.4. below depicts the value of information within an organisational context. The diagram points to two 'chains'. These are recognised as the, 'downstream chain', which is responsible for the process of exchange between the firm and its customers, and secondly, the 'upstream chain', where the exchange of information is between the firm and its suppliers. The diagram also highlights the exchange of information.

Figure 3.4. The 'Valuation Procedure', (Glazer, 1993)

In this instance, the consumer exchanges information with the supplier when money is exchanged. In relation to this, information is exchanged between the two parties when the exchange process takes place. The model goes on to suggest that the information that is passed between the supplier and the consumer has value. These are simply represented as 'Vs' for the supplier and 'Vc' for the customer. Central to this, is the information that is passed from the firm to the customer and supplier. This is represented as 'Vf'. With these three elements combined, they result in the overall information value which is returned as 'Vi'. Glazer's representation for measuring the value of information is accurate in so far as the only factors that are valid for measurement are an intermediary, a supplier and a recipient. However, the model fails to outline the causes of unknown origins of information, that will ultimately effect the overall information value. These could be undisclosed meetings, or information that has been added to the transaction by anonymous entities, that are both internal and external to the organisation.

Information has different meanings for different organisations, depending mainly on what value they place on that particular piece of information, and its attributes in that instant (Ahituv, 1980). Furthermore, the value of information produced by an information system, for use by tactical level management will be subjective. In contrast, approaches forwarded by the likes of Senn (1990), Lucey (1995) and Glazer (1993), attempt to quantify the value of information in relation to a set of guidelines or a proposed measurement.

In concluding this section, chapter two has outlined that the tactical management role exists regardless of the organisational type, and indeed as Wheatley (1992) outlined, the role of management at this level is to guide, standardise and control the various operations that are considered to be crucial for the everyday execution of the organisation's tasks. Therefore, the meaning of information in the context of tactical level management, will depend in part on not only the tasks they are responsible for, but also the organisational environment in which they operate.

3.7. THE USES OF INFORMATION

Information that is produced by an information system, is generally used by an actor. In the context of this study, this would be a tactical level manager. The possible uses that the actor has for this information were highlighted in a broad based list by Kaye (1995). These are; *Learning and understanding, Teaching, Instructing and training, Discovering and inventing, Problem solving, Decision making and choosing, Informed action and operation, Justifying, Explaining and accounting, Selling and marketing, Image creation, Persuading, Influencing and manipulating, Domination and subordination, Misleading, Deceiving and betraying.* The same author also makes it clear that sufficiency is a key concept when assessing the uses of information. Therefore managers tend to choose the first acceptable result, rather than seek the maximum pay off (Kaye, 1995).

The organisation will also require information regarding its external environment. In this instance, managers tend to use information for such things as planning, strategical analysis, organising, decision making or financial control, in order to realise a competitive advantage over other firms.

Information is an integral part in the formulating of tactical management decisions. Upon receiving information, the manager will proceed to the decision making process in order to derive a use for the current information. Cook & Slack (1984) proposed three types of decision that are supported by information, namely:

Strategic - Operational. Strategic decisions detail the relationships that the organisation has with its environment. Operational decisions are internal matters such as planning and scheduling.

Structured - Unstructured. Structured decisions are clear and well defined. Unstructured decisions are not defined in full and are ambiguous.

Dependent - Independent. Dependent decisions are ones that do not need the support of other decisions. Independent decisions need the support of other decisions.

These decision types are wide-ranging, in so much as their span covers operational to strategic levels within the organisation. However, it is the streamlining of both business objectives and IS objectives that will effect the overall quality of the information system, subsequently effecting the support provided to the above decision types.

3.8. INTEGRATING BUSINESS AND IS OBJECTIVES IN AN INFORMATION SYSTEMS ENVIRONMENT

From a tactical management perspective, information systems form the heart of many complex operational systems (Morden, 1993), owing to the value of information and its importance for the organisation (Ardell, 1993). The integration of information systems plans and business plans within the organisation, in order to realise optimum performance for creating a competitive advantage, is critical (Parsons 1983). In addition, by aligning both the information system and business elements, will undoubtedly allow for the synergistic effects of these two areas to be realised. In short, the alignment of the information system with the organisations business prerogatives, provides efficient gains in the management of information. Integration of these areas are shown in figure 3.5 below.

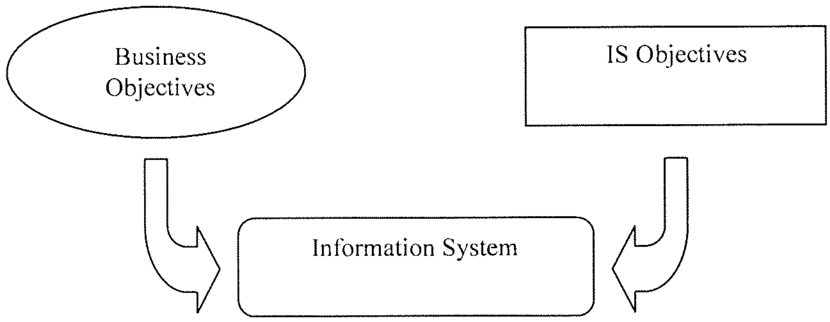

Figure 3.5. Business and IS objectives in an MIS environment

Tactical management tasks typically cross departmental boundaries. Therefore, information and business objectives integration, will require a system that supports this. Enterprise Resource Planning (ERP) systems aim to support a wide variety of organisational activities, via the use of a multi-module approach to integrating the information supply within the organisation. However, these systems in the past have come under criticism, as they have been known to be problematic in relation to their ability to fully integrate the information needs of the organisation (Sumner, 2000).

The span of control that the tactical level manager exercises over various departments within his/her functional domain, may cross both internal, and in the case of virtual enterprises, external organisational boundaries (Brockie & Golder, 2001). Therefore, the use of an information system that adequately reflects the real world domain, is necessary if the correct information is to be made available. Furthermore, apart from assisting the tactical level manager in the decision making process, information systems help management to identify the areas and domains where information is needed. They also allow value to be added to existing information. The relationship between the real world organisation and the information system however is bi-directional (Laudon & Laudon, 1998), and therefore there is a necessity for the information system to be aligned with the organisation. Evidence supporting this was presented by Laudon & Laudon, (1998). "There is a two-way relationship between organisation systems and information systems ...information systems must be aligned with the organisation in order to provide information needed by important groups.". In addition, it is widely recognised in object oriented design and analysis methods, that accurately modelling the different domains of the organisation, will pave the way for more realistic information systems models, that more accurately model the real world domain (Codd, 1970; Chen, 1976; Rumbaugh et al, 1991). However, when integration of both the business model and the information system is achieved, management of such systems will be required. This is considered in the next section.

3.9. MANAGING INFORMATION SYSTEMS

Information systems have many roles when used to manage company activities. For example, these may include, *Accuracy and information currency*, *Reliability and consistency*, *Clarity*, *Situational usefulness*, *Language and international application*, *Timeliness*, *Level of aggregation and summary*, and *Meeting needs for feedback*. (Morden, 1993). Furthermore, it has also been noted that the success or failure of modern organisations, now depends heavily on the adequacy of the information and records used as a basis for executive and managerial decision making (Thomas & Langemo, 1992). However, the decision making capabilities of tactical level managers is directly related to the management of the information system. For instance, administration of an information system's low level data and module updates. In essence, effective management of the organisations information systems, not only leads to supporting accurate decision making, but also maintaining the integrity of the information supply chain.

As mentioned previously, maintaining information systems integrity is crucial if the information system itself is to deliver accurate information. This is certainly true of information systems that are relied upon more than most for supporting the decision making process. Such systems that provide decision support, will be considered below.

3.10. SYSTEMS PROVIDING DECISION SUPPORT

The functionality of a Decision Support System (DSS), extends into the boundaries of both Executive Information Systems (EIS) and Knowledge Based Systems (KBS). Specifically, the DSS provides extended analytical capabilities for management who make critical decisions. In essence, a DSS is based around a model that supports decision support activities. This may be achieved by running simulations or by identifying problem areas based upon current data. As the purpose of a DSS is to complement the manager's decision making process, so there is a need for the DSS to accurately model the functional areas of the organisation with which it will be used as a tool for decision support. DSS, like Management Information Systems (MIS), generally connect to a relational database, and also perform operations based upon available low-level data. In addition, the typical DSS allows the tactical level manager to drill down on current information, therefore allowing the user the opportunity to view organisational level information from different levels of abstraction. Moreover, the graphing and charting of information, also allows the tactical level manager to view information in a suitable form that highlights trends and comparisons.

3.11. THE PROBLEMS WITH INFORMATION SYSTEMS

When designing an information system, it is crucial to accurately design and develop a system that will provide an appropriate solution to, in some circumstances a complex problem. (Maturana, Ferrer and Barañao, 2003). Developing such systems can in some cases be costly (Stamelos, Angelis, Morisio, Sakellaris, Bleris, 2003). In recent times, various concerns about the effectiveness of information systems, have lead to criticisms regarding their lack of responsiveness in relation to user needs, with the added disadvantage of tardiness in respect of the information being supplied (Yeo, 2002). Information systems support however, is usually built around a particular function. For example, production control or marketing analysis. This tends to ensure that the system is confined to a particular isolated domain.

From the tactical management perspective, the integration of information benefits both the interoperability and the reduction of duplicated effort. Furthermore, as the span of control for a tactical level manager may in some circumstances cross functional departments, the subsequent availability and accessibility of information, may theoretically realise an increase in the synergistic potential of these functional areas owing to the availability of more information. In contrast to this, 'information overload' may occur, whereby there is an abundance of information that results in effective decision making being compromised. In addition, it has been observed that the internet has also added to the exacerbation of this problem (Swash, 1998).

3.12. CONCLUSIONS

This chapter has introduced the role of information systems. Specifically, it has highlighted the role of such systems and their uses for supporting the information needs of the tactical manager. Traditionally, information systems have been used to support various elements of the management process such as decision making, co-ordination, control and analysis. Therefore, it is the quality and suitability of the information that the information system supplies, which will have a determining effect upon the quality of management decisions. The next chapter will focus upon this in more detail. Specifically, it will outline the nature of tactical management information, as well as providing an understanding as to the uses and scope of information needed by management at this level.

CHAPTER 4

Tactical Management Information

4.1. INTRODUCTION

Central to the concept of an information system, is the notion of validating and producing accurate information relative to the particular management need. However, in order for this to be realised, the relevant information requirements along with the corresponding characteristics of that information, must be made readily available. In short, identification of the required level of information along with the appropriate technological medium is essential.

Taking this into account, the final chapter in this section will consider two main areas relating to the scope and supply of tactical level information. These are considered below.

The Scope of Tactical Level Information. Chapter three underlined the use of information systems for supporting the tactical level domain. The specific intent of its focus, was to highlight the broad encompassing role of information systems along with various issues relating to the supply and use of data and information. This chapter in contrast, will specifically consider the use and scope of tactical level information. In doing so, it will detail its characteristics, corresponding level of abstraction, as well as its uses and application.

Various Approaches and Suggested Architectures. Current approaches and various technologies that are used when producing, disseminating and delivering information to the tactical management decision making process, will also be considered towards the end of this section.

4.2. DEFINING TACTICAL LEVEL INFORMATION

Tactical level information can be stored in many forms. For example, files, forms, records, discs, maps, photographs, brochures, samples, diagrams, videos, reports, printouts and tables etc. From an organisational context however, managers are both users and producers of information. Specifically, tactical level managers integrate operational level information and organisational strategy (Brockie & Golder, 2000). However, as made clear in chapter two, the tactical management role in relation to formulating business strategy will differ in relation to the organisational structure. From this perspective, the structure will directly affect not only the flow of information, but also the internal and external information that can be leveraged by the tactical level manager.

From a tactical management point of view, making decisions in order to overcome various problems, requires a full understanding of the span of control that the tactical decision maker has. Therefore, process information that is specifically designed for tactical decisions, will in effect, be fairly high-level, if not abstract in nature. From this perspective, typical tactical type information does not cover issues such as; 'The gear system broke down in the press machine'. In essence, the tactical level manager is more interested in, '..and therefore production for the last hour is down by 10% and expected targets for the day will not be met'. Evidently, management levels throughout the strategic hierarchy will require a different interpretation of the same information. This way of looking at functional, as opposed to abstract information, highlights the role of the tactical level manager when delegating instructions to the operational level, as to what course of action to take after these problems have been solved. For example, instructions to the operational level from the tactical level may be to organise overtime in order to compensate for lost production. In short, the concern of the tactical level manager is to resolve this, in order to meet the intended target. The tactical level therefore, requires information that is suitable for supporting the tactical level role in question. Information derived from the everyday operations of the organisation, as well as the strategic objectives that have been set by senior management, will allow for the preparation of tactical level goals and objectives. Figure 4.1 below describes the flow of objectives and information throughout the organisation. It shows a hierarchical system (strategic, tactical, and operational) when planning and formulating objectives at all levels (Ahituv & Neumann, 1990), (Mclean & Soden, 1977), (Wetherbe, 1994).

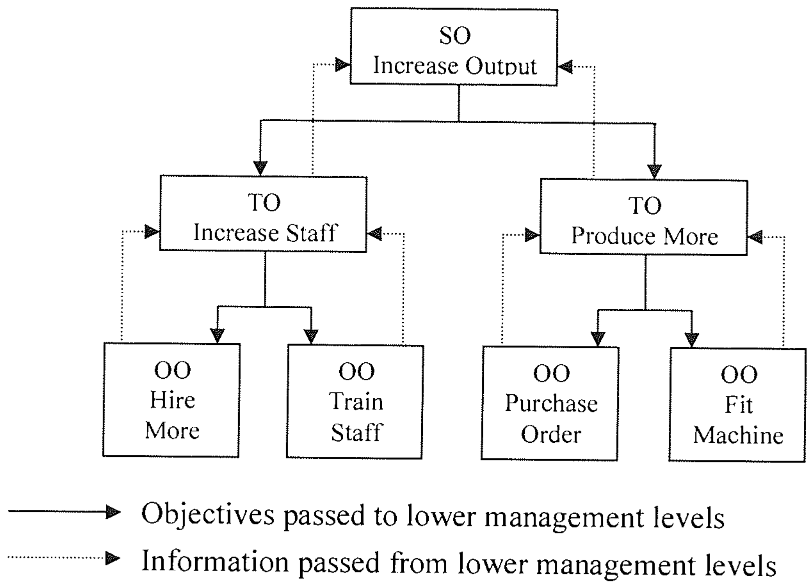

Figure 4.1. Objectives and information flow

The diagram above shows tactical objectives as being part of a hierarchical structure, whereby tactical objectives can be broken down into other 'Tactical Sub-Objectives' (TO). For example, Strategic objectives (SO) are set by senior managers, who then in turn pass these objectives to tactical level managers (TO), who then deploy these objectives as workable implementations throughout the operational level (OO). At all levels of the organisation, information is passed back to the management level directly above (Davis, 1995; Gayialis and Tatsiopoulos, 2004). This in turn allows for any necessary redefining of objectives.

Figure 4.1 suggests that information and objectives exist at varying levels of abstraction. From the perspective of this study, it shows that tactical management formulate tactical level objectives based upon tactical information. Information systems providing decision support, typically supply information at this level. High in analytical power, with an ability to provide the tactical manager with constantly updated information, most information systems that provide decision support, must provide the tactical manager with information that is responsive to changing conditions, as there is a need to supply the tactical level manager with accurate information (Knight & Silk, 1990). In particular, information systems provide the tactical level manager with a means of deriving information in order to support decisions that are semi-structured, rapidly changing, and which are not easy to specify in advance.

As mentioned previously, the tactical level is affected by objectives passed from the strategic level and information passed back from the operational level. As information and objectives change, the role of managers, in particular the tactical level manager, is to; "orient this chaos towards purposeful knowledge creation" (Nonaka, 1991: p.103). In short, strategic level managers articulate metaphors, symbols and concepts about the organisations future. Information at this level is both abstract, and encompasses a wide view of the organisation and it's direction. Tactical level management, whilst implementing strategic level objectives and managing operational level tasks, manage, as noted by Nonaka (1991), 'chaos' amongst their corresponding functional areas. Furthermore, from an information perspective, they also turn tacit knowledge (derived from experience) into explicit knowledge (that which is factual) and use it to develop new working practices, technologies and products. Tactical managers are therefore 'knowable engineers'. They architect and develop information that is generated from other sources.

Understanding the perspective and corresponding context of tactical level information, as mentioned previously, can be dependent upon the organisation within which the tactical level manager operates. Various methods exist that support the supply of information to the tactical level, these are highlighted below.

4.3. DELIVERING INFORMATION

Chapter three underlined the use of management information systems for supporting the tactical management information need. In addition to this, there have been various approaches, that can be considered to posses different architectures for delivering information to the management decision making process. Those areas that are considered relevant, have been investigated throughout the course of this study. A review of these areas will be considered throughout the remainder of this section.

4.3.1. Collaborative Approaches

Section 2.4.3. outlined the role of the virtual organisation and the way it allows information and knowledge to be available to many organisations. By allowing many disparate communities to show knowledge allows for a more robust knowledge sharing strategy, that provides greater multiple channels or knowledge sharing (Pan and Leidner, 2003; Koh and Kim (2003). In addition, As organisations become more integrated, so there are practical software solutions available which aim to support this integration. Many enterprise architectures such as CommerceNet (McConnel, 1997), and San Francisco (Abinavam, 1998), have largely influenced the integration process. Business Process Modelling Tools (BPMTs), such as ARIS, ARIS (2000) or the Workflow Analyzer by Metasoftware (Workflow Analyzer, 2001), which produce business models and simulations, aim to provide better understanding as to how the organisation operates. Other approaches have seen the development of software protocols such as the Light Weight Directory Access protocol (LDAP) (Benett, 2001), which allows individuals to locate a range of organisations and other individuals, business processes and resources such as devices on a network. Furthermore, the Simple Workflow Access protocol (SWAP) (Swenson, 1998), and the Workflow Management Facility (WMF) (Workflow Management facility, 1998), provide access to specific activities located in different enterprises.

The Virtual Enterprise Co-ordinator (VEC) (Ludwig & Wittingham, 1999), provides support for cross-organisational business processes. The Collaboration Management Model (CMM) (Georgakopoulos, et al, 1999), introduces application specific activity states and operations, service activities and service abstractions. By capturing application semantics, process models allow for the integration of heterogeneous services and processes. Although the integration of heterogeneous services is allowed in both the CMM and the VEC, they do not support application-specific states. Furthermore, the ability to provide synchronicity and conversational co-ordination of states is also omitted. In contrast, the CMM adequately supports this. Such collaborative approaches draw parallels with the functionality displayed by Enterprise Resource Planning (ERP) systems, in so much as they aim to integrate information available from the functional areas within the organisation. The ERP system however, details an enterprise-wide approach to information integration, instead of just providing an architecture for allowing communication between heterogeneous systems. The ERP approach is considered in more detail below.

4.3.2. Enterprise Resource Planning (ERP)

Companies such as SAP, Peoplesoft, J.D. Edwards, Oracle and Baan, are amongst the various providers offering Enterprise Resource Planning (ERP) systems. These systems provide the organisation with a multi-module approach to streamlining core business areas, such as finance, production and sales. ERP systems also help the organisation to efficiently manage its functional areas. For instance, product planning, inventory maintenance, customer service, dealing with supplier orders or tracking orders. Such systems are generally layered upon a relational database which is used for low-level data storage.

The need to harness and integrate data, in order to provide an efficient information flow throughout the various functions and levels of the organisation, can be seen as an argument for intra and inter organisational integration. It could be argued that ERP systems and Enterprise Application Integration (EAI) systems, provide an appropriate solution for this. However, the adoption of such systems have in the past been costly (Scheer & Habermann, 2000) and prone to operational inaccuracy (Sumner, 2000). Furthemore, in some circumstances, it may be feasible for various organisations to have different implementations of ERP systems. For example, a divisionalised structure may run its UK office with SAP and its USA office with Baan. Such inconsistencies may lead to problems of unconsolidated information.

From a tactical management perspective, the ERP system modularises the organisation's functional areas. However, as the system predominately supports the organisation's business functions, it would appear unsuitable for more complex tasks that require decision support. From this point of view, the span of control that the ERP system aims to exert, is enterprise-wide, and arguably is to large. Evidence for this can be derived from the creation of the EAI system, which was developed in part as an approach to improving ERP system layouts. The final approach considered in this section to delivering management information, highlights the use of integrative technologies. These are considered in more detail below.

4.3.3. Integrative Technologies

Developed by a consortium of developers at the Object Management Group (OMG), the Common Object Request Broker Architecture (CORBA), is a specification for managing, distributing and creating distributed program objects or components in a network. In essence, CORBA acts as an Object Request Broker (ORB). This means that all clients in a network that support ORB, can make requests for services from a server, without needing to know where the server is in the network. Similar to CORBA, the Distributed Component Object Model (DCOM) by Microsoft, also performs similar functions in the managing and distributing of program objects. Similarly, the Distributed Computing Environment (DCE), is another standard that allows communication between clients and servers in a network environment. These architectures are primarily used to integrate data exchange and communication within the client server architecture. Indeed, allowing information and program execution to be delivered throughout a network of distributed computers, paves the way for further integration throughout the organisation.

The use of Object Request Brokers (ORBs) within Distributed Transaction Processing (DTP) systems, allows for the business procedures that span the enterprise to be broken down into a set of processes. In addition, many Distributed Transaction Processing standards, such as those defined by the Open Group (XA, XCPI-C, TX, TxRPC, XATMI, Reference Model V.3), all aim to provide support for further integration within the distributed computing framework. DTP's support interoperability throughout the organisation, by providing an open environment that can support various client applications, such as databases, networks and legacy systems.

From the perspective of producing tactical information, the use of technologies listed in this section assist the connection of client PC's throughout the network of the organisation. Although this greatly increases the accessability of program objects between clients, it contributes little to the data modelling aspects and co-ordination requirements of information systems. In this instance, the existence of a middleware application is necessary in order to adequately model the organisations low level data, when supporting tactical management information.

4.4. CONCLUSIONS

Chapter three identified the use of Information Systems for providing information for supporting management decision making. This chapter has focused in greater detail upon the use and scope of information that is used by managers at the tactical level. Furthermore, it has also identified various technologies that are used for providing tactical management information. In outlining these technologies, this chapter has noted that these approaches are not wholly suitable for supporting the information needs of tactical management. Therefore, a further investigation into the data modelling approaches that are used when designing an information system would need to be achieved. This will be the emphasis of the next chapter.

This chapter also concludes section A. The purpose of this section, has been to delineate the wider perspectives of the tactical management role within the organisational context. In doing so, this section has focused upon three areas. Firstly, it has introduced the role of the tactical level manager and the environment in which they operate. Chapter three then directed attention towards the role of information systems and their use for supporting the tactical domain. Subsequently, chapter four highlighted the role of information in relation to the tactical management information need. Finally, as section A addressed the introductory issues of this study, section B will further develop this research by considering the various design approaches used to develop information systems, and their applicability for delivering tactical management information.

SECTION B

Approaches To Modelling Information Systems

B.1. INTRODUCTION

The distillation of section A highlighted the tactical management function in its environment, as well as outlining the wider perspectives of tactical level information, and the corresponding information systems that are used to deliver this information. The principle objective of section B, will be to narrow the focus of this study by considering in detail, various modelling methods that are used to more accurately model the business domain within the information system.

To do this, section B will be split into two chapters. These comprise;

- Chapter 5 - Modelling Management Information.
- Chapter 6 - Co-ordination Theory As A Modelling Framework.

An overview of these chapters are thus considered in more detail below.

B.1.1. Chapter 5 - Modelling Management Information

The computer based information system provides support for many management tasks. However, it is the design and architecture of this system, that in short, underlines both its usefulness and applicability. Such a view, is consistent with the relevant approaches that have been posited by various authors (Chen, 1976; Codd, 1970; Rumbaugh et al, 1991; Sutherland, 1995). The aim of this chapter therefore, is to provide a wide and comprehensive review of various static and dynamic data modelling methods, that are used to accurately model the information system in accordance with the real world domain that they represent.

B.1.2. Chapter 6 - Co-ordination Theory As A Modelling Framework

Various modelling methods discussed in chapter five, are useful for modelling not only the static areas of the organisation such as, *Customer_Name*, and S*upplier_Address*, etc, but also the more dynamic aspects, such as the behaviour and state of the customer and supplier object for instance. However, by themselves, these approaches add little to the analysis of processes that are carried out by tactical management. Therefore, chapter six will investigate a modelling approach named 'Co-ordination theory'. Developed at the Centre For Co-ordination Science (CCS), at the Massachusetts Institute of Technology (MIT), the co-ordination framework outlines an approach for analysing organisational processes. From the perspective of this study, ideas developed in the co-ordination framework, will be investigated in this chapter, in order to ascertain if they can provide a suitable approach for modelling tactical level tasks.

CHAPTER 5

Modelling Management Information

5.1. INTRODUCTION

Ideas taken from *data modelling,* suggest that the architecture of an information system should reflect the real world domain (Codd, 1970; Chen, 1976; Rumbaugh et al, 1991). In order to reflect this, the data model must possess both static and dynamic properties. Static properties signify for example, that a customer lives in a certain postcode area and has bought a particular product. Dynamic properties may suggest amongst other things that a customer orders *x* quantity of *product_A* from the sales department, and that despatch sends the ordered goods to the customer at 8.30am every morning. Approaches to modelling these functional processes of the business domain, have been realised in conventional data models. The more dynamic approaches to capturing business functionality, have been categorised using a number of methods, such as dataflow diagrams, sequence diagrams or activity maps. The principle aim of this chapter, is to provide a review of the various static and dynamic modelling methods, that are used to accurately model information systems in accordance with the real world domain that they represent.

5.2. CONVENTIONAL DATA MODELS

A data model is a collection of concepts used for describing data that is generally held within a database. This part of the research will focus upon two significant contributions to the data modelling field. These are:

- Relational data model (Codd, 1970).
- Entity Relationship data model (Chen, 1976).

Significantly, both these models allow for the identification of functional areas that are critical for modelling data at the database level.

5.2.1. The Relational Model

The relational model (Codd, 1970), usually accepted as a logical model, is made up of three parts, these are structural, manipulative and integrity. The structural part relates to definitions of relations or tables, the manipulative part to the operators that can manipulate those relations. The integrity part, details a set of rules that are required in order to represent semantics reflecting the real world domain. Furthermore, the model comprises rules that determine the management of primary and foreign keys. For example, all instances are identified by a unique primary key and all foreign keys should map to primary keys of the host relation. Both the relational model and the relational database however, lack semantic richness. For instance, the representation of classes, relationships and domains are usually represented as tables of data. The use of integrity constraints within the relational model ensures consistency. In essence, such constraints are usually made at the logical level. For example 'employee_salary is < £10,000'.

In analysing the constructs of the relational model, it is apparent that they provide a conceptual model of the problem domain. Depending upon the level of complexity, detailing a transition from the actual problem domain to the relational model, generally requires the intervention of an employee who has sufficient knowledge of the business domain. In addition to this, it is worth noting that in this context the relational representation does not adequately represent the problem domain. Furthermore, a lack of representation also occurs with regard to the range of possibilities represented by the relational functions, or the semantics that are provided by the integrity constraints.

5.2.2 Entity Relationship Model

In contrast to the Relational Model, the Entity Relationship Model (Chen, 1976), is generally accepted as a conceptual model. The number of elements that relate to the E-R model can be summarised as;

- **Entities**. Object representations such as customer or offices.
- **Attributes**. The attributes of represented objects (e.g. goods, address).
- **Relationships**. Used to model links between entities (e.g. customer bought goods).

An example of both the entity and relationship constructs are shown in figure 5.1 below. The diagram shows a range of entities such as 'Manager', 'Person', 'Office' and 'Building'. These entities have relationships with corresponding entities. For example, a 'Manager'entity is a 'Type of' 'Person' who 'Works in an' 'Office'. In this case, 'Type of' person who 'Works in an' are relationships. Attribute definitions would depend upon the context i.e. the managers name, the offices address, or the building location.

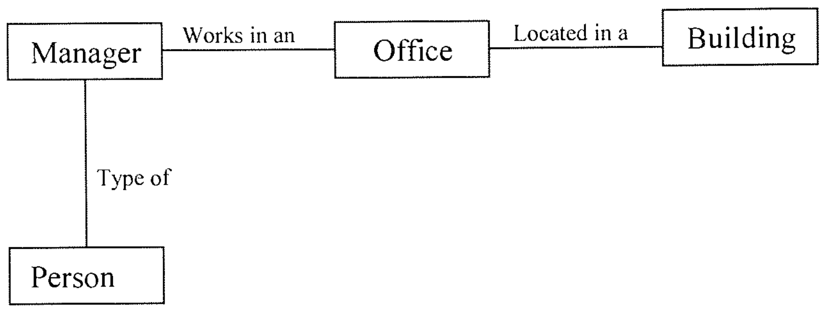

Figure 5.1. An Entity Relationship Diagram

As the E-R model has a relatively small number of constructs, it is fairly easy to use. However, analysis of the model realises a lack of rules detailing what actually constitutes an entity or an attribute. Furthermore, the relations construct is constantly used in order to signify the relationships between associated entities.

5.3. MODELLING ENTITIES AND OBJECTS

Codd (1979), proposed an extended relational model called RM/T. In addition, many authors have also proposed extended E-R models. It is also worth noting at this point that *Object Oriented Analysis* and *Design* methods, have also made a significant contribution to the data modelling paradigm. As a result of the significance and impact of extended relational models, a review of these will be considered.

5.3.1. The Extended Relational Model

In an attempt to capture more semantic detail, Codd (1979) developed the extended relational model. The revised model was named Relational Model Tasmania

(RM/T). The model made a significant contribution to the relational model in various ways. Some notable areas are listed below.

- An entity can be seen as 'any distinguishable object'. In this instance the object may be abstract or concrete.
- Entities may represent relationships.
- Every entity in the database is defined to be an instance of at least one entity type.
- Entities of the same type have the same attributes.
- Entities can have subtypes and supertypes.

In further developing the RM/T model, Codd identified three classes of entity. Each of these will be considered in turn.

Kernel Entity. Representing a real world object, examples include customer, employee, supplier.

Characteristic Entity. Describing an aspect of another superior entity, the characteristic entity is existence-dependent on the superior entity. This enforces integrity within the model. An example of a characteristic entity may be the maintenance history of a vehicle. Here, the maintenance history can only exist if the vehicle exists.

Associative Entity. An example of an associative entity may include the assignment of employees to a project. This entity represents a many to many relationship between two entities. An associative entity can only exist if all the entities that participate in the association exist.

Significantly, the RM/T allows for the identification of a class of an entity. This allows the operations appropriate to that class to be carried out. The RM/T model is capable of implementing specialisation abstractions owing to the concept of an E-relation, which holds the identifiers and entities which are members of a particular class, and a P-relation which holds attribute data. Furthermore, the use of holding identifiers and entities which are members of a particular class, enforces the ideas of object identity. A concept that was not outlined in the original relational model.

5.3.2. The Extended Entity Relationship Model (EERM)

As with the relational model, many authors have proposed enhancements to the E-R model. The E-R model originally proposed by Chen (1976), underlined two entity types. These were strong entities, that represented real world objects, and weak entities, which depended upon the existence of other entities. In addition, the use of cardinality constraints in this model, seems to be a more preferred alternative to using weak entities. Moreover, the addition of specialisation to the E-R model, has received much attention from various authors (Elmasri et al, 1985). Finally, a distinction has also been made between subset hierarchies and generalisation hierarchies. In this context, a subset hierarchy determines that it is possible for overlapping subsets to exist so that specialisation's of customer, for example, could represent an accountant or manager.

5.4. OBJECT ORIENTED ANALYSIS (OOA)

Object Oriented Analysis (OOA) (Coad & Yourdon 1991), underlines the use of classes and objects within the problem domain. In essence, the identifying of objects and classes of objects provides for the foundations of OOA. As opposed to the traditional functional view of a system, Object Oriented Analysis (OOA), is primarily concerned with the development of software engineering requirements that are expressed as a systems object model. The object model in this instance, is generally made up of a number of various objects that interact with each other.

When considering what types of object can be included in an object oriented model of the problem domain, Coad & Yourdon (1991) underline the following.

- Operational procedures: Problem domain procedures such as *order verification*, may be objects within the system, at a later stage of development there may be some of the procedures from the information systems itself, for example *user registration*.
- Structure: The relationships between objects within the system.

- Other systems and devices which connect with the main system described by the problem domain.
- Things or events remembered: Past events may be important objects in the system, even though they no longer exist or represent a transition in time.
- Roles played: Some objects within the problem domain, in particular human actors, may play several roles and although they are at one level, members of the same class (e.g. clients), may be sufficiently distinct in respect to the problem domain to need to be treated as different classes.
- Locations: Sites and geographical locations may be objects.
- Organisational units: Structures of the organisation will often be relevant objects, however it is necessary to be cautious of accepting them too early if process redesign is part of the project.

The list suggested by Coad & Yourdon is fairly extensive, in as much as it covers a broad range of object types. Establishing such a list will be useful when discussing the role of the Co-ordinated Business Object Model (CBOM) discussed in section C.

5.4.1 Object Characteristics

Objects posses various characteristics. Coad & Yourdon (1991), suggested that the properties of objects will generally include:

- Needed remembrance: the system needs to know things about an object.
- Behaviour: usually the object experiences change during the operation of the system under consideration.
- Attributes: usually an object has multiple attributes of interest.
- Instances: usually there is more than one instance of a given class.

In establishing the more dynamic aspects of objects, behaviour will be outlined in greater detail later in this section. Furthermore, object behaviour in relation to tactical level business processes will be considered in further detail in section C.

5.5. OBJECT MODELLING TECHNIQUE (OMT)

Alternative views as to what constitutes an object have been proposed by Rumbaugh et al (1991), as part of the Object Modelling Technique (OMT). Here, he states that an object can be defined as; "...a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand. Objects serve two purposes: They promote understanding of the real world and provide a practical basis for computer implementation. Decomposition of a problem into objects depends on judgement and the nature of the problem. There is no one correct presentation.". From this perspective, object decomposition will change in relation to the environment with which it represents. For example, a customer object in the retail industry, may have different attributes and belong to a different class than a customer object in the automotive industry. The remainder of this section will consider some of the more significant areas that are included within the OMT.

5.5.1. Classes

OMT notation makes the distinction between classes and objects. This is shown in figure 5.2 below. Having defined a class, it is clear that it is comprised of a set of objects that share a common structure and a common behaviour.

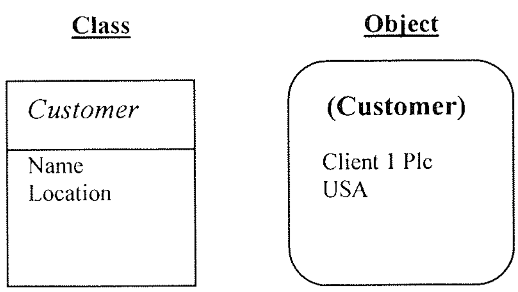

Figure 5.2. Class and corresponding object

Such a view is consistent with the definition proposed by Rumbaugh et al (1991); "An object class describes a group of objects with similar properties (attributes), common behaviour (operations), common relationships to other objects and common semantics.". Furthermore, when considering the role of objects in a class, Rumbaugh recognises that certain patterns of behaviour can be established between the object and its associated class; "Objects in a class have the same attributes and behaviour patterns. Most objects derive their individuality from differences in their attribute values and relationships to other objects.". Finally, the same author also recognises that the semantic relationship between many objects and one class, "share a common semantic purpose, above and beyond the requirement of common attributes and behaviour.". It is worth noting that the terms class and type are usually interchangeable. However, in order to be consistent with the literature, the author recognises that a class emphasises the classification of structure and behaviour, whereas a type places emphasis on the importance of conformance to a common protocol.

5.5.2. Associations and Links

A link can occur between two or more objects. It represents one instance of an association. Rumbaugh et al (1991) defines a link as; "a physical or conceptual connection between object instances". An association on the other hand, is a relationship between two classes. Associations usually denote a semantic connection. Rumbaugh et al (1991) defines an association as; "a group of links with a common structure and common semantics". The same author further points out that; "All links in an association connect objects from the same class." Figure 5.3 below, provides further clarity as to the properties of both associations and links.

Figure 5.3. Links and Associations, (Rumbaugh et al, 1991)

5.5.3. Object Structure

The principle dimensions of the object structure change in relation to the real world entities they are modelling. To merely treat all associations in the same way, would be inaccurate. From this perspective, the following associations will be examined in more detail.

5.5.3.1.Generalisation/Specialisation/Inheritance

Consistent with object oriented programming (Brachman & Levesque, 1985; Stefik & Bobrow, 1986; Wegner, 1987), objects can inherit properties from their generic parent classes. A class is made up of interacting objects. Furthermore, a class is a specialisation of another class. In this context, the notion of inheritance allows a subclass to inherit characteristics and properties from its superclass. The superclass can be regarded as a generalisation of the subclass or subclasses. In the diagram below, *Employee* is shown to be a generalisation of *Accountant*, *Technician* and *Director*. Vice versa, it can be realised that *Accountant*, *Technician* and *Director* are all specialisations of the class *Employee*. In this case, various characteristics such as *employee_id, employee_name* and *employee_address* may be inherited by *Accountant*, *Technician* and *Director* from *Employee*.

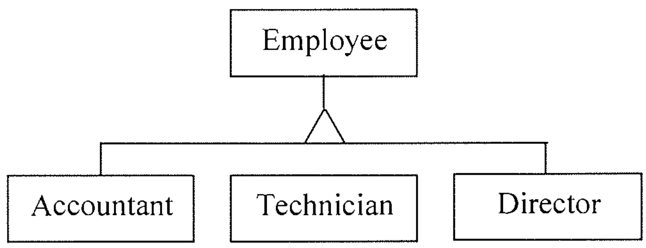

Figure 5.4. A Generalisation / Specialisation structure

5.5.3.2. Whole-Part Structures

The aggregation association in OMT, determines that an object may be composed of an assembly of other objects. Figure 5.5 below, provides an example of a whole-part structure.

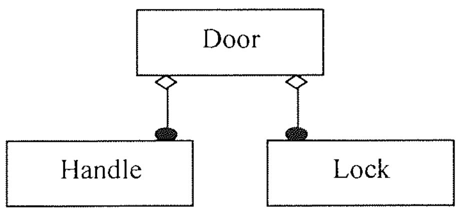

Figure 5.5. Whole-part structure

The example above denotes *Door* as an abstract class. It shows that it is composed of two other objects. These are *Handle* and *Lock.*

5.6. APPROACHES TO DYNAMIC MODELLING

Up to this point in the chapter, both conventional and object oriented data models have been considered. As a result of outlining these approaches, the static object model has emerged. This details clearly the role of objects and classes that have inheritance, relationship structures and aggregation. The static object model is therefore primarily concerned with structure. However, as the main emphasis of this study is to provide an information system that supports tactical level decision making, the challenge here is to model the dynamics of the real world domain based upon an object model that can handle change. From this perspective, it is not sufficient to simply model just the static characteristics of the object, but to consider in more detail the dynamic aspects as well. In contrast to the static model, the *dynamic model* is concerned with aspects such as, *Sequences of events*, *States that define events*, *Organisation of events and states*, *Sequencing of operations and time,* and *Events causing changes to the information system state*. Various properties that are linked with the dynamic model shall be considered in more detail throughout the remainder of this section.

5.6.1. States and Events; Process and Behaviour;

The *State*, and associated *Events*, as well as the *Process* and *Behaviour* properties of objects, effect changes and functionality of the information system. For example, it may be asserted that the triggering of *Events* by *Actions* change the S*tate* of objects, which can then in turn, change the B*ehaviour* of objects. These four areas, will be considered in further detail below.

State. The state of an object refers to the accumulative results of the behaviour of the object. Objects undergo changes over time. These are recognised by events such as deletions, additions or updates. The object has a state that is reflected in the information system. At a point in time, the state of an object encompasses all of the static properties of the object, plus the dynamic values of each of those properties.

Events. Examples of real events in the real world paradigm, must be tracked and kept up to date in the information system. For instance, a customer may order a part from a supplier. An update to a stock record would have to be kept by the information system, in order to reflect changes that had occurred in the supplier stock levels. This is shown in figure 5.6 below. The diagram describes certain operations that are characterised in the process of ordering parts. For instance, ordering a car part is managed via the client system, all data modelling in relation to that customer will be held within the data model. Finally, various database actions are executed, based upon the operations required. In this case, an update to the customer and stock record.

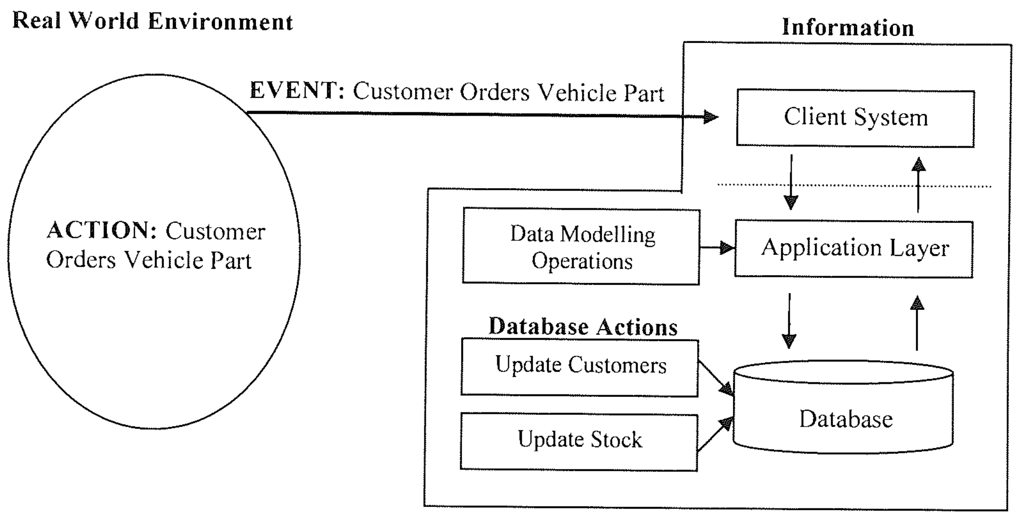

Figure 5.6. Object representations in the real world environment.

Figure 5.6 above, describes a dynamic model whose objects and data change as a result of an action. When modelling the real world paradigm however, it is possible for ambiguity to occur, in so much as the objects within an information system are designed to track changes in the real world, rather than provide an accurate representation of them. In addition, it is worth noting that this is not so important when discussing various attributes. For example, the attributes of both the supplier and the real world supplier are identical. However, it is necessary to outline such characteristics when discussing actions. In this instance, as can be recognised from the diagram above, there are two implicit actions. Firstly, that of a customer ordering a part and secondly, an update to the customer table in the supplier database. Here it is clear that both actions are fundamentally different.

Processes. If an action is to be denoted by the triggering of an event, then it is events that are responsible for making changes to the processes contained in the information system, that reflect the real state of the objects being modelled. In Object Oriented Programming (OOP), objects send messages to each other that trigger events. In contrast to this, events are also used to co-ordinate processes that link together the actions of individual objects. Such action helps to maintain integrity between the real world paradigm and the information system.

Behaviour. This term is used to describe how an object acts and reacts to state changes. The behaviour of an object differs in accordance to the complexity level of the object. For example, at a low level, behaviour translates into specific actions. At a higher level, the behavioural aspects of the object, relate more to the real world events that trigger change. Furthermore, interdependency between actions that trigger change, are also considered at this level.

5.7. UML AS A BUSINESS MODELLING TOOL

UML has become the industry modelling language for visualising, developing and documenting the constituent parts of an information system (Booch, Rumbaugh and Jacobson, 1998, Lunn, 2002). There are various standards and tools for mapping business processes, perhaps the most common one is the Unified Modelling Language (UML). The UML is the recognised modelling technique of the Object Modelling Group (OMG). In short, the UML provides a standard notation for modelling real world objects. Unifying Rumbaugh's Object-Modelling Technique (OMT), Jacobson's U*se Case Methodology* and Booch's methodology for describing objects and their relationships, the UML principally allows the user to model all the relevant static and dynamic characteristics of the proposed information system, by providing a number of modelling methodologies. For example, the *Business Use Case*, produces a high level assessment of the functionality of the business domain. The method represented in the *System Use Case,* provides information about the functionality of a computer system in relation to a particular business activity. *Activity Diagrams,* provide information about the flow of activities within the system. *Sequence Diagrams*, display information regarding the chronological order of events. *Collaboration Diagrams* focus upon the relations between objects within the problem domain whilst *Class Diagrams* display information relating to the object oriented makeup of the system. *Component Diagrams* are designed to group elements of a computer system into components. These components are then used in the development of the system. The *Deployment Diagram* describes the physical architecture of the system and finally the *Package Diagram* details relevant web server connections within the overall system.

As the role of UML is nowadays widespread, it is perhaps useful to investigate in further detail various approaches that adopt UML constructs, that have particular relevance upon the work contained in this thesis. In addition, it should be noted that there are many different products that aim to provide support for the modelling and implementation of business processes using the UML language. It is not the purpose of this section to list and discuss them all as this would be beyond the scope of this thesis. However, this section will draw upon some of the main contributions to this field, whilst outlining their applicability and relevance to this work.

5.7.1. The Catalysis Approach

Firstly, it is necessary to outline to the reader that the catalysis approach is sizeable and its constructs are deep and wide ranging. The purpose of this section is to provide to the reader a broad review of the salient parts of this approach, whilst drawing upon its relevance to this work. Catalysis describes itself as a "next generation approach for the systematic business-driven development of component-based systems, based on the industry standard Unified Modelling Language (UML)". Originally developed by D'Souza and Wills (1998). Information taken form the catalysis web site (Catalysis, 2003) suggests that purposes of the catalysis method are to:

- Help business users and software developers share a clear and precise vocabulary.
- Combine different business or component views into an integrated whole.
- Design and specify component interfaces so they plug together readily.
- Implement components by systematic assembly of other components.
- Develop greenfield applications using object and component technologies.
- Integrate heterogeneous and legacy components or systems into new development.

- Build high-confidence models and designs of business and technical components.
- Reuse domain models, architectures, interfaces, code, and processes based on patterns.
- Improve business processes with clear traceability to application development.

As the catalysis method is primarily concerned with the systematic business-driven development of component-based systems, it provides various constructs and indeed an overall approach for providing a business model that arguably could be used to provide support for tactical management. Evidence for this can be taken by the way catalysis provides improvements to business processes through application development. The design of components based upon a series of business processes is consistent with the component based approach to developing software in that it allows for a series of components to, in essence, be bolted/plugged into the overall software architecture. Such a system provides the facility for rapid development and upgrading of changing business processes. Such principals are becoming more popular as many software corporations are providing support for such software development. For example, Microsoft's COM and DCOM, as well as the OMG's CORBA. Such an approach to software development will prove fruitful when assessing the development of the Co-ordinated Business Object Support System (CBOSS) outlined in chapter eight.

5.7.2. Rational Rose

Rational Rose belongs under the hospices of IBM (Rational Rose, 2003). The main emphasis of rational rose is similar to that of catalysis in so much as the UML modelling language is used to provide more efficient systems that reflect changing processes in the real world paradigm. In short, it provides an environment for software modelling that raises the level of abstraction above the source code in order to make it easier for software developers to develop software in complex business environments. Rational rose as a development tool, is quite extensive and consequently offers different flavours of the tool for different software platforms. These tools however all posses the same architecture and design functions and therefore, this study will not review them all individually.

Similar to catalysis, Rational Rose provides for component based system development as well as providing an environment where mutual collaboration between employees can take place, in order to ensure that complex business issues are accurately represented in the eventual software design. Such emphasis on accurately capturing the particulars of the business model, allows for traceability from the eventual software solution to the actual business design.

5.7.3. Other Approaches That Use UML

As mentioned earlier there are many methods and tools that adopt the UML method. This section will simply draw upon a variety of perhaps the more familiar ones that are currently in use. Firstly, AllFusion by Computer Associates International, Inc. (CA), (AllFusion, 2003) offers a component based approach for developing and deploying e-business applications. In short, they provide an environment for simplifying the complexities of analysing, designing and implementing software applications by providing a tool that provides visualisation between the code and business process design levels. AllFusion provides a suite of tools that support these functions. However, this tool does not seem to offer anything that rational rose and catalysis does not. Secondly, Microsoft as part of their Visual Studio suite (Visual Modeller, 2003) support the design of component based architectures via the use of Visual Modeler. The suite of tools that accompanies Visual Studio, also allows the design to be implemented into a working piece of code. Again traceability from the design to the implementation stage is provided.

In concluding this section it is clear that all of the approaches that use UML are primarily concerned with the development of business driven component based applications. Moreover, tools that provide UML support are widely used by many organisations nowadays as part of the software design process. From this perspective, a great deal about the design and implementation of the CBOM in chapter seven can be ascertained. For example, the most appropriate design for supporting tactical management decision making, it construction, the use of a suitable component technology such as dynamic link libraries etc. Such areas will be outlined throughout section C.

5.8. THE BUSINESS OBJECT APPROACH

Up to now, this chapter has considered both static and dynamic object models. Such models are useful when modelling the real world functionality within the parameters of the information system. However, the approach set out within the domain of business objects, forwards more leverage for analysing specifically, the business functionality of the organisation. From this perspective, they harness the static and dynamic qualities of the various approaches mentioned so far, with the added capability of detailing process functionality. Therefore, this highlights certain advantages for tracking the series of process steps within the tactical management function.

In essence, the business object approach as noted by Jacobson (1996), acts as a presentation layer that brings together dispersed islands of data into a single coherent information package. In short, with its ability to model data based upon both static and dynamic behaviour, as well as business process properties, it provides the user with an accurate model of a particular business situation, in a real life business domain. From this perspective, the use of the business object is more appropriate for the context of this study than the traditional software object. Differences between the business object and the traditional object shall be considered in more detail throughout the remainder of this section.

5.8.1. Defining Business Objects

The academic literature does not seem to offer a standard definition of the term business object, however many suggestions have been posited. The Business Object Management Special Interest Group (BOMSIG), and the Business Object Domain Task Force (BODTF), both associated to the Object Management Group (OMG), define a business object as;

> "A representation of a thing active in the business domain, including at least its business name and definition, attributes, behaviour, relationships and constraints. A business object may represent, for example, a person, place or concept. The representation may be in a natural language, a modelling language, or a programming language" (Sutherland, 1995).

In differentiating between the business object and the traditional software object, the same author states;

> "A combination of state and a set of methods that explicitly embodies an abstraction characterised by the behaviour of relevant requests. An object is an instance of a class. An object models a real world entity and is implemented as a computational entity that encapsulates state and operations (internally implemented as data and methods) and responds to requests for services".

On considering these two definitions, it is apparent that the business object relates more closely to the representation "of a thing active in the business domain", that has identity and the ability to represent the more dynamic aspects of relationships and behaviour. For example, functions such as 'update stock record and send for despatch', or 'inventory is low, order more goods from purchasing', are suitable examples of the type of modelling characteristics of a business object. In short, the implied meaning of the business object denotes a conceptual view of a series of events or methods belonging to a specific business domain. The definition of a traditional software object on the other hand, is characterised as a software model of a real world entity, that necessarily, in this context, does not reflect a business domain. Gomes & Dias de Figueiredo (1999), note the difficulties in distinguishing between conventional and business objects. However, they consider, the OMG's definition to more closely embody the overall implied meaning of the business object concept. An object being an active thing, in specifically, a business domain. These differences between objects, business objects and associated business processes are described in figure 5.7 below.

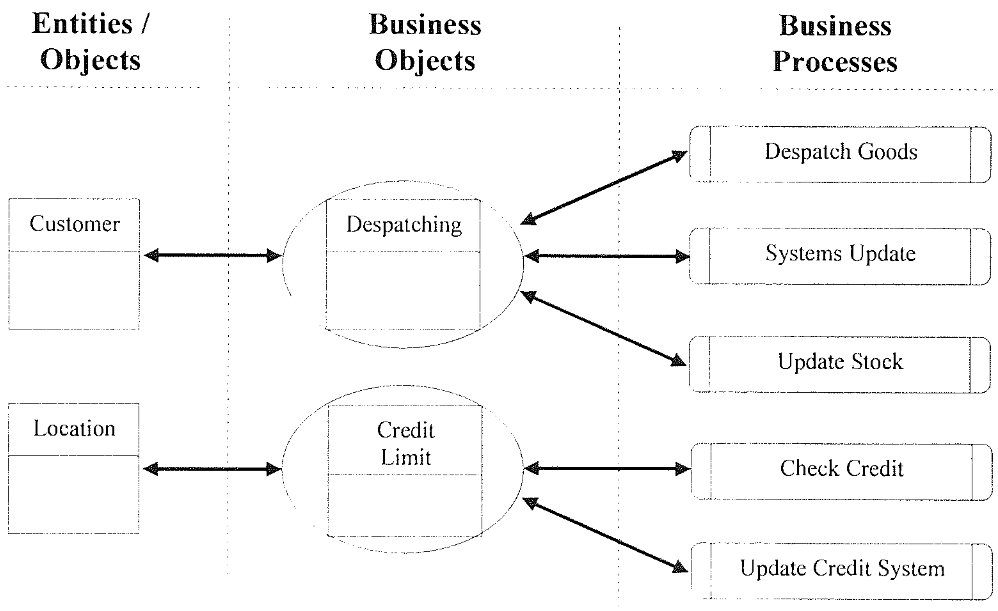

Figure 5.7, Objects, business objects and business processes

The figure above describes a business function describing two entity / objects, namely 'Customer' and 'Location'. Lines between these entity / objects, business objects and business processes, underline the interaction they have with each other.

Other attempts to define business objects have been made by various authors. Partridge (1996), denotes a business object as a generalisation from which an application class can inherit. For example, *person* rather than *employee* or *customer*, or *transaction* as opposed to *debit* or *credit*. Such a view denotes that the business object is centred around a more abstract depiction of objects in general. Moreover, it offers little in the way of delineating the more dynamic aspects of business objects. In contrast, Ramackers (1996) points to a business object being an entity in an entity relationship model. However, he recognises that the business object has dynamic properties by asserting that they have business rules and processes attached to them. From a system design perspective, a business object more often than not will belong to a much larger class. However, there are subtle differences that exist between these two areas. For example, both business objects and classes are used to describe certain elements, examples include *employees*, *suppliers* or *vehicles*. Both entities and objects however, can be generalised to provide entity types and object classes which represent similar entities and objects. Indeed, it has been proposed that the difference between an entity and an object can be recognised by entities being part of a static description, whereas objects can also represent part of a static description as well as defining a behavioural component (Benyon-Davies 1996: p.186). Furthermore, Ramackers (1996), does outline the dynamic role of the business object and therefore his interpretation is more consistent with Sutherland (1995).

Finally, Jacobson (1996) details a business object, as being the data structure that is gathered by an event or enquiry from several objects for display at the user interface. The same author highlights the presentation attributes of a business object, by providing the end user with a coherent display of information, from that which could be derived from dispersed data from various sources. For example, a customer order with all relevant details of the customer and all the orders they have made.

5.8.2. Business Process Objects

In providing further validation as to the overall encompassing definition of a business object, Sutherland (1995), underlines the notion of a *Business Process Object.*

> "A business object which represents a business process. A business process object is a specialisation (type) of a business object. The following words are intended to convey the difference between business object and business process object, but are not literal synonyms: action, set of steps, business verb, control object, routings, workflow".

Papazoglou, Riebbers, Tsalgatidou (2000) also recognise the existence of the business process object. In their interpretation, they define the business process object as, "a kind of active (or control) objects that bring together business objects to define a business process.". They characterise business process objects as, "a set of interrelated activities that collectively accomplish a specific business objective". In essence, a business object has both data and behaviour, a business process object, operates upon the properties of business objects. From this perspective, it changes the state and interaction of the business object within its environment.

The business process object requires business logic. For example, a business object may be built into the application in order to manage a process between two departments. In the example below, there are three processes that are to be managed, namely;

1) When order is confirmed the despatch department is notified.
2) The despatch department receives customer details.
3) The despatch department sends goods to the customer.

From a business process object perspective, the structured English to performing these operations may take the form;

```
Class Send_Goods(
Customer_Name As String
Sales_No As String
Despatch_No As String

      Sales{
      If Order Is Placed Then
            Notify Despatch_Dept
            Send Customer Details To Despatch_Dept}
```

```
Despatch{
If Order Is Received Then
      Get Customer_Details
      Send Goods To Customer})
```

The above example describes the business functionality of sending goods to a customer. The problem domain is set in a business environment and therefore, is consistent with the interpretation of the business object approach presented by both Sutherland (1995) and Papazoglou, Riebbers, Tsalgatidou (2000).

To summarise, the concept of this type of object, rests within the scope of the business object. In essence, it is a "A business object which represents a business process" (Sutherland, 1995). From this perspective, this study will view the business object as an all encompassing object, that holds information regarding not only the static and dynamic attributes of the business domain, but also its process dynamics. Therefore, in order to eliminate ambiguity, the term business object shall be used to represent the full functionality of the business object including its corresponding process functionality. Therefore the term business process object shall only be used when specifically required.

5.9 CONCLUSIONS

Having considered in section A, the role of information systems and their uses for providing management information. This chapter has narrowed the focus of this thesis by highlighting various modelling methods that are useful for representing the real world paradigm within an information systems architecture. Specifically, in achieving this, chapter five has outlined both static and dynamic approaches to object modelling. Moreover, it has introduced the concept of the business object, as suggested by the OMG, as well as detailing the uses of the business object approach for representing specific business domains.

In concluding this chapter, it is important to outline why the business object approach will be adopted throughout the remainder of this study instead of other approaches that have been mentioned in this chapter. In essence, this chapter has outlined the role of entities and objects for data modelling. Moreover, it has also outlined various data modelling approaches using both entities and objects. It is possible to derive from this chapter that the earliest approaches to data modelling detailed the use of entities, such as the Relational data model (Codd, 1970) and the Entity Relationship data model (Chen, 1976). These approaches however, have various limitations, for example the Entity Relationship data model realises a relatively small number of constructs, whilst a lack of rules detailing what actually constitutes an entity or an attribute have become apparent. The Relational model on the other hand does not adequately represent the problem domain. Also a lack of representation also occurs with regard to the range of possibilities represented by the relational functions, or the semantics that are provided by the integrity constraints. In order to overcome these issues various extensions to these model were introduced as the 'The Extended Entity Relationship Model (EERM)' and the 'The Extended Relational Model'.

As opposed to the traditional functional view of a system, Object Oriented Analysis (OOA), is more concerned with the development of software engineering requirements than the entity based approach mentioned above. In essence, the object oriented approach outlined a system of classes and objects that interact with each other at the data level. In essence, this takes areas such as 'Operational procedures', 'Object structure' and 'Object behaviour and states' into consideration. Various authors outlined throughout this section have made various contributions to the object oriented approach. Perhaps the most notable in recent times is that made by Sutherland (1995). Here he outlines the role of the 'Business Object' as a modelling approach. As this thesis has pointed out, there is a subtle difference between the notion of an object and a business object. In short, a business object is one that exists primarily in the business domain. From this point of view the same author also extended this notion of business objects by introducing the role of the business process object. As a "A business object which represents a business process". Here Sutherland outlined that the difference between a business object and a business process object can be established by observing the synonyms that describe the business process object, i.e. "action, set of steps, business verb, control object, routings, workflow". From a tactical level perspective, tactical level processes are carried out by tactical level management in an organisation. Therefore, from this point of view it is more accurate to adopt the term business object in the context of this study, as it more accurately represents the domain that this thesis is concerned with.

However, despite the fact that these objects detail process functionality, their constructs revolve around events changing states and behaviour. They do not provide any functionality for analysing tactical level processes in any greater depth than other dynamic modelling approaches. From this perspective, a higher level approach to returning information based upon tactical level processes is needed. Such an approach will be detailed in the next chapter.

CHAPTER 6

Co-ordination Theory As A Modelling Framework

6.1. INTRODUCTION

The last chapter examined various static and dynamic data modelling methods that are considered useful when modelling an information system. It also presented the business object modelling approach for serialising and representing business processes within the business object paradigm. From this perspective, capturing static information regarding tactical management processes is a relatively easy task, as many entity and object modelling methods provide the necessary constructs for achieving this. Their functionality is primarily concerned with objects and their behaviour. However, the effectiveness of the business object model for representing the real world domain, is largely dependent upon the accuracy of the information systems model. Therefore, if the model is inaccurate, then the information that is returned will be inadequate for supporting management decision making. As this study is primarily concerned with the tactical management level, this chapter will present the co-ordination framework as an approach for supporting the accuracy of an information system for supporting tactical management decision making. Specifically, as the co-ordination approach outlines a framework for analysing organisational processes, the emphasis is placed on serialising processes rather than serialising objects. Therefore, ideas leveraged from the co-ordination framework can be adapted for modelling tactical level processes. The aim of this chapter therefore, is to outline the principle dimensions of the co-ordination framework and its associated constructs.

6.2. CO-ORDINATION THEORY – AN OUTLINE

The basic idea that underpins the notion of co-ordination, in its simplest form, details a framework for better understanding activities within a given process. The study of co-ordinating activities has for some time been the focus of many research disciplines. For example, references to work in the field of systems theory and cybernetics are quite extensive (Ashby, 1956; Beer, 1967; Boulding, 1956; Emery, 1969; von Bertalanffy, 1950; Wiener, 1948). This also applies to the field of co-ordinating human beings in organisations (Galbraith, 1977 and Thompson, 1967). An examination of co-ordinated activities in markets has been outlined by Debreu (1959). Studies into the existence of co-ordination in artificial intelligence has been carried out by authors such as Fox (1981); Hewitt (1986); Huberman (1988); Miller and Drexler (1988) and Smith and Davis (1981). In the field of information technology, Mentzas (1993), noted the existence of a link between co-ordinating organisational tasks and decision making. Finally, the study of co-ordination in the field of operations and information management has been conducted by Whang (1995). Given the range of applicability of the term 'co-ordination', it is not surprising that there are a number of definitions emphasising different aspects of the co-ordination approach.

At this point, it is important to outline the specific focus of co-ordination theory within the scope of this study, and in doing so provide reasons as to why the co-ordination approach better serves the needs of this work. For the purposes of providing a suitable response to the problem area (see section 1.2), the definition proposed by Malone and Crowston (1990; 1994) will be used as the terms of reference in the area under investigation.

In essence, their approach underlines the use of co-ordination in an organisational context. They define co-ordination theory as; "The act of managing dependencies between activities performed to achieve a goal". The same authors also demonstrate that co-ordination implies that "all instances of co-ordination include actors performing activities" Consistent with Malone and Crowston, Singh, (1992), details co-ordination as "The integration and harmonious adjustment of individual work efforts towards the accomplishment of a larger goal". From this perspective, organisational goals are achieved by the execution of smaller processes. In short, co-ordination theory provides an understanding as to the architecture of organisational processes. It describes activities belonging to processes and thus presents an approach for managing dependencies between these associated activities.

Malone and Crowston (1994) therefore, distinguish their approach to co-ordination in contrast to others, by using the term 'Co-ordination Theory', which presents a framework that details the management of interdependencies between organisational activities (Hayashi and Herman, 2003). From this point of view, it may argued that this is not a 'theory' in classical terms, but more of a framework or an approach for better understanding the role co-ordination in an organisational context.

In order to clarify the use of the co-ordination framework within this research study, the remainder of this section will outline the reasons as to why this approach will be applied whilst also outlining how this approach better serves the needs of this work. Firstly, the notion of co-ordinating organisational processes has been around for some time (March and Simon, 1958). In addition, and as will be stated throughout this thesis, many authors have contributed towards the notion of co-ordination. For example, Thompson (1967) outlined the role of interdependencies in activities as did Crowston (1994). Zlotkin and Wyner (1995) posited their view as to the existence of various dependencies that naturally occur within organisational processes. Moreover, Malone et al (1999) have gathered much of this research together and applied it to a software application called the process handbook. Although work presented by these authors will be expanded throughout the course of this chapter, it is clear that by analysing the research that the contribution made by Malone and Crowston seems to reflect a synthesis of much of this past research. Hence providing for a rich vein of research. Moreover, it is also clear that their work is the most up to date and indeed cutting edge in the field.

Subsequently, it is important that this research fully examines the contribution made by Malone and Crowston. It is however important to also distinguish between their contribution and the intended contribution made by this research study. This is relatively clear, in so much as the co-ordination concepts developed by Malone and Crowston have arguably been applied at a workflow or in terms of figure 2.1 an operational level. This study however, will aim to extend these concepts and encapsulate them into a functional object model that can be used to support tactical decision making at the tactical level in the organisation.

The third point to emphasise in outlining why the co-ordination approach has been adopted by this study, is owing to its functionality when representing organisational processes. In short, the co-ordination approach developed by Malone and Crowston provides a framework that not only represents organisational processes, but also categorises smaller activities as belonging to composite processes. In addition to this, the notion that activities use or produce resources gives rise to the role of dependencies between activities, in so much as one activity within a process is dependent upon the output of another.

Finally, approaches that were outlined in section 4.3 such as integrative technologies and collaborative tools, are amongst the various approaches that are used to provide a more efficient flow of information to the tactical level manager. However, such approaches rely upon connecting already established client PC's in order to add further integration of the organisation's information supply. From this perspective, such approaches contribute little to the data modelling aspects of information systems. Moreover, such approaches also do not provide any opportunity to apply co-ordination concepts to the individual information systems design.

6.3. THE CONSTITUENT PARTS OF CO-ORDINATION

The co-ordination framework as outlined by Malone and Crowston (1994), has over the past decade, been adopted and applied to various research areas by many authors. However, despite the popularity of this approach, the literature does not present a comprehensive categorisation of the component parts of the co-ordination framework for the purposes of representing co-ordination constructs in a diagram. Therefore, this study has had to develop a diagramming technique that has the ability to represent tactical level processes along with the appropriate co-ordination constructs. Table 6.1 below, represents these constructs, along with a corresponding notation.

The notation presented in table 6.1 below, where applicable, will be used to portray diagrams that require co-ordination constructs. In order to fully investigate the theory of co-ordination in more detail, the areas listed in the table below will be explored throughout the remainder of this chapter.

Co-ordination Construct	Notation	Function
Activities		
Breakdown of Process	A	Defines sub-components of a process
Dependency Types		
Sharing	S	Two tasks shares one resource
Fit	FI	Two resources needed to achieve task
Flow	FL	Task *n* makes resource *y*. Task *x* uses resource *y*
Resources		
Resource	R(n)	Denotes the Resource and its ID
Produces		
Produces	PRODUCES	If present, denotes that the resource is produced by its associated activity
Uses		
Uses	USES	If present, denotes that the resource is used by its associated activity
Actors		
Actor	$A_{(t)}$	Denotes an actor and its ID.
Time Constraint		
Time Constraint	T	Activity must be completed within a certain time

Table 6.1. Co-ordination elements and notation

6.4. GOALS, PROCESSES AND ACTIVITIES

Goals, processes and activities, are amongst some of the elements that make up the co-ordination approach. In order to provide the reader with a full description of the co-ordination framework, this section will briefly discuss the role of goals, processes and activities, whilst also linking their role to the needs of this work.

Within the co-ordination theory domain, goals, processes and activities are inter-linked with each other. Goals are set by organisations in order to achieve an ideal state, whilst processes are devised by management and implemented throughout the organisation's workforce, in order to achieve set goals. Activities on the other hand, are carried out by employees in order to realise the successful accomplishment of these processes. An attempt to track and model these arrangements of processes and activities have been suggested by Malone et al (1999). Here, they describe a concept called the 'Process Compass' which can be used to analyse business processes. Concepts used in the process compass are implemented in the 'Process Handbook' (Malone, Crowston and Herman, 2003)

Figure 6.1 below, presents the principle dimensions of this model. The diagram represents 'parts of' a process, in the vertical axis and the different 'types', in the horizontal axis. The vertical axis is very similar to the concept of aggregation detailed in *object orientation.* The idea behind the model, determines that the process compass can be used to analyse activities in a process from a conceptual perspective. Contained in a series of directional movements, these are summarised below.

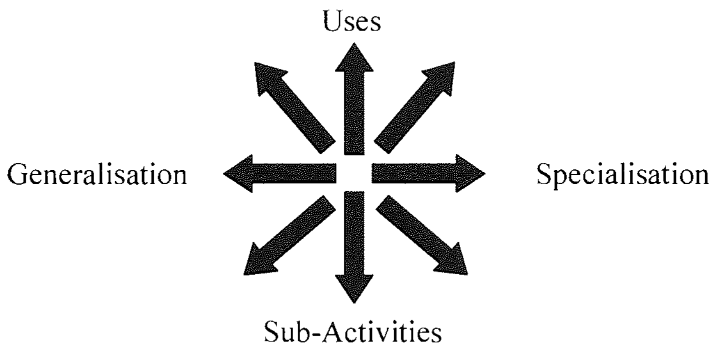

Figure 6.1. The Process Compass

- Uses (UP). To the larger activities of which this one belongs to.
- Sub-Activities (DOWN). To Sub-Activities that belong to the current process.
- Generalisation (LEFT). To different activities of which this one is a type.
- Specialisation (RIGHT). To different types of the activity.

The process compass provides a great deal of flexibility for analysing the behaviour of processes. Its contribution is to provide more information about the process flow, rather than simply stating that 'process_1' has four steps A, B, C and D. Furthermore, it is also worth pointing out that the process compass differentiates between a process and an activity. Albeit subtle, a process is made up of various activities. For example, a process 'Despatch Product', may be made up of the activities, 'Sign Despatch Order', 'Update Stock Record', 'Send Product To Client'.

As detailed above, the concepts described within the process compass detail four different types of relationships that processes and activities have with each other. From this perspective, such ideas provide a good insight into the relationships between processes and activities. Such ideas will prove useful when formulating an object model supporting tactical management decision making.

6.4.1. Process Generalisation and Specialisation

Chapter four outlined the differences between strategic, tactical and operational level information. Such information from a managerial perspective is used to execute activities. Such activities however, as detailed in the process compass above, can belong to more composite activities. Furthermore, the same activities can also have sub-activities belonging to them. As this study is primarily concerned with the tactical level, it is worth investigating the notion of generalisation and specialisation in more detail. This is for two distinct reasons. Firstly, it is perceivable that tactical level activities can have sub-activities. Secondly, if objective six outlined in section 1.5 is to be achievable, then it would seem necessary to investigate this area outlined by the process handbook.

Using the notion of *generalisation* and *specialisation* from *object orientation* to analyse organisational processes, represents a shift from analysing objects. Figure 6.2 below, delineates an example of the process 'Sell Something'.

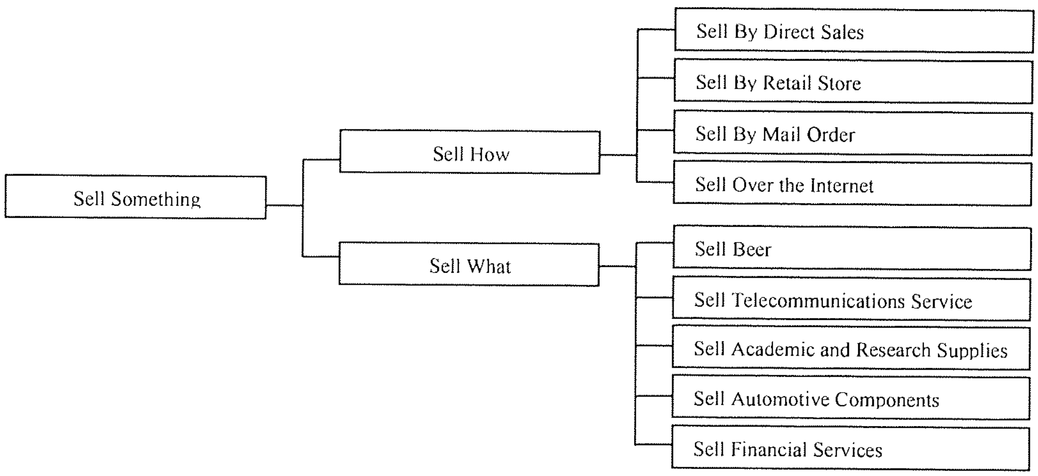

Figure 6.2. Specialisation of processes, adapted from Malone et al (1999)

Adapted From Malone et al (1999), the figure above shows specialisation and generalisation relationships amongst processes. For example, by moving from left to right, signifies a specialisation of processes. In contrast by moving from right to left, shows generalisations of activities. In this case 'Sell How' can be viewed as a generalisation of 'Sell By Direct Sales', and 'Sell By Retail Store'.

6.4.2. Inheritance Relationships Within Processes

To assert that tactical level processes can be decomposed into smaller processes, or can be grouped together to form more general abstract processes, would outline the existence of a parent / child relationship between the properties of processes. Undoubtedly, this draws comparisons with the notion of inheritance, suggested in *object orientation.* In this context, sub-processes inherit properties from their generic parents. Such flexibility not only provides for greater analysis of organisational processes in regards to their 'type of' relationships, but also highlights the hierarchical relationships that activities and processes have with each other.

6.4.3. Activity Timespan

Another characteristic of co-ordination issues, refers to the timespan of activities. Dittrich (1991), suggested that the length of time required for an activity, reflects the management level that will carry it out. For example, any activity that can be carried out within a timespan of seconds to hours would typically be carried out by operational level staff. An activity that took from days to months would be the typical time length of an activity that would be carried out by tactical management. Activities that could take up to a year would generally be carried out by strategic level staff.

Influenced Timespan	Activity Objective	Examples
Seconds-Days	Operational Control	Establishing a conference
Hours-Months	Tactical Control	Preparation of an agenda
Months-Years	Strategic Control	Activity planning

Table 6.2. Timespan of co-ordinated group activities, adapted From Dittrich (1991)

In concluding this section, goals processes and activities will feature more prominently throughout the remainder of this thesis, owing to their importance in relation to the co-ordination framework.

6.5. RESOURCES AND RESOURCE FLOWS

Section 6.2 above, detailed that co-ordination is centred around managing interdependencies between activities. Such interdependencies are generally outlined as one activity relying upon the output of another. From this point of view, it is the output of the resource that an activity may be dependent upon. If therefore, an issue regarding the production of a resource is apparent then an interdependency will exist. The co-ordination framework as defined by Malone and Crowston (1994), does not go into detail as to what constitutes a resource. From this perspective, resource properties will change in relation to the problem domain. For example, Dellorocas (1996) developed a resource taxonomy that focuses primarily upon a co-ordination perspective for integrating software components. Therefore, some of the resource elements listed in the diagram below are specifically centred around this approach.

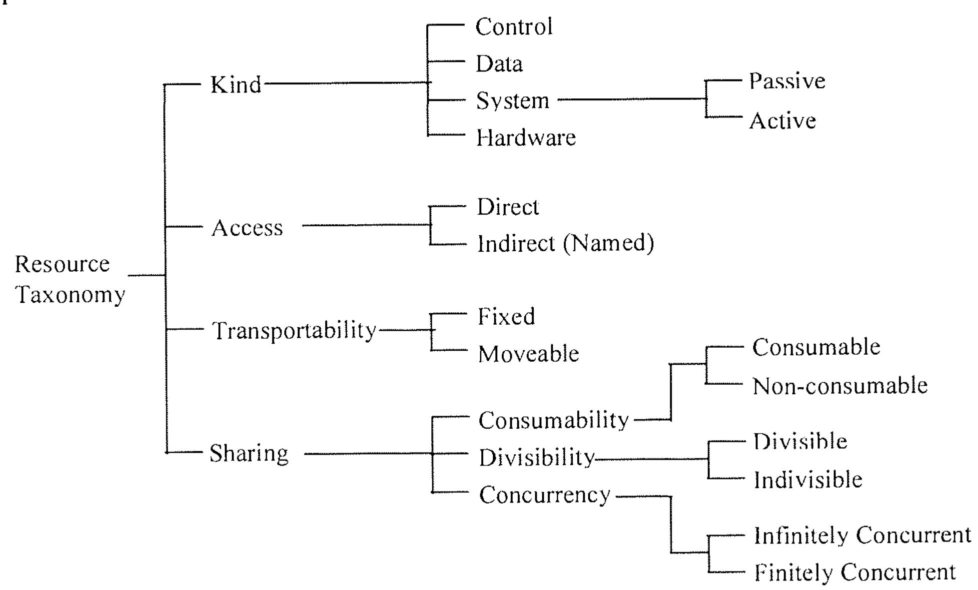

Figure 6.3. A taxonomy of resources, adapted from Dellorocas (1996)

Despite the diagram above detailing a resource model for integrating software components, various elements that comprise the model, can be considered to apply to most resources. Most notably, 'Kind', 'Sharing', 'Consumability', 'Divisibility' and

'Concurrency'. An overview of the principle areas of the above diagram however are considered in more detail in tables 6.3, 6.4, 6.5 and 6.6 below.

'Kind' Properties	Function
Control	The controller behind the execution. From a tactical management perspective, this would refer to the activity.
Data	This refers to a resource that is specifically data in type. For example, sales data or production data.
System	System resources are offered by non-human systems such as operating systems. Two types of system resource are identified as Passive and Active.
Passive	Passive resources are identified as shared memory.
Active	Named servers are identified as active resources.
Hardware	Refers to a resource that is hardware based. For example, raw materials or machinery.

Table 6.3. The 'Kind' attributes adapted from Dellorocas (1996)

The first area of the taxonomy details the 'Kind' attributes. These refer to the kind of resource under review. Three further sub-categories are identified as belonging to the 'Kind' property, these are, 'Control', 'Data', 'System' and 'Hardware'. Moreover, 'System' has two further properties. These are, 'Passive' and 'Active'. These are shown in Table 6.3. The 'Access' property listed in table 6.4 below, details how producers and users access resources. Two further sub-properties are identified. These are 'Direct' and 'Indirect'.

'Access' Properties	Function
Direct	Data resources are communicated from producers to users.
Indirect (Named)	Data resources that are communicated directly from producers to users.

Table 6.4. The 'Access' attributes adapted from Dellorocas (1996)

The 'Transportability' property detailed in table 6.5, outlines the manoeuvrability of a resource. Two further sub-properties are listed as belonging to the 'Transportability' property. These are, 'Fixed' and 'Moveable'. These are highlighted in more detail in the table below.

'Transportability' Properties	Function
Fixed	Fixed resources have one location in the system. Hardware resources are fixed.
Moveable	Moveable resources can be moved. For example, data or information.

Table 6.5. The 'Transportability' attributes adapted from Dellorocas (1996)

Various authors who adopt a co-ordination theory approach to organisational process analysis, have detailed the role of resource shareability (Hayashi and Herman, 2002; Yoshioka and Herman, 1999; Zlotkin, 1995). The final resource property to be considered, outlines 'Sharing'. This is detailed in table 6.6 below.

Sharing	Function
Consumability	Determines if the resource is being destructively consumed.
Consumable	A consumable resource can not be used many times. For example, raw materials.
Non-consumable	This type of resource can be consumed many times, such as data, software programs, or plans.
Divisibility	Determines if the resource can be divided between activities.
Divisible	The resource can be divided i.e. 70% to activity 1 and 30% to activity 2.
Indivisible	The resource can not be divided i.e. only one activity can use the resource at a time. For example, a printer can only print one job at a time.
Concurrency	Determines if a resource in use can be used by more than one user at a time.
Infinitely Concurrent	Determines that more than one user can use the resource at the same time.
Finitely Concurrent	Determines a set number of users who can use the resource at any one time.

Table 6.6. The 'Sharing' attributes adapted from Dellorocas (1996)

Resource shareability, as mentioned previously, describes a resource that can be shared between many activities. For example, information being shared between many employees. Figure 6.4 below presents examples of both shareable and non-shareable resources. The shareable resource, is demonstrated by Activities 1 and 2, both making demands upon Resource 1.

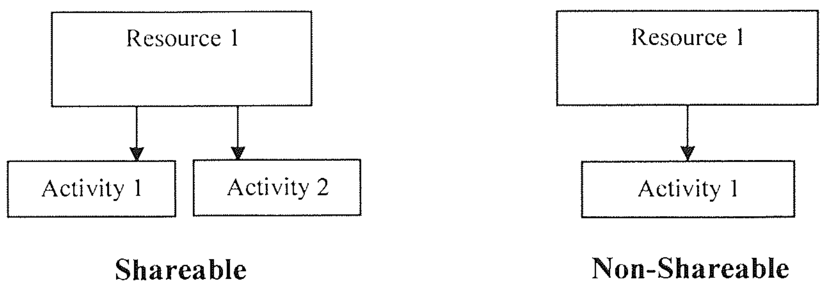

Figure 6.4. Shareable and non-shareable resources

Consumability on the other hand assumes that something can be used destructively, for instance raw materials. The notion of consumability, denotes that a supply and demand relationship exists between the activity and the resource.

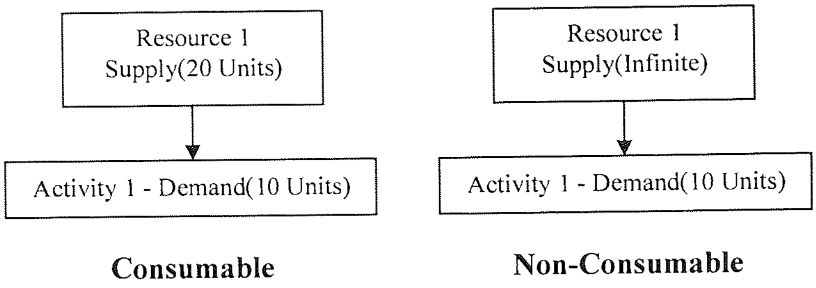

Figure 6.5. Consumable and non-consumable resources

The diagram above shows both consumable and non-consumable resource types. The arrows leading from the resources are directed down towards the activities, displaying the flow of the resource. In this case, the consumable resource has a finite supply, determining that it can only supply twenty units at a time. A non-consumable resource on the other hand has an infinite supply, and can be used ad infinitum. Reusability can also be outlined when considering the consumability of a resource. Reusability in this context, refers to how many times an activity can use a resource over a period time (von Martial, 1989, p.62). For instance, raw materials such as petrol and fuel cannot be reused. Resources such as data or information in contrast, are reusable.

Resource concurrency dictates how many users can use a resource at any one time. A resource that is 'Finitely Concurrent', allows a set number of users to use the resource, 'Infinitely Concurrent', outlines that an infinite amount of users can use the resource at any one time. An example of these types are shown in the diagram below.

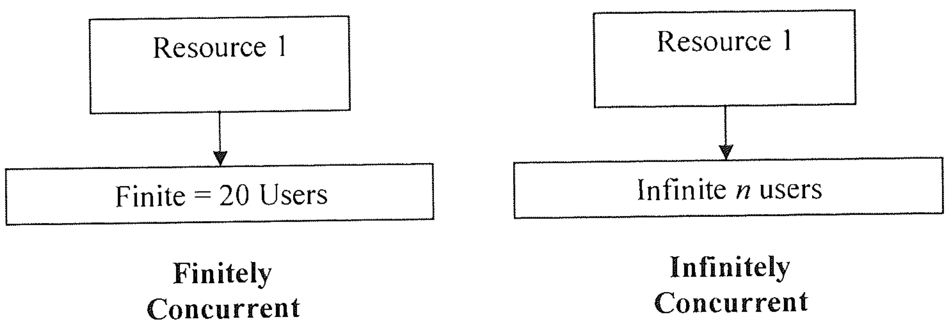

Figure 6.6. Finitely and Infinitely concurrent resources

Dellorocas notes that the notion of 'Consumability', 'Divisibility' and 'Concurrency', was presented in a much earlier study by Wyner and Zlotkin (1995). Although figure 6.3 delineates a relatively thorough explanation of the component properties of a resource, it does not take into account the time-perspective. From this point of view, no measurement as to how long a consumable resource will last can be judged. Furthermore, there is no way to record the actual measurement of the resource i.e. its quantity, although, in some cases, it is conceivable that this may not be needed. The taxonomy does not provide any indication as to the supply or origin of the resource. From a tactical management perspective, resources may be of different types. As mentioned earlier, resource types will be dependent upon the actual purpose of the process. The taxonomy presented by Dellarocas therefore, presents some areas that can be considered useful when highlighting what may be termed the properties of a 'tactical level resource'. Up to this stage in the chapter, it has been outlined that activities use resources. It has also been highlighted that activities are dependent upon the use of resources. The next section shall consider the role of actors, who, under the co-ordination framework, are responsible for the execution of activities.

6.6. ACTOR ASSIGNMENTS

Crowston (1994), outlines two types of actors, namely; 'Specialists' and 'Generalists'. Specialists are actors that perform activities that require a certain degree of specialisation. Generalists on the other hand, perform activities that are general in nature. As Crowston himself points out, "any actor can perform the task". Crowston's suggested idea of actors details a mechanism that is structured by skill level. However, as Crowston points out, "In reality, things are rarely so precise, [and more often than not] organisations will be located along the spectrum between these two extremes". Crowston, in detailing his idea of 'Specialists' and 'Generalists' does not strive to make any distinctions between actors that are human or systemic in type. Actors are merely distinguished by skill level. Other authors however, have made distinctions between actors that are human (Kim, 2000), and systemic in type, such as software components (Dellarocas, 1996). Human actors are assigned responsibilities that vary in levels of complexity. Therefore, as with employees in an organisation, actors can be categorised as being either strategic, tactical or operational. However, from this perspective, it is difficult to classify a systemic actor in the same way, therefore, the systemic actor in the context of this study, is an actor that is part of either a strategic, tactical or operational level process.

6.7. DEPENDENCY TYPES

Dependencies exist between activities. In essence, dependencies determine that one activity is dependent upon the production of another. Over time, many researchers have concentrated on the existence of dependencies between activities. However, it could be argued that the focus has been placed primarily upon the describing of dependency patterns, rather than explicating the effects of a dependency in relation to what actors can and should do (Crowston, 1989; 1994). The numerous approaches that have been proposed in relation to the study of dependencies, tend to highlight the dependency from different paradigms. For example, Pennings (1974), sets down the view that dependencies were related to four different sources. Notably, 'Task', 'Role/Position', 'Social' and 'Knowledge'. Moreover, Thomas (1957) noted that dependencies between activities can be of a competitive and facilitative nature. In 1987, Victor and Blackburn defined a dependency as an, "extent to which a unit's outcomes are controlled directly by or are contingent upon the actions of another unit". Such a definition is consistent with McCann and Ferry (1979); "when actions taken by one referent system affect the actions or outcomes of another referent system". However, within the context of this study, the dependency is a channel that links activities. These channels allow resources to pass back and forth between activities. In essence, dependencies exist where either a resource is dependent upon an activity or an activity is dependent upon a resource. This concept of dependencies draws parallels with Zlotkin (1995). Shown in figure 6.7 below, three different types of dependency patterns are categorised. These are 'Fit', 'Flow' and 'Sharing'.

Figure 6.7. Dependency types, (Zlotkin, 1995)

The three different types of dependency displayed in figure 6.7 above will be considered in further detail below.

6.7.1. Fit Dependency

A 'Fit' dependency, occurs when two tasks contribute to the same resource. For example, for a corporate budget to be produced, all of the organisations departments must produce their own individual budgets. Given two resources (R_1) and (R_2) and one activity (A_1), a fit dependency can be expressed as ((R_1) + (R_2)) / (A_1). In essence, for (A_1) to execute, both (R_1) and (R_2) must be available.

6.7.2. Sharing Dependency

A 'Sharing' dependency refers to a resource that can be shared between two or more activities. For example, a prototype software product could be provided to one or more partners who have to test the conformity of their add-ons before the completed design can be approved. Given two activities (A_1) and (A_2) and one resource (R_1), a fit dependency can be expressed as (R_1) / ((A_1) + (A_2)). In this context both activities (A_1) and (A_2) use (R_1). An example of a shared dependency is shown in figure 6.8 below.

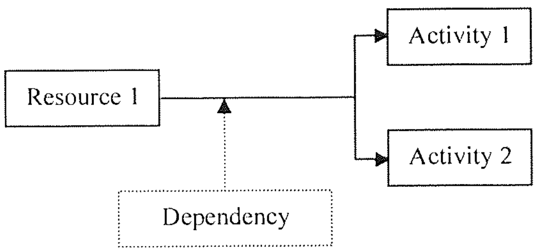

Figure 6.8. A Shared dependency

6.7.3. Flow Dependency

A 'Flow' dependency refers to an activity that uses a resource, which depends upon another activity. For example, a hardware product provided by one partner has to be customised by another before being delivered to the client. In this context, given two activities (A_1) and (A_2), and one resource (R_1), a flow dependency can be expressed as (A_1) + (($R1$) / (A_2)). Therefore, activity (A_1) produces (R_1), and (A_1) uses (R_1).

6.7.4. Constraint Types

In addition, Malone et al (1999), notes three different types of constraints that apply to 'flow' dependency types. These are 'usability', 'accessibility' and 'pre-requisite'. These constraint types are considered in greater detail below.

6.7.4.1. Usability

The constraint type 'Usability', determines that a resource must be made available when the activity requires it. It could be suggested that the element of usability, not only distinguishes between time (the item should be available for use), but also one of suitability (the item should be fit for the purpose with which it is to be used). Although time scheduling and synchronisation are ways in which this particular dependency can be managed, it should be noted that in order to further reduce iterations within the process, an understanding as to the process design should be considered. In this context, it is necessary to underline the factors that cause usability dependency constraints and how they can be managed. Dellorocas (1996), listed five points outlining those responsible for ensuring usability. These are defined as; 'Designers', 'Producers', 'Consumers', 'Producers and Consumers' and 'Third Parties'.

6.7.4.2. Accessibility

An 'Accessibility' constraint type, requires the resource to be in the 'right place'. In essence, it must be made available for use. As with a usability constraint, the management and co-ordination of the producers and users is crucial if accessibility is to be achieved (Dellorocas, 1996; Malone et al, 1999). In essence, the resource or item that passes between a producer and user can only be accessible if the user activity has access to it. For example, a user may not need to mount an engine in a car, but the user activity may need access in order to check the engine number first. In this instance, placing the producer and user in closer proximity to each other, can be recognised as a solution for minimising accessibility dependencies (Dellorocas, 1996; Malone et al, 1999). Alternatively, transportation of the resource may be needed in order to reduce accessibility.

6.7.4.3. Pre-requisite

A 'Pre-requisite' constraint type, determines that an item must be produced before it can be used. For example, a process to despatch a car to a sales showroom, cannot be completed until the car has been fitted with an engine. Here, a tactical manager cannot issue the dispatch of a batch of cars, until the engines are fitted. Techniques for managing pre-requisites, can involve keeping an inventory of the resource or making it to order as and when it is required. Such a consumer driven method is synonymous of the 'Just In Time' system of management.

Finally, the constraint types listed above are noted as specialist types of flow dependency within the co-ordination framework. However, it would, owing to their implied meaning, be more accurate to recognise their functionality as constraints upon dependency types in general, rather than specific flow dependencies, as usability, accessibility and pre-requisite constraint types can be occur in both sharing and fit dependency types.

6.8. INTERDEPENDENCIES AND CO-ORDINATION MECHANISMS

Thompson (1967), building upon the "Simon-March-Cyert" approach, lists three co-ordination mechanisms as 'Standardisation', 'Plan' and 'Mutual adjustment', which are used in response to three patterns of interdependency. These are 'Pooled', 'Sequential' and 'Reciprocal'. In a 'Pooled' interdependency, each actor contributes and depends on the whole, and the whole supports individual parts. A 'Sequential' interdependency, is a special form of the pooled interdependency type, in that it signifies one actor being dependent upon another in the system for its functioning.

Figure 6.9, Functional Interdependence, (Thompson, 1967)

Finally, a 'Reciprocal' interdependency, refers to the outputs of one actor, that are the inputs of the other and vice versa. In short, reciprocal interdependence is both pooled and sequential.

In previous sections, the various component parts of co-ordination have been presented. The constructs that have been considered so far, are used within the co-ordination domain to practically highlight and identify occurring interdependencies within a process. Various authors have previously conducted research into the existence of interdependencies (Malone and Crowston, 1994; Tanniru and Jain, 1989; and Thompson, 1967). Furthermore, Kim, Myung & Emdad (2001) when conducting a simulation study regarding decision cycle times upon various co-ordination strategies, identified three types of interdependency. These are 'Common Model', 'Pre-requisite' and 'Concurrent'. Their approach draws parallels with that outlined by Thompson (1967). The only notable differences seem to be in the use of terminology. This is shown in figure 6.10 below.

Thompson (1967)		**Kim, Myung & Emdad (2001)**
Pooled	→	Common Model
Sequential	→	Pre-requisite
Reciprocal	→	Concurrent

Figure 6.10. Comparison of interdependency types

Kim, Myung & Emdad's study presented three types co-ordination mechanism that can be employed to manage these interdependencies, these are standardisation for managing common model interdependencies, synchronisation co-ordination for managing pre-requisite interdependencies, and scheduling co-ordination for managing concurrent interdependencies. Again these co-ordination mechanisms also draw similarities with those presented by Thompson (1967). Those being 'Standardisation', 'Plan' and 'Mutual Adjustment'.

6.9. CONCLUSIONS

In summary, this chapter has outlined co-ordination theory as a method for "managing interdependencies between activities performed to achieve a goal"

(Malone and Crowston, 1994). From this perspective, the underlying purpose of co-ordination theory is to provide a set of constructs that can be used to provide deeper analysis of organisational processes, and their associated sub-activities. Therefore, the key elements of the co-ordination approach can be summarised thus.

- Co-ordination establishes that all processes have lower level activities, which are connected via dependencies.
- Co-ordination details the existence of three dependency types. These are 'Fit', 'Flow' and 'Sharing'.
- Constraint types such as 'Pre-requisite', 'Accessibility' and 'Usability' are also noted as being potential constraints to dependency types.
- Co-ordination uses notions of generalisation and specialisation from *Object Orientation*, in order to provide further analysis as to inheritance relationships between processes.
- Co-ordination presents analytical capabilities for analysing interdependencies.

Co-ordination theory extends OO modelling techniques such as inheritance relationships between composite processes and atomic activities, as well as generalisation and specialisation relationships, by placing the modelling emphasis on processes rather than objects. Specifically, co-ordination theory, as this chapter has outlined, provides a method for analysing organisational processes from differing levels of abstraction, by outlining a process along with its associated activities, resources and actors.

However, the challenge facing this study, from this point forward, is to present an object modelling approach that will embody the co-ordination perspective within an information systems paradigm, in order to provide a system of delivering co-ordinated tactical management information. Such an approach has been outlined in the design and implementation of the Co-ordinated Business Object Model (CBOM). This will be the main emphasis of the next chapter.

SECTION C

Development, Implemention And Testing

C.1. INTRODUCTION

Up to now this thesis has presented in section A, the role of the tactical level manager and the requirements he/she has for information when formulating tactical level objectives. The same section also outlined the use of computer based information systems as a decision supporting tool for tactical management decision making. The examination of both static and dynamic data modeling methods in section B, outlined the various deficiencies within these conventional approaches for modeling organisational processes. Subsequently, the same section also clarified co-ordination theory as a more robust framework for analysing organisational processes. As a modeling approach, co-ordination holds various advantages over other modeling methods for analysing processes, owing to its depth of constructs, as well as its emphasis on serialising processes rather than modeling objects. From this perspective, the co-ordination approach itself partitions business activities into discrete domains which represent communicable elements, that map naturally to an object oriented world view. Therefore, as organisational processes differ in representative levels of abstraction, knowledge leveraged from the co-ordination framework can be used to provide greater analysis of tactical level processes.

The aim of section C is to provide a solution to the problem addressed in section 1.2, whilst providing evidence for its validity. Therefore, section C will be comprised of three chapters, these are considered in more detail below.

C.1.1. Chapter 7 – The Co-ordinated Business Object Model (CBOM)

The Co-ordinated Business Object Model (CBOM) has been developed throughput the course of this study. An investigation of the component parts of the CBOM, will recognise the primary focus of chapter seven. Shown in figure C.1 below, the architecture of the CBOM comprises two layers.

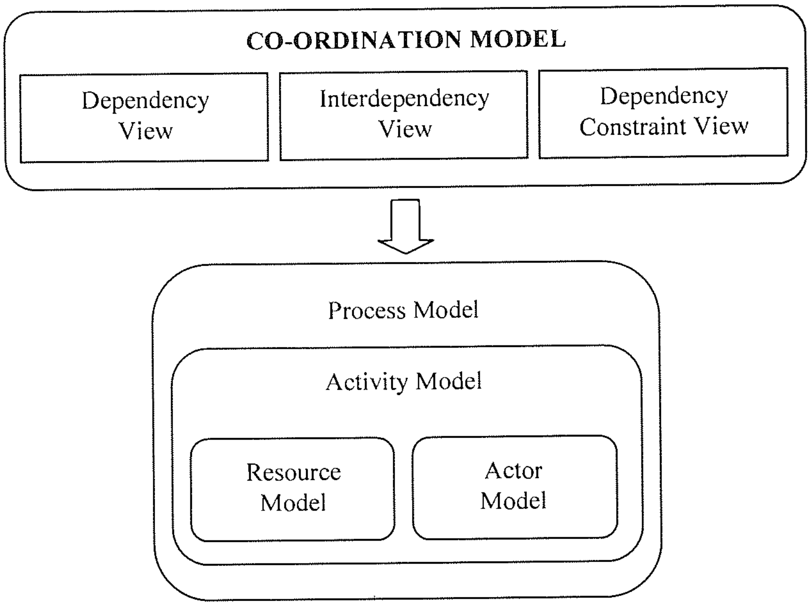

Figure C.1. CBOM Component Objects

The first layer consists of four object models. Namely; the 'Process', 'Activity', 'Resource' and 'Actor' models. Information regarding tactical level processes are captured at this level. The second layer acts as a co-ordination model, which provides co-ordinated information based upon the four object models. This model however, will be presented in more detail throughout chapter seven.

C.1.2. Chapter 8 – The Co-ordinated Business Object Support System(CBOSS)

The Co-ordinated Business Object Support System (CBOSS), has been developed in order to support an implementation of the CBOM model. Specifically, the CBOSS is a demonstrator application that contains the necessary functionality for not only supporting CBOM constructs, but to also provide a test bed with which to test the robustness and efficacy of the CBOM model. Chapter eight will outline the functionality of the CBOSS, whilst specifically targeting the systems ability to provide tactical management information.

C.1.3. Chapter 9 – CBOM Testing And Evaluation

In order to ground the theory development of this study, a set of verification tests will be carried out using the CBOSS. Three processes and their corresponding results will be presented in this chapter.

CHAPTER 7

The Co-ordinated Business Object Model (CBOM)

7.1. INTRODUCTION

The last chapter introduced the role of co-ordination and its associated constructs, as a framework for analysing organisational processes. As the co-ordination framework is principally concerned with managing dependencies between activities, this chapter will suggest that co-ordination can be adapted to help analyse tactical level processes. From this perspective, the challenge facing this study from this point, is to combine the co-ordination constructs presented in chapter six, into a functional business object model that will provide a more robust analysis of tactical level processes. Consequently, the 'Co-ordinated Business Object Model' (CBOM) has been developed throughout the course of this study, with the intention of producing an object model that supports tactical level decision making. Consisting of a two-layer architecture, the CBOM will comprise four object models in the first layer. These are the *Process*, *Activity*, *Resource* and *Actor* objects. Here, data which relates to a tactical level process is captured by these objects. The second layer will comprise the 'Co-ordination Model', this will consist of three 'Views' that will return information based upon the recognised dependencies, interdependencies and dependency constraints within a given tactical level process. A more thorough investigation of the component parts of the CBOM will be presented throughout the course of this chapter.

This chapter will argue that the CBOM can be used to support tactical management decision making. By studying the research literature, the application of co-ordination for supporting tactical management tasks has not been considered before. From this perspective, it will be argued that the CBOM makes a novel contribution towards the field of information systems research.

7.2. STEPS TOWARDS THE FORMULATION OF THE CBOM

Before considering the specific areas of the CBOM, this section will aim to bridge the gap between the research presented in both sections A and B and the proposed object model that will be detailed in section C. The evidence will specifically outline four areas that bring forth the overall criteria for the development of the CBOM. In essence, they will be regarded throughout the remainder of this thesis as a series of requirements that point to what the CBOM should deliver.

- ***To model tactical management information.***
- ***To support the modelling of tactical level processes.***
- ***To provide a model that can be implemented into a fully functional information system.***
- ***To support the tactical management decision making process.***

The above points are expanded in further detail throughout the remainder of this section.

To model tactical management information. Section A of this study was split into three distinct areas that focused around tactical management, the environment in which tactical managers operate and the information needs managers at the tactical level require. Therefore, to be consistent with the research presented in section A, a fundamental requirement of the CBOM must be to support the role of tactical management in their individual environments. From this perspective, chapters three and four outlined the tactical management requirement for information, as well as the various information systems that are available for supporting these management needs. In addition, various attempts already listed in section 1.3, outlined that integrative and collaborative technologies have arguably proved fruitless when supporting specifically the tactical management information need. Moreover, when considering the activities of the tactical manager, figure 4.1 delineates their role in relation to formulating objectives that must both incorporate the wishes of strategic management, along with also being suitable for implementing those objectives throughout the operational level. Therefore, tactical management will have a requirement for up to date information based upon the performance of operational objectives. Such information will also point to the effectiveness of the corresponding tactical level objectives. Chapter nine of this thesis will also present evidence that strengthens this argument in a series of verification tests.

As a fundamental requirement of the CBOM is to model tactical management information, so such requirements should bind seamlessly with the appropriate objectives originally stated in section 1.5. In order to provide more validity to the assertion that a fundamental requirement of the CBOM must be to model tactical management information, it is worth outlining the specific objectives that apply to this area. Firstly, the first objective was; '*To explore the tactical management role in the organisational structure. This will be achieved by; Investigating the hierarchical structure of organisations,* and '*Exploring the tactical management role in the organisational structure'*. Following on from this, the second objective stated, expressed a need; '*To investigate the information needs of tactical management'*. Such an investigation into these three objectives formed the foundations for outlining the nature of tactical level information. In essence, it has outlined the parameters of tactical management information, and hence determined what this first point in the CBOM requirements list must actually achieve.

To support the modelling of tactical level processes. Section C.1.2. outlined the role of the Co-ordinated Business Object Support System (CBOSS). The CBOSS will form the information system front end that the user will interact with. In essence, the CBOSS will be an implementation of the CBOM. The CBOSS however, will be discussed in further detail in chapter eight. A major function of the CBOM, must be to interact with the user. This interaction must support both the inputting and outputting of data. In this case information that is derived from the co-ordination model, must be presented to the user in a way that he or she can understand. Therefore, it is essential that the CBOSS contains the functionality to allow users to be able to support the modelling of tactical level processes. In part this will be done by using the various forms that represent parts of the co-ordination model i.e. *Activities*, *Resources*, *Dependencies* etc as well as using various CBOM constructs outlined in table 6.1. In order for the user to be able to interact with these constructs, it is essential that the CBOSS actually represent them. This will be achieved by building into the eventual CBOSS design a toolset, that will allow the user to perform various tasks, such as add, delete and update activities, resources and dependencies between activities.

The need to address this particular requirement, is consistent with objective six outlined in section 1.5. Here it was stated that there is a need to '*To develop a software tool for demonstrating the modelling framework suggested in objective 5*'. The software tool in question here is notably the CBOSS.

To provide a model that can be implemented into a fully functional information system. By itself the CBOM is useless, unless it can be implemented into a fully functional operational information system. Chapter four outlined the significance of both management information systems (MIS) and decision support systems (DSS) for supporting the decision making capabilities of tactical management. In addition, chapter five outlined the role of both static and dynamic data modelling techniques that are considered useful, w hen designing the information system so it better represents real world entities, as well as the corresponding operations it is based upon. Concluding section B, chapter six outlined the role of co-ordination as a framework for representing organisational processes, specifically within an information systems context. Such is the research presented throughout these earlier chapters, the third point to be outlined in the requirements list for the design and development of the CBOM, outlines the need for the CBOM to be implemented into a fully functional information system. By implementing the CBOM model into such an information system, holds many advantages. For example, users can interact with the CBOM model, information can be stored about processes, from this they can retrieve this data and change it should they wish. In essence, implementing the CBOM into an information system allows users to gain the maximum benefit out of using the model.

The need for creating a system to support the CBOM implementation integrates with some of the objectives outlined in section 1.5. For instance, objective three expresses a need; '*To investigate the role of information systems that are used to support tactical management tasks*'. Such an investigation outlined not only the importance, but also the role of information systems for supporting tactical level tasks. Secondly, objective four outlined a requirement; '*To investigate modelling* *approaches for supporting tactical management decision making'*. This has led to a deep understanding of the design issues relating to information systems. Finally, and perhaps most pertinent to this requirement area was objective six; *'To develop a software tool for demonstrating the modelling framework suggested in objective 5'*. As mentioned earlier in the requirement above and also in section C.1.2, the CBOSS as a software prototype, will support the proposed CBOM implementation. Therefore, the constructs built into the CBOM should be able to translate from the theoretical design outlined in this chapter, to a fully functional software system. The design of the CBOSS will be explored in further detail in chapter eight.

To support the tactical management decision making process.. The final point in this list of requirements, details the role of the CBOM when supporting the tactical management decision making process. Arguably, this is perhaps the most important point as it encompasses the main emphasis of the CBOM. Management decision making in general is supported by a range of information systems. However, as chapter three noted, it is Management Information Systems (MIS) and systems that provide decision support such as Decision Support Systems (DSS) that are most frequently used amongst tactical level managers. Indeed, it is these systems that deliver information to the manager which in turn supports their decision making capabilities. Chapter five provided a detailed description of various static and dynamic data modelling methods, whilst chapter six described the co-ordination framework. Co-ordination is an integral part of the CBOM model. In essence, it uses the notion of co-ordination to manage dependencies between activities. Co-ordination constructs are used throughout the CBOM to provide the functionality of the co-ordination framework. In essence, it is the combination of the research presented specifically in both sections A and B, that provides the foundations for this fourth point in the CBOM requirements list.

The fifth objective outlined in section 1.5 made a requirement; *'To develop a modelling framework for supporting tactical management decision making'*. The need to develop such a framework, provides a template with which to guide the development of the CBOM, so that it supports tactical management tasks. In support of this, objective six outlined a need; ,*'To develop a software tool for demonstrating the modelling framework suggested in objective five'*. The development of the Co-ordinated Business Object Support System (CBOSS), as a tool for supporting tactical management decision making, has been forwarded in this work as an implementation of the CBOM.

Perhaps the best way to test whether or not the CBOM meets the requirements set down at the start of this section is to test it in a real world organisation. The seventh objective listed in section 1.5 details a need to '*Test the modelling framework suggested in objective 5, in order to derive results as to the efficacy of the framework for supporting tactical management decision making*'. In order to fulfil this objective and indeed to also establish whether or not the CBOM supports tactical decision making via the CBOSS, a series tests will be outlined in chapter nine. These tests will revolve around the ability of the CBOM to support the tactical decision making process in two real world organisations. The results of this testing procedure will be outlined in more detail in chapter nine.

To conclude, this section has presented a list of four points, that in short, outline the requirements for the development of the CBOM. In essence, the purpose of this section has been to provide the reader with an understanding as to how this thesis has arrived at the CBOM, from the research presented in sections A and B. As section C concentrates primarily upon the CBOM, CBOSS and the testing of the model under three rigorous scientific tests, it will be useful to revisit the four points listed at the start of this section. This shall be presented in chapter ten. which will be the last chapter in section C. Such a review will prove useful for demonstrating whether or not the requirements have been met.

7.3. A BRIEF GUIDE TO THE CBOM

Up to now it has been asserted that the CBOM is an object model that can be implemented in order to better serve the decision making capabilities of tactical management. However, before considering the more technical areas of the CBOM, it is important, for clarity, to highlight which parts of the CBOM relate to the co-ordination material presented in chapter six and the object oriented material outlined in chapter five. Making such distinctions will allow the reader to see how the

CBOM has emerged from these subject areas. This distinction will be made throughout the remainder of this section.

7.3.1. Co-ordination Constructs

Chapter six provided a broad review of co-ordination constructs. It is useful to briefly summarise these constructs, as they are used in the CBOM model. The CBOM architecture (discussed in section 7.7), outlines a two layer model that is comprised of four object models that interact with a co-ordination layer. These four object models are 'Processes', 'Activities', 'Resources' and 'Actors'. The notion that actors execute processes and activities and that processes own activities which in turn have resources attached to them, are borrowed from the co-ordination domain. Some of the attribute values in the CBOM however do not derive from the literature. In essence some of them are implicitly included such as 'Actor_Name' or 'Actor_Department', this is because it would not be likely that an actor, such as a tactical level employee, would not have a name or indeed belong to a department. Therefore, in order for the CBOM to be coherent it has been necessary to add certain attribute values that underpin the needs of the model for accurately representing tactical management processes.

To be consistent with the research presented so far, the CBOM needs to possess the relevant constructs necessary to support tactical decision making. In essence, these can be regarded as the needs of the CBOM for performing its operations. These constructs from a co-ordination perspective, are represented as a series of attributes. All these attribute values are outlined in tables 7.1 to 7.4 below. The purpose of these tables is to make clear to the reader the origin of the attribute values, and indeed why the author choose to include these attributes in not only the four object models, but also the co-ordination model. The tables below are composed of two columns, these are 'Attribute Name' and 'Derived', signifying if the attribute originated from either the literature pertaining to co-ordination or was simply implicitly added by the author. Values that were added by the author, relate to areas that were required for testing the CBOM in a real world environment. For example, it is crucial to know the process name, or how many activities are designated to a particular process. Moreover, some attribute values such as 'AR_Usage_Status' and 'Resource_Coupling' have been added by the author as a result of the need to incorporate certain functionality into a piece of software. The tables below outline the attribute names of the four object models as well as the co-ordination model. Along with this, a description of where the attributes are derived from, as well as a short outline of the attributes purpose are provided below.

Attribute Name	Derived	Purpose
Activity_ID	Added by author	Represents the ID of the activity
Activity_Name	Added by author	Represents the name of the activity
Time_Allocated	Added by author	Represents the amount of time allocated to the activity
Execution_Date	Added by author	Represents the date the activity will execute
Actual_Time_Taken	Added by author	Represents the length of time the activity has taken
Time_Differential	Added by author	Represented by: Time_Allocated - Actual_Time_Taken
Management	Co-ordination	Represents who is manages the activity
Activity_Sequence	Co-ordination	Represents the position of the activity in relation to other activities in the process.
Sub_Activity	Co-ordination	Represents id the activity is a sub-activity of another activity
Pooled_Qualifier	Added by author	Represents the pooled interdependency status of the activity (see 6.8 for a more in depth discussion of interdependencies)
Outside_Activity_Dependent	Added by author	Represents where another activity in the same process is dependent upon the current activity.
Dep_On_Resource	Added by author	Represents where the current activity is dependent upon the output of another activity in the same process.
Dependency_Link	Co-ordination	Represents the dependencyID associated to the current activity
Resource_Coupling	Added by author	Represents which resources are connected to the current activity
Dependency_Link_Name	Added by author	Represents the dependency name associated to the current activity
AR_Usage_Status	Added by author	Represents if the activity uses or produces a resource

Table 7.1. Activity Attributes

Attribute Name	Derived	Purpose
Process_ID	Added by author	Represents the ID of the process
Process_Name	Added by author	Represents the name of the process
Activity_Ownership	Co-ordination	Represents which activities belong to the current process
Process_Start_Date	Added by author	Represents the start of the process

Table 7.2. Process Attributes

Attribute Name	Derived	Purpose
Resource_ID	Added by author	Represents the ID of the resource
Resource_Name	Added by author	Represents the name of the resource

Activity_Designation	Co-ordination	Represents the which activity this resource belongs to
Shareability	Co-ordination	Represents if the resource is shareable between two or more activities
Divisible	Co-ordination	Represents if the resource is divisable between two or more activities
Consumability	Co-ordination	Represents if the resource is consumable
Concurrency	Co-ordination	Represents if the resource can be used by two or more activities at the same time
Usability	Co-ordination	Represents if the resource is usable
Availability_Date	Added by author	Represents when the resource will be available for use
Tangibility	Co-ordination	Represents if the resource is tangible
Position	Added by author	Represents where the resource is located
Time_Span	Added by author	Represents how ephemeral the resource is.
Resource_Accessible	Co-ordination	Represents the accessibility of the resource
Resource_Usable	Co-ordination	Represents the usability of the resource

Table 7.3. Resource Attributes

Attribute Name	**Derived**	**Purpose**
Actor_ID	Added by author	Represents the activity ID
Actor_Name	Added by author	Represents the name of the activity
Actor_Designation	Added by author	Represents who is designated to execute the activity.
Department	Added by author	Represents the department of the actor
Organisation	Added by author	Represents the organisation of the actor
Location	Added by author	Represents the location of the actor
Contact	Added by author	Represents the contact details of the actor

Table 7.4. Actor Attributes

The co-ordination model is primarily a combination of the attribute values presented in the four object models (*Processes, Activities, Resources and Actors*). Therefore, all three views that are listed in the co-ordination model, 'Dependency', 'Interdependency' and 'Constraint', use a combination of the attributes listed above. However, as will be explained in further detail throughout this chapter, they are used to return different information, which is relevant to the operational issues of tactical level processes. A review of the attributes listed in the 'Dependency', 'Interdependency' and 'Constraint' models are outlined in tables 7.5, 7.6 and 7.7 below.

Attribute Name	Derived	Purpose
Dependency_View_ID	Added by author	To distinguish between different dependencies in the current process.
Dependency_View_Name	Added by author	To provide a dependency with a name.
Process_Designation	Added by author	Sets the Process_Designation attribute to the value stored in the Process.Process_Name
Resource	Added by author	Sets the Resource attribute to the value stored in the Resource.Resource_Name.
Consumer_Activity	Added by author	Sets the Consumer_Activity attribute to the value stored in the Activity.Activity_ID
Producer_Activity	Added by author	Sets the Producer_Activity attribute to the value stored in the Activity.Activity_ID
Dependency_Type	Added by author	Sets the Dependency_Type attribute to either 'FIT','FLOW' or 'SHARING'.

Table 7.5. Dependency Attributes

Attribute Name	Derived	Purpose
Interdependency_View_ID	Added by author	To distinguish between different interdependencies in the current process.
Interdependency_View_Name	Added by author	To provide a interdependency with a name.
Process_Designation	Added by author	Sets the Process_Designation attribute to the value stored in the Process.Process_Name.
Activity	Added by author	Sets the Activity attribute to the value stored in the Activity.Activity_Name
Resource_Dependent	Added by author	Sets the Resource_Dependent attribute to the value stored in the Activity.Activity_Name
Activity_Dependent	Added by author	Sets the Activity_Dependent attribute to the value stored in the Activity.Activity_Name
Sequential_Status	Added by author	Sets the Sequential_Status attribute to either 'SEQUENTIAL' or 'NON-SEQUENTIAL'
Reciprocal_Status	Added by author	Sets the Reciprocal_Status attribute to either 'SEQUENTIAL' or 'NON-SEQUENTIAL'.
Pooled_Status	Added by author	Sets the Pooled_Status attribute to either 'POOLED' or 'NON-POOLED'.

Table 7.6. Interdependency Attributes

Attribute Name	Derived	Purpose
Process_Designation	Added by author	Sets the Process_Designation attribute to the value stored in the Process.Process_Name.
Producer_Activity	Added by author	Sets the Producer_Activity attribute to the value stored in the Activity.Activity_Name where the Activity.AR_Usage_Status = 'PRODUCES'.
Resource_Name	Added by author	Sets the Resource_Name attribute to the value stored in the Resource.Resource_Name.
Producer_Date	Added by author	Sets the Producer_Date attribute to the value stored in the Activity.Execution_Date
Resource_Created	Added by author	Sets the Resource_Created attribute to the value stored in the Resource.Availability_Date.
User_Activity	Added by author.	Sets the User_Activity attribute to the value stored in the Activity.Activity_Name, where the Activity.AR_Usage_Status = 'USES'.
User_Date	Added by author	Sets the User_Date attribute to the value stored in the Process.Process_Start_Date.
Resource_Accessible	Added by author	Sets the Pooled_Status attribute to either 'YES' or 'NO'.

Resource_Usable	Added by author	Sets the Pooled_Status attribute to either 'YES' or 'NO'.
Constraint_Type	Added by author	Sets the Pooled_Status attribute to either 'ACCESSIBILITY'., 'USABILITY' or 'PREREQUISITE'.

Table 7.7. Constraint Attributes

The tables above are intended to provide the reader with a brief outline of the constituent attribute values in the CBOM. A more thorough analysis will be provided through the remainder of this chapter. The final part of this section will now outline the contribution of object oriented constructs.

7.3.2. Object Oriented Constructs

The architecture of the CBOM is split into four objects, 'Processes', 'Activities', 'Resources' and 'Actors'. By shaping the architecture of the model in such a way allows the distinct separation of the four object areas. From a software development perspective, this allows the objects to be divided into easily understandable modules. Within these models, object oriented constructs have also been adopted. Finally, it can be regarded that if co-ordination constructs are responsible for the modelling of the four object models, then the object oriented principles outlined within this chapter are responsible for representing the attributes. These however, shall be considered in further detail in the next section.

7.4. IMPORTANT ELEMENTS IN CBOM MODELLING

The CBOM, comprises a modelling approach that has been developed throughout the course of this study. The principle aim of the CBOM is to present appropriate tactical level information based upon tactical level processes. Drawing upon knowledge from both *object orientation* and *co-ordination theory*, the CBOM represents an object model that is richer than other models for representing tactical level information. Key constructs in the specification of the CBOM are introduced together with the notation that will be used throughout the rest of this chapter.

7.4.1. Instantiation

Classes are instantiated in order to create objects. CBOM supports instantiation by using a double colon '::'. In this context an object instance such as the one below, suggests that the process 'Assess_Corporate_Finances' is an instance of the class 'Process';

```
Process::Assess_Corporate_Finances
```

7.4.2. Attributes

Attributes detail information about objects at the atomic level that cannot be subdivided into meaningful component pieces. Attributes have both a description and a value. CBOM expresses attributes in three ways depending upon whether they are lists, single values or an enumeration. For example, lists detail an attribute that has multiple values. These are expressed in brackets ().

```
Produce_Sales_Budget.Outside_Activity_Dependent =
                                    (Produce_Production_Budget,
                                     Produce_Finance_Budget);
```

Strings "" are used to express single values. When single values occur, brackets are omitted.

```
Produce_Sales_Budget.Outside_Activity_Dependent =
                                    "Produce_Production_Budget";
```

Strings, enclose individual attribute values in the example above. In some circumstances however, attributes have enumerated values. For example, in the 'Activity' object, the attribute 'Pooled_Qualifier' can only have one of two values. Expressed in upper case, these are 'CONTRIBUTE' or 'NON-CONTRIBUTE'. CBOM details a description of these in brackets. For instance;

```
#Activity.Pooled_Qualifier = (CONTRIBUTE | NON-CONTRIBUTE);
```

When expressing values however, the brackets are removed. For example;

```
Produce_Sales_Budget.Pooled_Qualifier = CONTRIBUTE;
```

Finally, when referring to object names, strings shall be omitted. For example;

```
Assess_Corporate_Finances.Process_Name = Assess_Corporate_Finances;
```

In this case, reference is made to the 'Assess_Corporate_Finances' example process, which will be outlined in further detail in section 7.6.

7.4.3. Derived Attributes

Derived attributes are formulated as a result of a calculation. The formulas for these are explicitly outlined throughout this study where applicable. For example, in the 'Activity' model (see figure 7.4), the attribute 'Time_Differential' is populated with;

```
Time_Differential = Time_Allocated - Actual_Time_Taken;
```

As 'Time_Allocated' and 'Actual_Time_Taken' are also part of the activity object, the object designation does not precede these attributes. However, where the attributes are derived from other objects, the object is explicitly identified. For example;

```
Actual_Time_Taken = Process.Process_Start_Date + SysDate;
```

In this instance the 'Process_Start_Date' is required from the 'Process' object. The resultant effect of adding the 'SysDate' (The current systems date) determines the value of the 'Actual_Time_Taken'.

7.4.4. References To Object Instances

CBOM details a difference between a single instance of an object, and a process that refers to the process object. These same ramifications also apply to the 'Activity', 'Resource' and 'Actor' models. In order to differentiate between these two states, the '#' sign will be used in order to refer to 'A Process', 'An Activity', 'A Resource' and 'An Actor' in the attribute description.

```
#Process.Process_ID;
```

The above refers to an instance of a 'Process_ID' that belongs to a particular process. Example values highlighting this may take the form;

```
Assess_Corporate_Finances.ProcessID = "Pro001";
```

Here the '#' is omitted as the process is explicitly outlined. The '#' throughout the remainder of this chapter shall only be used when expressing object descriptions. Finally, where references to the object model are made, the '#' is omitted. The example below outlines a general reference to the 'Process_ID' attribute in the process object.

```
Process.Process_ID;
```

By outlining just the object, would detail that a particular value is required from that object, for instance;

```
Resource_Name:= #Resource;
```

7.4.5. Object Paths

The CBOM uses the dot '.' notation to demonstrate a path towards an attribute value. The example below denotes an object path leading to the 'Activity_ID'.

```
#Process.Activity.Activity_ID;
```

7.4.6. Pointers To Objects

Various attributes in the four object models that makeup the CBOM are populated with the values from other objects. In order to outline these values, a pointer will be used. The notation for the pointer will take the form of a colon succeeded by an equals sign ':=' an example of this is given below.

```
#Resource.Activity_Designation := #Activity.Activity_ID;
```

The above suggests that the 'Activity_Designation' value is populated with the 'Activity.Activity_ID' from the 'Activity' object. However, when the value is expressed, the ':=' is omitted, in order to provide the more familiar form;

```
Sales_Data.Activity_Designation = Produce_Sales_Budget.Act001;
```

7.4.7. PartOf Relationships

PartOf relationships are denoted by the use of curly '{}' brackets. PartOf relationships detail elements that are part of the object. These relationships in

CBOM are used to detail attributes that are part of objects. They are also used to highlight activities that are part of tactical level processes.

```
Process1{Activity1, Activity2, Activity3};
```

Where values relating to the 'Assess_Corporate_Finances' process would detail the following;

```
Assess_Corporate_Finances {Produce_Sales_Budget,
                           Produce_Production_Budget,
                           Produce_Finance_Budget,
                           Produce_Corporate_Budget};
```

Here, all four activities 'Produce_Sales_Budget, Produce_Production_Budget', 'Produce_Finance_Budget' and 'Produce_Corporate_Budget' are part of the 'Assess_Corporate_Finances' process. However, in the case that an activity has sub-activities, then the notation ':{' without the closing bracket '}' will be used. For example;

```
Process1:{Activity1 :{Activity1.1;
```

suggests that 'Activity1' is a part of 'Process1', and 'Activity1.1' is a part of 'Activity1'.

7.4.8. Generalisation and Specialisation

Generalisation and specialisation constructs in object oriented programming (Stefik & Bobrow, 1986; Wegner, 1987; Brachman & Levesque, 1985), are abstraction principles that provide hierarchical structuring of the semantics of a data model. Such principles can also be applied to processes (Wyner, Jintae, 2003). The CBOM uses the square brackets '[ ]', to recognise generalisation and specialisation relationships. For example;

```
Activity[Sub_Activity1, Sub_Activity2];
```

With possible values detailing;

```
Produce_Sales_Budget[Produce_British_Version, Produce_French_Version];
```

Where 'Produce_British_Version' and 'Produce_French_Version', are specicalisations of 'Produce_Sales_Budget'. To denote further specialisations of 'Activiy1' the ':[' notation is used. For instance,

```
Activity:[Activity1:[Activity1.1;
```

Where example values relating to the 'Produce_Sales_Budget' example would detail;

```
Produce_Sales_Budget:[Produce_British_Version :[Produce_English_Version;
```

Here 'Produce_English_Version' is a specialisation of the activity 'Produce_British_Version'. Closing brackets ']' in this context are omitted.

7.4.9. Separators

The comma ',' is used in CBOM to separate multiple values in an attribute. The semicolon ';' is used to separate attribute values. As a consequence, the semicolon is used as a terminator and hence placed at the end of the attribute description. This is shown below.

```
#Activity.Dependency_Link*(
          Resource_Coupling,
          Dependency_Link_Name,
          AR_Usage_Status = (USES|PRODUCES));
```

Values relating to these attributes would also be separated in the same way. For example.

```
Dep001("Sales_Data","SalesB_To_SalesD",USES);
Dep002("Sales_Budget","SalesB_To_USalesB",PRODUCES);
```

Finally the pipe '|' is used to separate 'or' values. For example;

```
#Activity.Pooled_Qualifier = (CONTRIBUTE | NON-CONTRIBUTE);
```

The above details that the attribute 'Pooled_Qualifier' can only have a value equal to 'CONTRIBUTE' or 'NON-CONTRIBUTE'. However, commas are used to separate multiple values, as is the case for expressing a dependency type. In this example, 'FIT and FLOW' are separated.

```
ACF_Dependency_View_1.Dependency_Attributes =
                                    ("Prod_Sales_Data_Summary",
                                     "Produce_Corporate_Budget",
                                      Produce_Production_Budget,
                                      Produce_Finance_Budget,
                                     (FIT,FLOW));
```

The example above denotes that some values are in strings. This is owing to the 'Dependency_Attributes' field containing four attribute values (see figure 7.10).

7.4.10. Iterations

Where object attributes can hold more than one value, the star '*', shall be used to explicitly state this in the object model diagram. For example, in the 'Activity' object in figure 7.4, the attribute 'Outside_Activity_Dependent', is succeeded with a '*'. For example , 'Outside_Activity_Dependent*'. Inferring that multiple values can be assigned to this attribute. For example;

```
Produce_Sales_Budget.Outside_Activity_Dependent =
                                    (Produce_Production_Budget,
                                     Produce_Finance_Budget);
```

These iterations shall also be explicitly stated when detailing the attribute description. For instance;

```
#Activity.Outside_Activity_Dependent*;
```

7.5. IMPORTANT ELEMENTS IN CBOM DIAGRAMMING

Presented in this chapter are three diagrams that have been developed in order to better represent CBOM constructs. This diagramming approach has been developed throughout the course of this study, in order simplify the representation of tactical level processes and associated co-ordination constructs. Therefore, the focus of this section is to bring to the attention of the reader, important elements in this diagramming approach.

Firstly, as shown in figure 7.1 (Process Model Level1: Assess Corporate Finances), a description of the activities and resources that make up the process shall be provided. Secondly, the modelling notation developed in table 6.1 shall be used to distinguish the various co-ordination elements as shown in figure 7.6 (Process Model Level2: Assess Corporate Finances). Finally, the 'Co-ordination View' diagram such as the one presented in figure 7.7 (Co-ordination View: Assess Corporate Finances), is consistent with Zlotkin (1995) in so much as it details the use of rectangles for representing activities and ovals for representing resources. For brevity, ovals are not used in level one and level two diagrams as resources are explicitly represented with an identifier and corresponding name, for example, '$R(R_3)$: Production_Data', which details that the resource 'Production_Data' is denoted with an 'R', which represents a resource, and an identification equal to 'R3'.

7.5.1. Nodes

Figure 7.6 provides an example of a 'Process Model Level 2'. An 'S' is attached to some resources in order to signify that they are shared. Although the identification of shared resources is implicit from the diagram, the 'S' node allows the user to quickly identify resources that are used by two or more activities. Secondly, an 'FI' is used to indicate a fit dependency. Finally, 'FL' is used represent flow dependencies.

7.6. THE ASSESS CORPORATE FINANCES EXAMPLE PROCESS.

Before considering the component parts of the CBOM, it is necessary to point out that for clarity, all areas of the CBOM discussed in this chapter will refer to the 'Assess Corporate Finances' budget example. This is shown in figure 7.1. The goal of the 'Assess Corporate Finances' process, is to produce a corporate budget. This entails the execution of four activities. Namely, 'Produce Sales Budget', 'Produce Production Budget', 'Produce Finance Budget', and 'Produce Corporate Budget'. The remainder of this chapter shall use the process outlined in figure 7.1 below as a vehicle for demonstrating the CBOM modelling constructs.

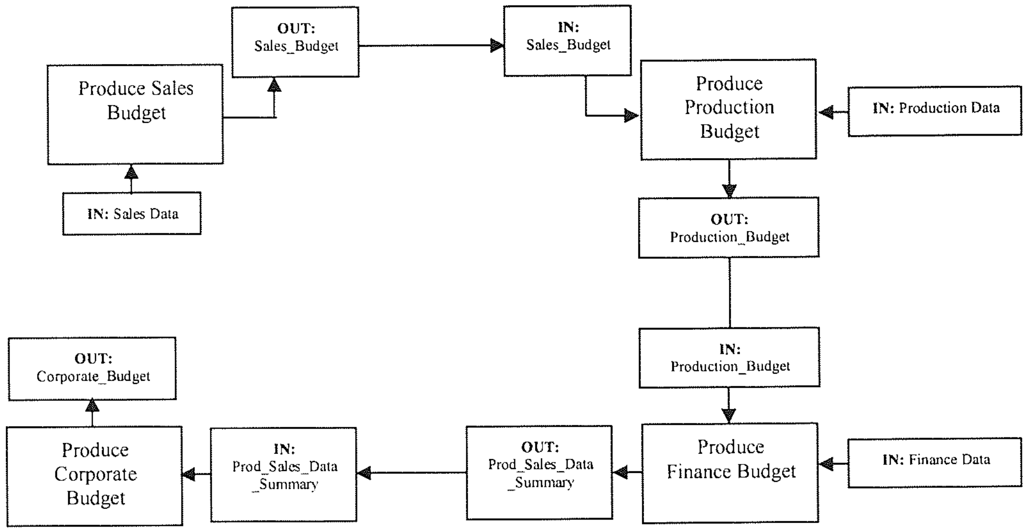

Figure 7.1. Process Model Level 1: Assess Corporate Finances

7.7. THE CBOM ARCHITECTURE

The purpose behind the CBOM, is to provide an object model that incorporates co-ordination constructs. From an operational perspective, the CBOM will incorporate the functionality to support tactical level decision making. The CBOM comprises a two-tier architecture. This section will outline in greater detail, the component parts of the CBOM.

7.7.1. The CBOM Environment

The CBOM comprises two layers. The first layer consists of four object models. These are the 'Process Model', 'Activity Model', 'Resource Model' and 'Actor Model'. This is shown in figure 7.2 below. All four object models interact with a co-ordination model, which in turn provides the necessary functionality for analysing tactical level processes. As a second layer to the CBOM, the co-ordination model is made up of three co-ordination views. These are:

- Dependency View
- Interdependency View
- Constraint View

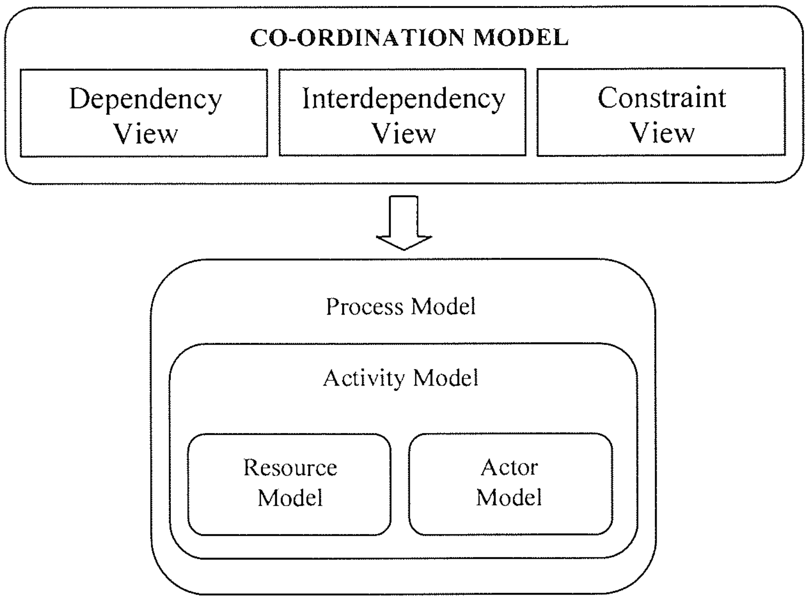

Figure 7.2. CBOM component objects

The co-ordination model shall be considered in more detail in section 7.7.6 below.

7.7.2. The Process Model

The process model is used to represent tactical level processes. In short, there are four attributes that comprise the process model, these are shown in figure 7.3.

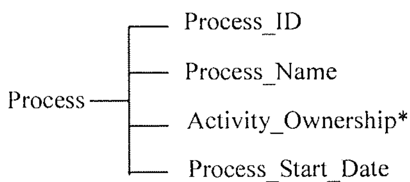

Figure 7.3. The Process Model In CBOM

These four attributes that comprise the process model will be considered in more detail below;

Process_ID. The 'Process_ID' represents the unique identifier for all processes. A description of the 'Process_ID', with a representative value relating to the 'Assess Corporate Finances' budget example, would take the form;

```
#Process.Process_ID;
Assess_Corporate_Finances.ProcessID = "Pro001";
```

Process_Name. The 'Process_Name' complements the 'Process_ID'. A description of this attribute can be summarised as;

```
#Process.Process_Name;
```

and with a corresponding value;

```
Assess_Corporate_Finances.Process_Name = Assess_Corporate_Finances;
```

Activity_Ownership. The CBOM models activities belonging to composite processes (Camarinha-Matos, 2001, Malone and Crowston, 1994). Therefore one or more activities are modelled as being 'PartOf' a single process.

```
#Process1.Activity_Ownership*;
```

The above determines that multiple activities are part of 'Process1'. Grouping instances of activities that belong to the same process would take the form;

```
Assess_Corporate_Finances{Produce_Sales_Budget,
                          Produce_Production_Budget,
                          Produce_Finance_Budget,
                          Produce_Corporate_Budget};
```

thus detailing that four activities belong to the 'Assess_Corporate_Finances' process.

Process_Start_Date. The 'Process_Start_Date' represents the date when the process starts. The description of this attribute would take the form;

```
#Process.Process_Start_Date;
```

and with a representative value;

```
Assess_Sales_Budget.Process_Start_Date = "10OCT03";
```

7.7.3. The Activity Model

The principle dimensions of the activity model are presented in figure 7.4. Its corresponding attribute values are considered in further detail below.

- Activity
  - Activity_ID
  - Activity_Name
  - Time_Allocated
  - Execution_Date
  - Actual_Time_Taken := #Process.Process_Start_Date + SysDate
  - Time_Differential = Time_Allocated – Actual_Time_Taken
  - Management := #Actor.Actor_ID
  - Activity_Sequence
  - Sub_Activity*
  - Pooled_Qualifier = (CONTRIBUTE | NON-CONTRIBUTE)
  - Outside_Activity_Dependent*
  - Dep_On_Resource*
  - Dependency_Link*
    - Resource_Coupling := #Resource.Resource_Name
    - Dependency_Link_Name
    - AR_Usage_Status = (USES | PRODUCES)

Figure 7.4. Activity model in CBOM

Activity_ID. This attribute is a unique identifier which holds the values of the activity identification. For instance;

```
#Activity.Activity_ID;
Produce_Sales_Budget.Activity_ID = "Act001";
```

Activity_Name. The 'Activity Name' is used to distinguish between activities in a process. For example, an activity that produces the corporate budget, may be specified as;

```
#Activity.Activity_Name;
```

Where a corresponding value would take the form;

```
Produce_Sales_Budget.Activity_Name = Produce_Sales_Budget;
```

Time_Allocated. Section 6.4 detailed activity assignments along with relevant issues regarding timespan. Dittrich (1991), suggested a correlation between the time allocated for an activity and the management level that would typically carry out that activity. In drawing parallels with Dittrich, CBOM captures time details that are applicable to the activity using four attribute values. The first is the 'Time Allocated'. This attribute details the length of time allocated for an activity to execute. Its description and value are shown below.

```
#Activity.Time_Allocated;
Produce_Production_Budget.Time_Allocated = "1_Day";
```

Execution_Date. The second attribute details the 'Execution_Date'. This attribute highlights the date the activity will be executed. Its description and value are detailed as;

```
#Activity.Execution_Date;
Produce_Production_Budget.Execution_Date = "21OCT03";
```

Actual_Time_Taken. This attribute outlines the actual length of time an activity has taken to complete. It is calculated using the 'Process_Start_Date' attribute from the 'Process' object, and the current system date (SysDate). The ':=' symbol is used to denote that the 'Actual_Time_Taken' attribute requires a specific value from another object. In this case the 'Process_Start_Date'. A description of this attribute is given below.

```
#Activity.Actual_Time_Taken :=  #Process.Process_Start_Date + SysDate;
```

Here a corresponding value would realise;

```
Produce_Sales_Budget.Actual_Time_Taken = "1_Day";
```

Time_Differential. Finally, this attribute is a derived attribute that is calculated using the attributes 'Time_Allocated' – 'Actual_Time_Taken'. A description of this attribute would take the form;

```
#Activity.Time_Differential = Time_Aloocated - Actual_Time_Taken;
```

With a corresponding value referring to the 'Produce_Sales_Budget' activity;

```
Produce_Sales_Budget.Time_Differential = "0_Days";
```

Management. The 'Management' attribute links the actor to the current activity. A description of this attribute outlines the 'Actor_ID';

```
#Activity.Management := #Actor.Actor_ID;
```

An example value relating to the 'Produce_Sales_Budget' activity would take the form;

```
Produce_Sales_Budget.Management = "01SA";
```

Activity_Sequence. The 'Activity_Sequence' attribute simply outlines the position of the current activity in its associated process. The 'Produce_Sales_Budget' activity in the example below, is first to execute.

```
#Activity.Activity_Sequence;
Produce_Sales_Budget.Management = "1";
```

Sub_Activity. This attribute outlines activities that can be considered to belong to other activities in the same process. A description of this attribute is shown below.

```
#Activity.Sub_Activity*;
```

The 'Assess Corporate Finances' example process does not contain an example of a sub-activity. However, for the purpose of providing an example with which to illustrate this attribute, the 'Produce_Sales_Budget' activity shown below, has a value that is equal to 'Produce_UK_Version'. In this case, 'Produce_UK_Version' is a sub activity of 'Produce_Sales_Budget'.

```
Produce_Sales_Budget.Sub_Activity = "Produce_UK_Version";
```

Pooled_Qualifier. Section 6.8. detailed three interdependency types as described by Thompson (1967). Namely, *Pooled*, *Sequential* and *Reciprocal*. A pooled interdependency occurs when a number of activities contribute towards the overall goal of the process. CBOM models pooled interdependencies by establishing if the activity contributes towards the overall goal of the process. In this context, the 'Pooled_Qualifier' attribute is used. Two values are assigned to this attribute. These are 'CONTRIBUTE' which refers to an activity that contributes towards the overall process, and 'NON-CONTRIBUTE' which determines if the activity does not contribute towards the overall goal of the process. An example of a contributing and non-contributing activity is shown in figure 7.5 below.

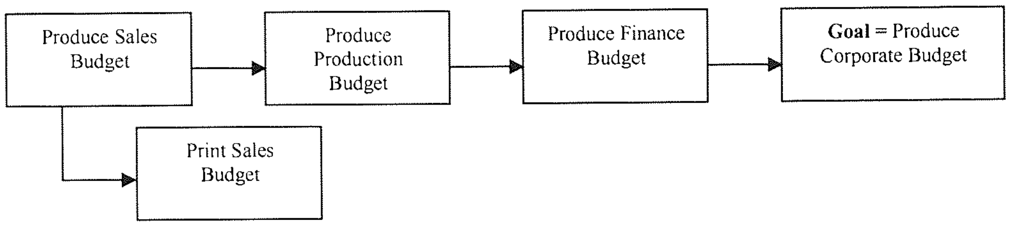

Figure 7.5. 'Print Sales Budget' as a 'Non-Contributing' activity

The example above augments the process presented in figure 7.1 by adding the activity 'Print Sales Budget'. In this context, 'Print Sales Budget' can be regarded as not contributing towards the production of the corporate budget, as without it, the overall goal of the process can still be achieved. A description of the 'Pooled_Qualifer' attribute would take the form;

```
#Activity.Pooled_Qualifier = (CONTRIBUTE | NON-CONTRIBUTE);
```

Where a corresponding value relating to the 'Produce_Sales_Budget', would take the form;

```
Produce_Sales_Budget.Pooled_Qualifier = CONTRIBUTE;
```

Moreover, the attribute value relating to the 'Print_Sales_Budget' would recognnise;

```
Produce_Sales_Budget.Pooled_Qualifier = NON-CONTRIBUTE;
```

Outside_Activity_Dependent and Dep_On_Resource. A reciprocal interdependency occurs when an activity not only relies upon the output of another activity, but also has activities that rely upon its output. The activity model specifies reciprocal interdependencies by assigning values to two attributes. Namely, 'Outside_Activity_Dependent' (where another activity in the same process is dependent upon the current activity), and 'Dep_On_Resource' (where the current activity is dependent upon the output of another activity in the same process). As both these attributes contribute towards the occurrence of a reciprocal interdependency, they shall be considered together. Examples of these two attributes are presented in figure 7.6 below. These are further clarified in table 7.8.

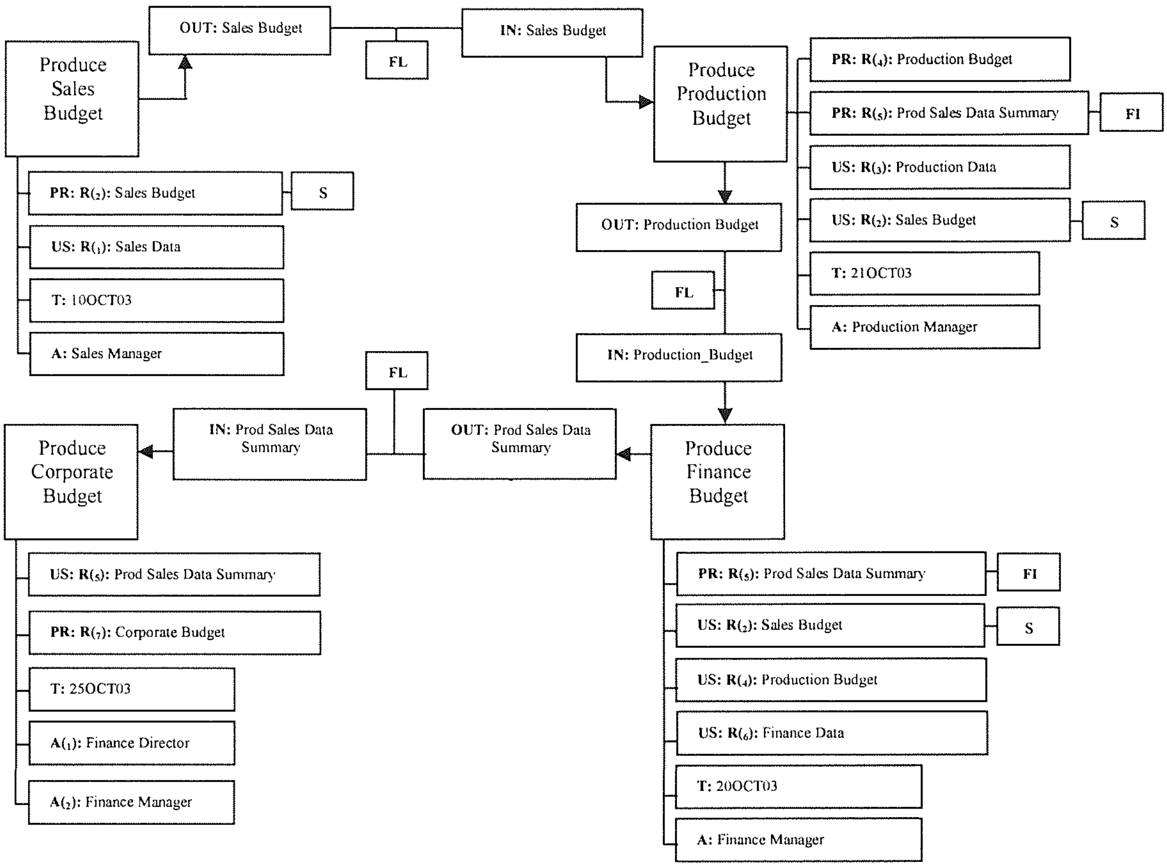

Figure 7.6. Process Model Level 2: Assess Corporate Finances

For ease of understanding, table 7.8 below presents all the 'Outside_Activity_Dependent' and 'Dep_On_Resource' occurrences outlined in figure 7.6 above.

Activity	Outside_Activity_Dependent	Dep_On_Resource
Produce Sales Budget	Produce Production Budget' Produce Finance Budget	None
Produce Production Budget	Produce Finance Budget	Produce Sales Budget' Produce Finance Budget
Produce Finance Budget	Produce Corporate Budget, Produce Production Budget	Produce Sales Budget' Produce Production Budget
Produce Corporate Budget	None	Produce Sales Budget' Produce Production Budget' Produce Finance Budget

Table 7.8. Resource dependency occurrences

Taking the activity 'Produce Production Budget' as an example, table 7.8 shows that this activity is dependent upon both the 'Produce_Sales_Budget' and the 'Produce_Finance_Budget'. This is expressed with a representative example;

```
#Activity.Dep_On_Resource*;
```

Where values relating to the 'Produce_Sales_Budget' realise;

```
Produce_Sales_Budget.Dep_On_Resource =
                                        (Produce_Sales_Budget,
                                         Produce_Finance_Budget);
```

This details the existence of a sequential interdependency. However, the 'Produce_Finance_Budget' is also dependent upon the output of the activity 'Produce_Production_Budget'. A description of this attribute would take the form;

```
#Activity.Outside_Activity_Dependent*;
```

Where the value relating to the 'Produce_Production_Budget' would realise;

```
Produce_Production_Budget.Outside_Activity_Dependent =
                                                     Produce_Finance_Budget;
```

Thereby denoting the existence of a reciprocal dependency, as the 'Produce_Finance_Budget' and the 'Produce_Production_Budget' are dependent upon each other. Although the data required to map activity dependencies are captured in these two attributes, the interdependency occurrences are presented to the tactical manager via the 'Interdependency View'. This will be discussed in further detail in section 7.7.6.2.

Dependency_Link. The dependency link consists of the attributes:

- Resource_Coupling
- Dependency_Link_Name
- AR_Usage_Status

The 'Dependency_Link' attribute represents the dependency link that occurs between an activity and a resource. Figure 7.7 details a 'Co-ordination View' of the

'Assess Corporate Finances' process example. Specifically, this diagram displays all eleven dependency links and their corresponding dependency link names.

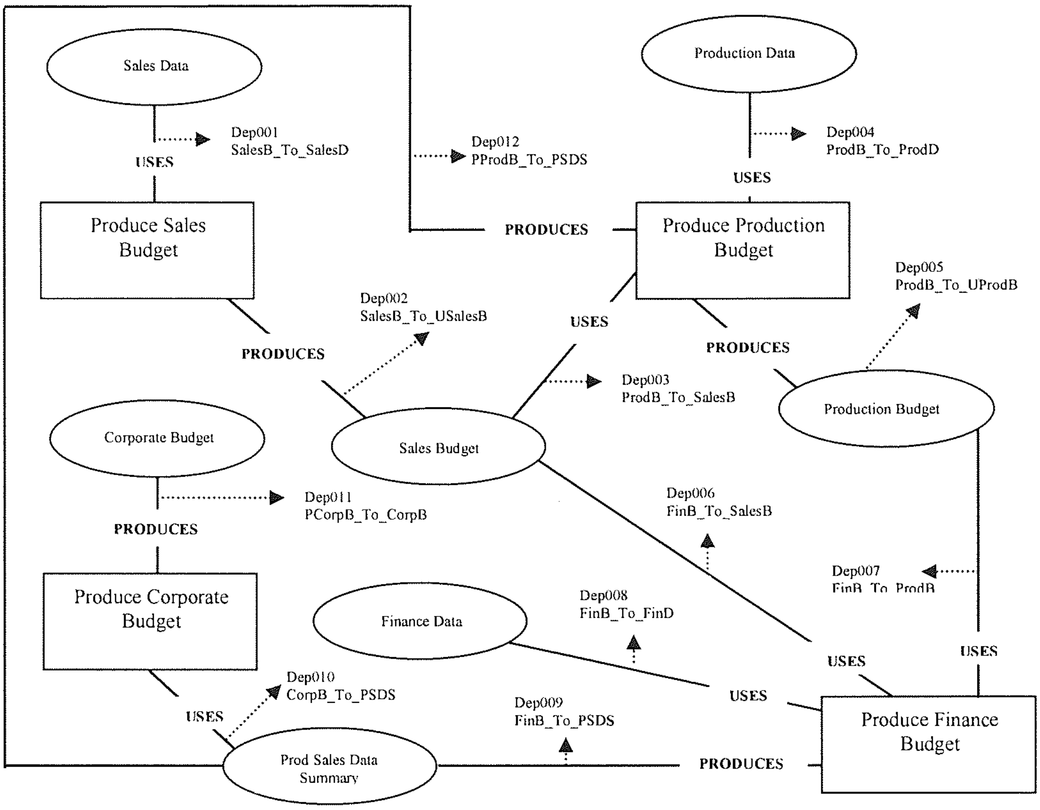

Figure 7.7. Co-ordination View: Assess Corporate Finances

Consistent wth Zlotkin (1995), activities are represented as rectangles and resources as ovals. A description of the 'Dependency_Link' attribute takes the form:

```
#Activity.Dependency_Link*(
          Resource_Coupling := #Resource.Resource_Name,
          Dependency_Link_Name,
          AR_Usage_Status = (USES|PRODUCES));
```

The 'Resource_Coupling', 'Dependency_Link_Name' and the 'AR_Usage_Status', are linked to one instance of the 'Dependency_Link'. For example;

```
Produce_Sales_Budget.Dep001 =
    (Sales_Data, "SalesB_To_SalesD", USES);

Produce_Sales_Budget.Dep002 =
    (Sales_Budget, "SalesB_To_USalesB", PRODUCES);
```

Here both the resources 'Sales_Data' and 'Sales_Budget' are linked to the 'Produce_Sales_Budget' activity

The 'Dependency_Link' attribute, simply outlines all occurances of dependency links between resources and activities. The 'Dependency_Link_Name', compliments all dependency links with a meaningful name. For instance, in the example above, the 'Dependency_link' equal to 'Dep002' has a 'Dependency_Link_Name' equal to 'SalesB_To_UsalesB'. Also modelled in the 'Dependency_Link' is the 'Resource_Coupling' attribute. This denotes the resource to which the current activity is connected to via the dependency link. In this case, the 'Sales_Budget'.

The 'AR_Usage_Status' stands for 'Activity Resource Usage Status'. Its purpose is to detail the 'Uses' or 'Produces' relationship an activity has with a resource. It is important to underline at this point, that the dependency link does not refer to dependency types i.e. 'fit, flow or sharing'. In essence, the dependency link details the simplest form of the dependency type, that being a link between one resource and one activity. It can be observed that it is the accumulation of these individual dependency links that make up the dependency type. For example, a flow dependency is delineated in figure 7.7, as the activity 'Produce_Production_Budget' uses the 'Sales_Budget' resource which the activity 'Produce_Sales_Budget' produces. The formulation of these dependency types (fit, flow and sharing) will be considered in more detail in section 7.7.6.1.

7.7.4. The Resource Model

Section 6.5 considered a taxonomy of resources suggested by Dellarocas (1996). Where applicable, various parts of this typology have been used within the CBOM resource model. Figure 7.8 below delineates the principle dimensions of this model. Dellarocas focused specifically upon a co-ordination perspective for integrating software components, therefore parts of his model are not suitable within the context of this research. When developing the CBOM resource model however, it was necessary to outline the various attributes required whilst keeping central their appropriateness for supporting tactical management requirements. The component parts of this model are considered in more detail throughout the remainder of this section.

Resource
- Resource_ID
- Resource_Name
- Activity_Designation := Activity.Activity_ID
- Shareability = (SHAREABLE | NON-SHAREABLE)
- Divisible = (DIVISIBLE | INDIVISIBLE)
- Consumability = (CONSUMABLE | NON-CONSUMABLE)
- Concurrency = (CONCURRENT | NON-CONCURRENT)
- Usability = (REUSABLE | NON- REUSABLE)
- Availability_Date
- Tangibility = (TANGIBLE | INTANGIBLE)
- Position
- Time_Span
- Resource_Accessible = (YES | NO)
- Resource_Usable = (YES | NO)

Figure 7.8. Resource model in CBOM

Resource_ID. The 'Resource ID' is used to distinguish between each resource in the same process. In short, it acts as a unique identifier. Its description with a possible value are shown below.

```
#Resource.Resource_ID;
Sales_Data.Resource_ID = "R1";
```

Resource_Name. This attribute simply captures the name of the resource. Its description and value relating to the 'Sales_Data' resource are given below.

```
#Resource.Resource_Name;
Sales_Data.Resource_Name = Sales_Data;
```

Activity_Designation. The 'Activity_Designation' is a derived attribute that takes its value from the 'Activity_ID' in the activity object. For example;

```
#Resource.Activity_Designation := #Activity.Activity_ID;
Sales_Data.Activity_Designation = Produce_Sales_Budget.Act001;
```

Shareability. Where one or more activities use a resource, then a sharing relationship can be recognised to exist (Malone and Crowston, 1994). Dellarocas (1996), mentions that Wyner and Zlotkin (1995) detail three notable elements that should be taken into account when considering shareable resources. These are 'Divisible, Concurrency and Consumability'. Consistent with the approach suggested by Wyner and Zlotkin (1995), this study also adopts these three areas. The shareability of the resource is given one of two possible values. These are expressed as;

```
#Resource.Shareability = (SHAREABLE | NON-SHAREABLE);
```

Where the resource 'Sales_Data' takes the value;

```
Sales_Data.Shareability = SHAREABLE;
```

Divisible. The divisible properties of the resource denote to what extent the resource can be divided between various activities. The divisible property simply denotes if the resource is divisible or non-divisible. This attribute is described thus;

```
#Resource.Divisible = (DIVISIBLE | NON-DIVISIBLE);
```

Which details that the resource 'Sales_Data' is indivisible;

```
Sales_Data.Divisible = INDIVISIBLE;
```

Consumability. A consumable resource cannot be reused. For example, the burning of raw materials. Non-Consumable resources on the other hand can be used ad infinitum, such as information or plans. CBOM details resource consumability by assigning the resource consumability attribute with a value equal to 'Consumable' or 'Non-consumable'. This is described as;

```
#Resource.Consumability = (CONSUMABLE | NON-CONSUMABLE);
```

With a possible value relating to the 'Sales_Data' resource;

```
Sales_Data.Consumability = NON-CONSUMABLE;
```

Concurrency. Concurrency details the simultaneous supply of a resource, to two or more activities regardless of its divisible state. This attribute has one of two values, namely, 'Concurrent' or 'Non-concurrent'. A description and value relating to the 'Sales_Data' resource are shown below.

```
#Resource.Concurrency = (CONCURRENT | NON-CONCURRENT);
Sales_Data.Concurrency = CONCURRENT;
```

Usability. The concept of resource usability has been the topic of discussion by many authors in the field of co-ordination theory (Malone et al, 1999; Dellarocas, 1996; Malone and Crowston, 1994; Crowston, 1994; Crowston, 1991). Resource usability determines a resource that can be reused. Information, drawings or plans may be considered reusable resources, whilst resources such as raw materials are not reusable as they are consumable. The CBOM defines usability as either 'Reusable' or 'Non-reusable'. The description of this attribute along with a corresponding value would take the form;

```
#Resource.Usability = (REUSABLE | NON-REUSABLE);
Sales_Data.Usability = REUSABLE;
```

Availability_Date. The 'Availability_Date' simply denotes the date the resource will be made available for use. Expressed in a date format, the description and a possible value are given below;

```
#Resource.Availability_Date;
Sales_Data.Availability_Date = "10OCT03";
```

Tangibility. This attribute refers to the tangibility status of the resource. In short, tangible resources are those that are physical in nature, such as raw materials. Intangible resources are those that are not physical, resources similar to data, information and knowledge could be considered intangible. Resource tangibility within CBOM is simply recorded as either 'Tangible' or 'Intangible'. For example;

```
#Resource.TANGIBILITY = (TANGIBLE | NON-TANGIBLE);
```

Where a possible value relating to the 'Sales_Data' resource would recognise;

```
Sales_Data.Tangibility = INTANGIBLE;
```

Position. The tactical management function may extend to managing tactical level processes throughout virtual organisations. It is feasible to suggest that participating companies will be responsible for carrying out activities that are consistent with their specialisation. From this perspective, resources may be positioned in different locations. This attribute draws parallels with the 'Transportability' attribute suggested by Dellorocas (1996) (see table 6.5). The position status therefore, details where a resource resides. This is shown as;

```
#Resource.Position;
Sales_Data.Position = "FHCF_Ltd";
```

Where the company name 'FHCF_Ltd' is the location of the resource 'Sales_Data'.

Time_Span. Dittrich (1991) (see table 6.2) hypothesises about the timespan of activities. This study extends the timespan issue to resources. Within a tactical level process, resources such as data may require updating. Therefore, the life span of a given resource may be short i.e. one process cycle, whilst other resources may be used for a slightly longer period such as three process cycles. From this point of view, a user defined value is used to detail the amount of cycles the resource will be used. A description and example value of this are shown below;

```
#Resource.Time_Span;
Sales_Data.Time_Span = "For_Duration_Of_Process";
```

Resource_Accessible. The 'Resource_Accessible' attribute simply denotes if the resource is accessible. Expressed as an enumerated attribute, it takes the form;

```
#Resource.Resource_Accessible = (YES | NO);
```

Where values relating to the 'Sales_Data' resource would recognise;

```
Sales_Data.Resource_Accessible = YES;
```

Resource_Usable. The 'Resource_Usable' attribute denotes if the resource is usable. Identical to the 'Resource_Accessible' attribute, the 'Resource_Usable' is also enumerated. A description and value of this attribute are given below;

```
#Resource.Resource_Usable = (YES | NO);
Sales_Data.Resource_Usable = YES;
```

7.7.5. The Actor Model

The co-ordination framework determines that activities either produce or use resources (Crowston, 1994; Malone and Crowston, 1994; Malone et al, 1995). CBOM represents this whilst modelling actors as belonging to activities. The principle dimensions of the Actors model are presented in figure 7.9. below.

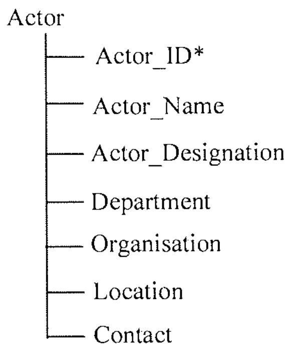

Figure 7.9. Actor Model in CBOM

The 'Actor_ID' uniquely identifies individual actors. Six further attributes also form the 'Actor' model. These are 'Actor_Name' (denotes the name of the actor), 'Actor_Designation' (details who or what will carry out the activity), 'Department' (details the department the actor belongs to), 'Organisation' (denotes the organisation the actor belongs to), 'Location' (details the location of the actor) and 'Contact' (denotes contact details). These attributes are described thus.

```
#Actor.Actor_ID*;
#Actor.Actor_Name;
#Actor.Actor_Designation;
#Actor.Department;
#Actor.Organisation;
#Actor.Location;
#Actor.Contact;
```

Where corresponding values relating to the 'Sales_Manager' actor would detail;

```
Sales_Manager.Actor_ID = "01SA";
Sales_Manager.Actor_Name = Sales_Manager;
Sales_Manager.Actor_Designation = "David_Gibson";
Sales_Manager.Department = "Sales_Department";
```

```
Sales_Manager.Organisation = "FHCF_Ltd";
Sales_Manager.Location = "London";
Sales_Manager.Contact = "Ext2938";
```

7.7.6. The Co-ordination Model

The overall purpose of the co-ordination model is to provide co-ordinated information based upon the data captured in the four object models. Figure 7.2 displayed the CBOM architecture as being in two parts. The first part represents the four object models, 'Process', 'Activity', 'Resource' and 'Actor'. The second level represents the co-ordination model, which consists of three views. Namely;

- **Dependency View.** Represents all 'Fit, Flow and Sharing' dependency types in the current process.

- **Interdependency View.** Represents all 'Reciprocal, Sequential and Pooled' interdependencies in the current process.

- **Constraint View.** Represents all 'Accessibility, Usability and Prerequisite' dependency qualifiers in the current process.

The attribute values that makeup these three views, are either derived or calculated from the 'Process', 'Activity', 'Resource' or 'Actor' models. Moreover, although no new data is required, new information is produced. The first of these views in the co-ordination model, the 'Dependency View', will be considered in further detail below.

7.7.6.1. The Dependency View

The 'Dependency View' represents all dependency type occurances in a particular process. These dependency types were represented in figure 6.7 as 'Fit', 'Flow' and 'Sharing'. The principle dimensions of the 'Dependency View' are presented in figure 7.10.

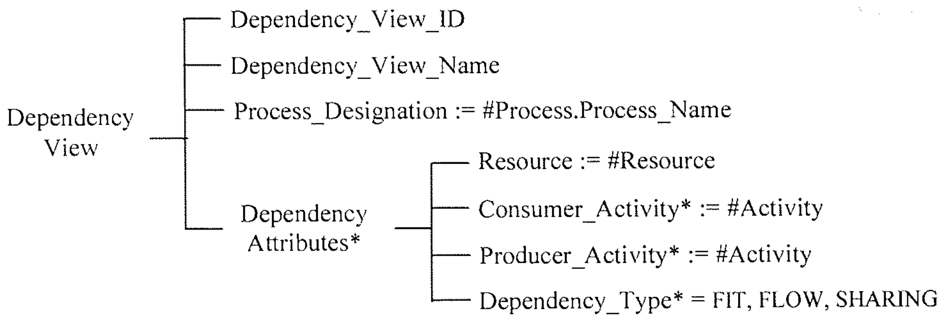

Figure 7.10. The Dependency View

The individual attribute values presented in figure 7.10 are considered below.

Dependency_View_ID. This attribute acts as the unique identifier for the dependency view. It is described with a representative value thus;

```
#Dependency_View.Dependency_View_ID;
ACF_Dependency_View_1.Dependency_View_ID = "DV001";
```

Dependency_View_Name. The 'Dependency_View_Name' details the name of the dependency view. This is described with a value relating to the 'ACF_Dependency_View_1' as;

```
#Dependency_View.Dependency_View_Name;
ACF_Dependency_View_1.Dependency_View_ID = ACF_Dependency_View_1;
```

Process_Designation. The 'Process_Designation' attribute details the current process in which either a fit, flow or sharing dependency type exists. This attribute is taken from the 'Process' object. This is described as;

```
#DependencyView.Process_Designation := #Process.Process_Name;
```

Where a corresponding value for this attribute details;

```
ACF_Dependency_View_1.Process_Designation = "Assess_Corporate_Finances";
```

Dependency_Attributes. This attribute details all sub-attributes that are required for outlining dependency occurances within a process. The 'Dependency_Attribute' consists of five nested attributes. These are considered in table 7.9 below.

Attribute Name	Purpose	Origin
Resource	Details the resource used in the dependency type.	Taken from the resource object.
Consumer_Activity	Details all activities that use the specificed resource.	Taken from the activity object.
Producer_Activity	Details all activities that produce the specificed resource.	Taken from the activity object.
Dependency_Type	Specifies the dependency type as 'Fit', 'Flow' or 'Sharing'.	Calculated in the 'Dependency View'.

Table 7.9. Dependency attributes

The description of this attribute takes the form;

```
#DependencyView.Dependency_Attributes* = (Resource := #Resource,
                         Consumer_Activity* := #Activity,
                         Producer_Activity*  := #Activity,
                         Dependency_Type* = FIT|FLOW|SHARING);
```

Where values corresponding to the 'Sales_Budget' resource represent;

```
ACF_Dependency_View_1.Dependency_Attributes = ("Sales_Budget",
                                     Produce_Production_Budget,
                                     Produce_Finance_Budget,
                                     "Produce_Sales_Budget",
                                     (FLOW,SHARING));
```

By analysing the example above, it can be observed that the 'Resource' attribute is represented with the value 'Sales_Budget', the 'Consumer_Activity' is equal to the 'Produce_Production_Budget' and the 'Produce_Finance_Budget'. The 'Producer_Activity' is represented by the 'Produce_Sales_Budget', and finally, the 'Dependency_Type' represents both 'Flow' and 'Sharing" dependency types. The 'Dependency_Type' is calculated based upon the values contained in the 'Dependency_Attributes'. For example, the values 'Flow' and 'Sharing' are detailed in the example above. This is owing to the producer activity "Produce_Sales_Budget" generating the "Sales_Budget" resource, and the consumer activities 'Produce_Production_Budget' and 'Produce_Finance_Budget' using the same resource.

The dependency patterns (Fit, Flow and Sharing) represented in figure 6.7, are constructed in their basic form using one resource and two activities. It is feasible to suggest that if more than two activities use a resource then a sharing dependency would exist, or if two activities contributed towards the production of a resource then a fit dependency would exist. Likewise, the same principles for the existence of multiple activities using and producing a resource at the same time would represent the occurance of a flow dependency. What remains constant however is the existence of one resouce in the dependency type. From this perspective, all dependency types are modelled based upon single resources. CBOM uses these same principles for deriving dependency types in any given tactical level process. In doing so, it uses the following rules when modelling these dependency types.

For Sharing Dependencies. If the 'Consumer_Activity' attribute is populated by two or more activity values then the 'Dependency_Type' will be populated with the value 'SHARING'.

For Fit Dependencies. If the attribute 'Producer_Activity' is populated by two or more activity values, then the 'Dependency_Type' will be populated with the value 'FIT'.

For Flow Dependencies. If the attribute 'Producer_Activity' is populated, and the attribute 'Consumer_Activity' is populated, then the 'Dependency_Type' shall be populated with the value 'FLOW'.

7.7.6.2. The Interdependency View

The principle dimensions of the 'Interdependency View' are shown in figure 7.11 below.

Interdependency_View
- Interdependency_View_ID
- Interdependency_View_Name
- Process_Designation := #Process.Process_Name
- Interdependency_Attributes*
  - Activity := #Activity
  - Resource_Dependent* := #Activity
  - Activity_Dependent* := #Activity
  - Sequential_Status = (SEQUENTIAL | NON-SEQUENTIAL) := #Activity
  - Reciprocal_Status = (RECIPROCAL | NON-RECIPROCAL) := #Activity
  - Pooled_Status = (POOLED | NON-POOLED) := #Activity

Figure 7.11. The Interdependency View

Section 6.8 detailed three interdependency types suggested by Thompson (1967). Namely, Pooled, Sequential and Reciprocal. These interdependency types have been addressed in the activity model in section 7.7.3. Whereby, the use of the attributes, 'Outside_Activity_Dependent', 'Dep_On_Resource', and 'Pooled_Qualifier', model the necessary data required for detailing interdependencies within a tactical level process. The use of the 'Interdependency View' is to take these attribute values and establish 'Pooled', 'Sequential' and 'Reciprocal' interdependency occurances.

Interdependency_View_ID. This attribute acts as the unique identifier for the interdependency view. It is described as;

```
#Interdependency_View.Interdependency_View_ID;
```

With a possible representative value equal to;

```
ACF_Interdependency_View_1.Interdependency_View_ID = "IDV001";
```

Interdependency_View_Name. The 'Interdependency_View_Name' details the name of the interdependency view. This is described with a possible value relating to the 'ACF_Interdependency_View_1';

```
#Interdependency_View.Interdependency_View_Name;
ACF_Interdependency_View_1.Interdependency_View_Name =
                                              ACF_Interdependency_View_1;
```

Process_Designation. This attribute is used to represent the process that the interdependency view is modelling. The description and corresponding value for this attribute can be observed below as;

```
#InterdependencyView.Process_Designation := #Process.Process_Name;
ACF_Interdependency_View_1.Process_Designation =
                                             "Assess_Corporate_Finances";
```

Interdependency_Attributes. All the attributes that are required to determine interdependency occurrences (Sequential, Reciprocal and Pooled) in a tactical level process, are listed in the 'Interdependency_Attributes'. For brevity, these are summarised in table 7.10 below.

Attribute Name	Purpose	Origin
Interdependency_ Attributes		
Activity	Denotes the current activity.	Derived form the activity object.
Resource_Dependent	Denotes if the current activity is dependent upon a resource produced by another activity.	Derived from the activity object. Uses the values in the 'Dep_On_Resource' attribute.
Activity_Dependent	Denotes if another activity is dependent upon the current activity.	Derived from the activity object. Uses the value in the 'Outside_Activity_Dependent' attribute.
Sequential_Status	Details the sequential status as either 'Sequential' or 'Non-Sequential'.	Derived from the activity object.
Reciprocal_Status	Details the sequential status as either 'Reciprocal' or 'Non-Reciprocal'.	Derived from the activity object.
Pooled_Status	Details the pooled status as being either 'Pooled' or 'Non-Pooled'.	Derived from the activity object. Using the values contained in the 'Pooled_Qualifier' attribute.

Table 7.10. Interdependency attributes

As detailed in section 6.8, where a reciprocal interdependency occurs, a sequential interdependency also occurs. From this perspective, a reciprocal interdependency is modelled in CBOM, when the current activity is dependent upon the output of another activity, and that activity is dependent upon the output of the current activity. A sequential interdependency occurs when the current activity is dependent upon the output of another activity only. Therefore, the 'Resource_Dependent' and 'Activity-Dependent' attributes represent all activities that are dependent upon the output of another activity, as well as those that are not. The 'Sequential_Status' and the 'Reciprocal_Status' are used to provide information as to where these types of dependencies occur, whilst a pooled interdependency is signified using the 'Pooled_Status' attribute. The description of the 'Interdepdency_Attributes' is given below;

```
#Interdependency_View.Interdependency_Attributes*(
       Activity := #Activity,
       Resource_Dependent* := #Activity,
       Activity_Dependent* := #Activity,
       Sequential_Status = (SEQUENTIAL | NON_SEQUENTIAL) := #Activity,
       Reciprocal_Status = (RECIPROCAL | NON_RECIPROCAL) := #Activity,
       Pooled_Status = (POOLED | NON-POOLED) := #Activity);
```

Where values relating to the 'ACF_Interdependency_View_1' correspond to;

```
ACF_Interdependency_View_1.Interdependency_Attributes =
                                             ("Produce_Sales_Budget",
                                              NULL,
```

```
(Produce_Production_Budget,
 Produce_Finance_Budget),
 NON_SEQUENTIAL,
 NON_RECIPROCAL,
 POOLED);
```

The above denotes the existence of a 'Pooled' interdependency type. In this case 'Sequential' and 'Reciprocal' types are not detected.

7.7.6.3. The Constraint View

The 'Constraint View' is the third view in the co-ordination model. The purpose of the 'Constraint View' is to represent occurances of 'Prerequiste', 'Accessibility' and 'Usability' constraints in a given tactical level process. The principle dimensions of this view are considered in figure 7.12.

```
Constraint View
├── Process_Designation := #Process
└── Constraint_Attributes*
    ├── Producer_Activity* := #Activity
    │   ├── Resource_Name := #Resource
    │   ├── Producer_Date := #Activity
    │   └── Resource_Created := #Resource
    ├── User_Activity* := #Activity
    │   ├── User_Date := #Activity
    │   ├── Resource_Accessible = (YES | NO) := #Resource
    │   └── Resource Usable = (YES | NO) := #Resource
    └── Constraint_Type = ACCESSIBILITY, USABILITY, PREREQUISITE
```

Figure 7.12. The Constraint View

Process_Designation. This attribute value acts in the same way as the 'Process_Designation' attribute in both the dependency and interdependency views. In short, it outlines the process name. A description of this attribute takes the form;

```
#DependencyConstraintView.Process_Designation := #Process;
```

Where a corresponding value relating to the 'ACF_Constraint_View_1' would recognise;

```
ACF_Constraint_View_1.Process_Designation = Assess_Corporate_Finances;
```

Constraint_Attributes. This attribute consists of two principle areas. Namely, 'Producer_Activity' and 'User_Activity'. These shall be considered in more detail below.

Producer_Activity. The 'Producer_Activity' attribute outlines the activity that produces the resource. This is described thus;

```
#Constraint_View.Constraint_Attributes.Producer_Activity* := #Activity(
                        Resource_Name:= #Resource,
                        Producer_Date := #Activity,
                        Resource_Created = YES|NO := #Resource);
```

Where possible values relating to the 'Produce_Sales_Budget' recognise;

```
ACF_Constraint_View_1.Constraint_Attribtes.Producer_Activity.
                 Produce_Sales_Budget = ("Sales_Budget","10OCT03",YES);
```

The 'Resource_Name' details the resource that is to be used by the 'User_Activity'. The 'Producer_Date' attribute highlights the date the activity or activities will produce the required resource. Finally, the 'Resource_Created' attribute, represents a boolean value that denotes if the resource has been created.

User_Activity. This attribute details the 'User_Date', 'Resource_Accessible' and 'Resource_Usable' attributes. The 'User_Activity' attribute outlines the activity that requires the use of a resource, whilst the 'User_Date' denotes the date on which the activity requires the use of the resource. The 'Resource_Usable' attribute details if the resource is usable, whilst the 'Resource_Accessible' attribute denotes if the resource required by the current activity is accessible.

Section 6.7.4.2, outlined that accessibility constraints occur when a resource is not in the right place and therefore is not accessible by the activity that requires its use. From this point of view, the activity can't execute, as it does not have access to the required resource. In order to demonstrate this, the 'Assess Corporate Finances' budget example has been augmented to demonstrate an accessibility constraint between the activities 'Produce_Sales_Budget', and 'Produce_Production_Budget'. Shown in figure 7.13 below, an extra activity between the 'Produce_Sales_Budget' and 'Produce_Production_Budget' named 'Approved_By_Sales_Manager' has been inserted. The example shows the activity 'Produce_Production_Budget' requiring the use of the resource 'Sales_Budget'. However, before it can be used, the 'Approved_By_Sales_Manager' activity must execute.

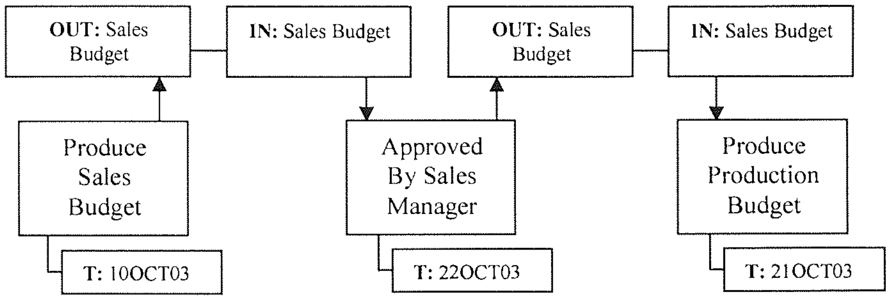

Figure 7.13. A Usability constraint

The 'Produce_Sales_Budget' executes on the 10th October 2003, the 'Produce_Production_Budget' executes on the 21st October 2003, however, the 'Approved_By_Sales_Manager' activity executes on the 22nd October 2003. Therefore, although the sales budget resource has been created, it is not accessible as the ' Approved_By_Sales_Manager ' activity has not been executed. In CBOM, descriptions and associated values of the 'User_Activity' attribute would take the form;

```
#Constraint_View.Constraint_Attributes.User_Activity* := #Activity(
                        User_Date, := #Activity,
                        Resource_Accessible = (YES | NO) := #Resource
                        Resource_Usable = (YES | NO) := #Resource);
```

Whilst corresponding values relating to figure 7.13 above would recognise;

```
ACF_Constraint_View.Constraint_Attributes.Produce_Prouction_Budget =
                                                ("10OCT03", NO, YES);
```

Where the resource is not accessible because the 'Approved_By_Sales_Manager' activity has not executed. However, the resource is usable, as it has already been produced by the 'Produce_Sales_Budget' activity.

Constraint_Type. Finally, the 'Constraint_Type' attribute highlights the type of constraints that exists in the current process. In this case, 'Accessibility', 'Usability' and 'Prerequisite'. This is described in CBOM as;

```
#DependencyConstraintView.Constraint_Attributes.
            Constraint_Type = (ACCESSIBILITY|USABILITY|PREREQUISITE);
```

Values relating to the example delineated in figure 7.13, detail the existence of an accessibility constraint, as the activity 'Produce_Production_Budget' does not have access to the 'Sales_Budget' resource. For instance;

```
ACF_Constraint_View_1.Constraint_Attributes.Constraint_Type = ACCESSIBILITY;
```

In constrast to an accessibility constraint, a usability constraint occurs when a recource is not ready for use when an activity requires it. Figure 7.6 provides an example of a usability constraint. The resource 'Prod_Sales_Data_Summary', is produced by both the 'Produce_Production_Budget' and 'Produce_Finance_Budget' activities, thereby signifying the existence of a 'Fit' dependency type. However, the activity, 'Produce_Corporate_Budget' uses the 'Prod_Sales_Data_Summary' resource. Therefore, if either the 'Produce_Production_Budget' or 'Produce_Finance_Budget' activities do not execute, then the 'Prod_Sales_Data_Summary' resource will not be available for use by the 'Produce_Corporate_Budget' activity. CBOM would model an instance of this as;

```
ACF_Constraint_View_2.Process_Designation = "Assess_Corporate_Finances";
ACF_Constraint_View_3.Constraint_Attributes.Produce_Production_Budget =
                          ("Prod_Sales_Data_Summary","21OCT03","NO");
ACF_Constraint_View_2.Constraint_Attributes.Produce_Finance_Budget =
                          ("Prod_Sales_Data_Summary","20OCT03","NO");
ACF_Constraint_View_2.Constraint_Attributes =("Produce_Corporate_Budget"
                                                          ,"NO","NO);
ACF_Constraint_View_2.Constraint_Attributes.Constraint_Type = USABILITY;
```

It could be argued that both usability and accessibility constraints are similar to each other. However, although the differences are small, both usability and accessibility provide a subtle change in emphasis as regargds the constraint type that occurs in a tactical level process. From this point of view, CBOM models 'Usability' as occuring when a resource cannot be used or is rendered unusable by an activity that is waiting to use it. From this perspective, it would be possible for a resource to have been created but it may not be fit for its purpose. Hence, a usability constraint would exist. In contrast, ' Accessibility', denotes a resource that is usable and therefore exists, but cannot be accessed by the activity that requires it.

Finally, a 'Prerequisite' constraint occurs when an activity cannot execute, because it is relying upon the production of a particular resource by another activity. An example of a prerequisite constraint can be observed in the 'Assess Corporate Finances' process. This is shown in figure 7.14 below.

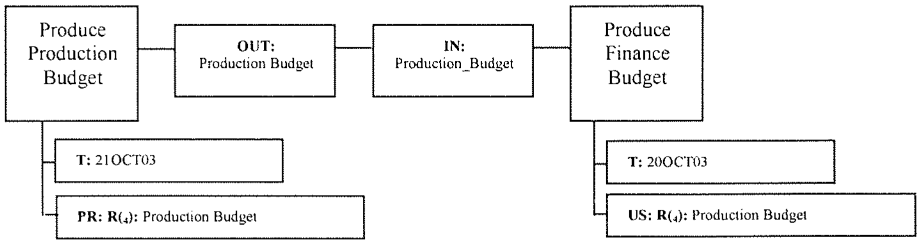

Figure 7.14. A Prerequisite constraint

Here, the 'Production_Budget' resource is produced on the 21st October 2003, whereas the 'Produce_Finance_Budget' requires the use of this on the 20th October 2003. As this is not possible, a prerequisite constraint occurs. This is represented in CBOM as;

```
ACF_Constraint_View_3.Process_Designation = "Assess_Corporate_Finances";
ACF_Constraint_View_3.Constraint_Attributes.Produce_Production_Budget =
                                        ("Production_Budget","21OCT03","NO");
ACF_Constraint_View_3.Constraint_Attributes.Produce_Finance_Budget =
                                                ("20OCT03","NO","NO");
ACF_Constraint_View_3.Constraint_Attributes.Constraint_Type = PREREQUISITE;
```

This example would also outline the existence of an accessibility and usability constraint as well. However, as mentioned earlier, it is possible for a resource to be both unusable and inaccessible. In this case, this could highlight a pre-requisite constraint also. However, it is possible for a resource to be both usable and accessible, yet a pre-requisite constraint could still occur, if for example, the execution of a particular activity was not achieved. This in short is where the emphasis is placed when assessing pre-requisite constraints in CBOM.

7.8. ASSESS CORPORATE FINANCES CBOM DATA LISTING

This section presents a full listing of the four CBOM objects and their corresponding attributes and values. These are taken from the 'Assess Corporate Finances' process example.

7.8.1. Process Model

Figure 7.12, details four activities that all belong to the 'Assess_Corporate_Finances' process. The associated values relating to the 'Assess_Corporate_Finances' process are shown below:

```
Process::Assess_Corporate_Finances
{
   Process_ID = "Pro001";
   Process_Name = "Assess_Corporate_Finances";
   Activity_Ownership = {Produce_Sales_Budget,
                         Produce_Production_Budget,
                         Produce_Finance_Budget,
                         Produce_Corporate_Budget};
   Process_Start_Date = "10OCT03";
}
```

7.8.2. Activity Model

The principle dimensions of the activity model are represented in section 7.7.3. The activity model is used to represent all activities within a given tactical level process. From this perspective, the activity model lists four different activities. Namely; 'Produce_Sales_Budget', 'Produce_Production_Budget', 'Produce_Finance_Budget' and 'Produce_Corporate_Budget'. The complete listings of these activities and their corresponding attributes are presented below.

```
Activity::Produce_Sales_Budget
{
   Activity_ID = "10";
   Activity_Name = "Produce_Sales_Budget";
   Time_Allocated  = "10_Days";
   Execution_Date = "10OCT03";
   Actual_Time_Taken = "10_Days";
   Time_Differential = "0_Days";
   Management = "01SA";
   Activity_Sequence = "1";
   Sub_Activity = NULL
   Pooled_Qualifier = CONTRIBUTE;
   Outside_Activity_Dependent = (Produce_Production_Budget,
                                 Produce_Finance_Budget);
   Dep_On_Resource = NULL;
   Dependency_Link = (
   Dep001("Sales_Data","SalesB_To_SalesD",USES),
   Dep002("Sales_Budget","SalesB_To_USalesB",PRODUCES));
}

Activity::Produce_Production_Budget
{
   Activity_ID = "11";
   Activity_Name = "Produce_Production_Budget";
   Time_Allocated  = "1_Days";
   Execution_Date = "21OCT03";
   Actual_Time_Taken = "1_Days";
   Time_Differential = "0_Days";
   Management = "01PR";
   Activity_Sequence = "2";
```

```
   Sub_Activity = NULL
   Pooled_Qualifier = CONTRIBUTE;
   Outside_Activity_Dependent = "Produce_Finance_Budget";
   Dep_On_Resource = (Produce_Sales_Budget, Produce_Finance_Budget);
   Dependency_Link = (
   Dep003 = ("Sales_Budget","ProdB_To_SalesB",USES),
   Dep004 = ("Production_Data","ProdB_To_ProdD",USES),
   Dep005 = ("Production_Budget","ProdB_To_UProdB",PRODUCES),
   Dep012 = ("Prod_Sales_Data_Summary","PProdB_To_PSDS",PRODUCES));
}

Activity::Produce_Finance_Budget
{
   Activity_ID = "12";
   Activity_Name = "Produce_Finance_Budget";
   Time_Allocated = "5_Days";
   Execution_Date = "20OCT03";
   Actual_Time_Taken = "5_Days";
   Time_Differential = "0_Days";
   Management = "01FM";
   Activity_Sequence = "3";
   Sub_Activity = NULL
   Pooled_Qualifier = CONTRIBUTE;
   Outside_Activity_Dependent = (Produce_Corporate_Budget,
                                 Produce_Production_Budget);
   Dep_On_Resource = (Produce_Sales_Budget,Produce_Production_Budget);
   Dependency_Link = (
   Dep006 = ("Sales_Budget","FinB_To_SalesB",USES),
   Dep007 = ("Production_Budget","FinB_To_ProdB",USES),
   Dep008 = ("Finance_Data","FinB_To_FinD",USES),
   Dep009 = ("Prod_Sales_Data_Summary","FinB_To_PSDS",PRODUCES));
}

Activity::Produce_Corporate_Budget
{
   Activity_ID = "13";
   Activity_Name = "Produce_Corporate_Budget";
   Time_Allocated  = "4_Days";
   Execution_Date = "25OCT03";
   Actual_Time_Taken = "4_Days";
   Time_Differential = "0_Days";
   Management = "01FD";
   Activity_Sequence = "4";
   Sub_Activity = NULL
   Pooled_Qualifier = (CONTRIBUTE);
   Outside_Activity_Dependent = NULL;
   Dep_On_Resource = ("Produce_Sales_Budget, Produce_Production_Budget,
                      Produce_Finance_Budget");
   Dependency_Link = (
       Dep010 = ("Prod_Sales_Data_Summary","CorpB_To_PSDS",USES),
       Dep011 = ("Corporate_Budget","PCorpB_To_CorpB",PRODUCES));
}
```

7.8.3. Resource Model

Figure 7.7 outlines seven resources. These comprise, 'Sales Data', 'Production Data', 'Finance Data', 'Prod_Sales_Data_Summary', 'Sales_Budget', 'Production_Budget' and 'Finance_Budget'. The resource model with its corresponding attribute values are listed below.

```
Resource::Sales_Data
{
   Resource_ID = "1";
   Resource_Name = "Sales_Data";
```

```
   Activity_Designation = "Act001";
   Shareability = SHAREABLE;
   Divisible = INDIVISIBLE;
   Consumability = NON-CONSUMABLE;
   Concurrency = INCONCURRENT;
   Usability = REUSABLE;
   Avalability_Date = "10OCT03";
   Tangibility = INNTANGIBLE;
   Position = "FHCF_Ltd";
   Time_Span = "For_Duration_Of_Process";
   Resource_Accessible = YES
   Resource_Usable = YES
}

Resource::Production_Data
{
   Resource_ID = "3";
   Resource_Name = "Production_Data";
   Activity_Designation = "Act002";
   Shareability = NON-SHAREABLE;
   Divisible = INDIVISIBLE;
   Consumability = NON-CONSUMABLE;
   Concurrency = INCONCURRENT;
   Usability = REUSABLE;
   Avalability_Date = "21OCT03";
   Tangibility = INTANGIBLE;
   Position = "FHCF_Ltd";
   Time_Span = "For_Duration_Of_Process";
   Resource_Accessible = YES
   Resource_Usable = YES
}

Resource::Finance_Data
{
   Resource_ID = "6";
   Resource_Name = "Finance_Data";
   Activity_Designation = "Act003";
   Shareability = NON-SHAREABLE;
   Divisible = INDIVISIBLE;
   Consumability = INCONSUMABLE;
   Concurrency = INCONCURRENT;
   Usability = REUSABLE;
   Avalability_Date = "20OCT03";
   Tangibility = INTANGIBLE;
   Position = "FHCF_Ltd";
   Time_Span = "For_Duration_Of_Process";
   Resource_Accessible = YES
   Resource_Usable = YES
}

Resource::Prod_Sales_Data_Summary
{
   Resource_ID = "5";
   Resource_Name = "Prod_Sales_Data_Summary";
   Activity_Designation = "Act002, Act003, Act004";
   Shareability = NON-SHAREABLE;
   Divisible = INDIVISIBLE;
   Consumability = NON-CONSUMABLE;
   Concurrency = INCONCURRENT;
   Usability = REUSABLE;
   Avalability_Date = "25OCT03";
   Tangibility = INTANGIBLE;
   Position = "Sales_Department";
   Time_Span = "For_Duration_Of_Process";
   Resource_Accessible = YES
   Resource_Usable = YES
}
```

```
Resource::Sales_Budget
{
  Resource_ID = "2";
  Resource_Name = "Sales_Budget";
  Activity_Designation = "Act001, Act002, Act003";
  Shareability = SHAREABLE;
  Divisible = INDIVISIBLE;
  Consumability = NON-CONSUMABLE;
  Concurrency = INCONCURRENT;
  Usability = REUSABLE;
  Avalability_Date = "20OCT03";
  Tangibility = INTANGIBLE;
  Position = "FHCF_Ltd";
  Time_Span = "For_Duration_Of_Process";
  Resource_Accessible = YES
  Resource_Usable = YES
}

Resource::Production_Budget
{
  Resource_ID = "4";
  Resource_Name = "Prod_Sales_Data_Summary";
  Activity_Designation = "Act002, Act003";
  Shareability = SHAREABLE;
  Divisible = INDIVISIBLE;
  Consumability = NON-CONSUMABLE;
  Concurrency = INCONCURRENT;
  Usability = REUSABLE;
  Avalability_Date = "21OCT03";
  Tangibility = INTANGIBLE;
  Position = "FHCF_Ltd";
  Time_Span = "For_Duration_Of_Process";
  Resource_Accessible = YES
  Resource_Usable = YES
}


Resource::Corporate_Budget
{
  Resource_ID = "7";
  Resource_Name = "Prod_Sales_Data_Summary";
  Activity_Designation = "Act004";
  Shareability = SHAREABLE;
  Divisible = INDIVISIBLE;
  Consumability = NON-CONSUMABLE;
  Concurrency = INCONCURRENT;
  Usability = REUSABLE;
  Avalability_Date = "26OCT03";
  Tangibility = INTANGIBLE;
  Position = "FHCF_Ltd";
  Time_Span = "For_Duration_Of_Process";
  Resource_Accessible = YES
  Resource_Usable = YES
}
```

7.8.4. Actor Model

Figure 7.6 defines four actors. These four instances are presented below.

```
  #Actor.Location
  #Actor.Contact


Actor::Sales_Manager
{
  Actor_ID = "01SA"
```

```
   Actor_Name =  Sales_Manager
   Actor_Designation = "David_Gibson"
   Department = "Sales_Department"
   Organisation = "FHCF_Ltd"
   Location = "London"
   Contact = "Ext2938"
}

Actor::Production_Manager
{
   ActorID = "01PR";
   Actor_Name = "Production_Manager";
   Actor_Designation = "Steve_Johnson";
   Department = "Production_Department";
   Organisation = "FHCF_Ltd";
   Location = "London";
   Contact = "Ext2976";
}

Actor::Finance_Manager
{
   ActorID = "01FM";
   Actor_Name = "Finance_Manager";
   Actor_Designation = "Steve_Johnson";
   Department = "Finance_Department";
   Organisation = "FHCF_Ltd";
   Location = "London";
   Contact = "Ext2998";
}

Actor::Finance_Director
{
   ActorID = "01FD";
   Actor_Name = "Finance_Director";
   Actor_Designation = "Dan_Malone";
   Department = "Finance_Department";
   Organisation = "FHCF_Ltd";
   Location = "London";
   Contact = "Ext2999";
}
```

7.8.5. The Co-ordination Model

The principle dimensions of the co-ordination model are discussed in section 7.7.6 above. The three views that comprise the co-ordination model are summarised below.

7.8.5.1. The Dependency View

As discussed in section 7.7.6.1, the principle aim of the 'Dependency View', is to highlight to the tactical level manager, all occurrences of 'Fit, Flow and Sharing' dependency types within a particular tactical level process. A full listing of the values associated to this view relating to the 'Assess Corporate Finances' process example are given below.

```
Dependency_View::ACF_Dependency_View_1
{
Dependency_View_ID = "DV001"
Dependency_View_Name = "ACF_Dependency_View_1"
Process_Designation = "Assess_Corporate_Finances";
Dependency_Attributes = ("Prod_Sales_Data_Summary",
                         "Produce_Corporate_Budget",
                         "Produce_Production_Budget,
                          Produce_Finance_Budget",
                          Fit,Flow);
}

Dependency_View::ACF_Dependency_View_2
{
Dependency_View_ID = "DV002"
Dependency_View_Name = "ACF_Dependency_View_2"
Process_Designation = "Assess_Corporate_Finances";
Dependency_Attributes = ("Sales_Budget",
                         "Produce_Production_Budget",
                         ("Produce_Finance_Budget",
                         "Produce_Sales_Budget")
                          Flow,Sharing);

}
Dependency_View::ACF_Dependency_View_3
{
Dependency_View_ID = "DV003"
Dependency_View_Name = "ACF_Dependency_View_3"
Process_Designation = "Assess_Corporate_Finances";
Dependency_Attributes = ("Production_Budget",
                         "Produce_Production_Budget",
                         "Produce_Finance_Budget",
                          Flow);
}
```

7.8.5.2. The Interdependency View

The 'Interdependency View', was discussed in section 7.7.6.2. A complete listing for all interdependency occurrences relating to the 'Assess Corporate Finances' process are listed below.

```
Interdependency_View::ACF_Interdependency_View_1
{

   Interdependency_View_ID = "IDV001";
   Interdependency_View_Name = "ACF_Interdependency_View_1";
   Process_Designation = "Assess_Corporate_Finances";
   Interdependency_Attributes = ("Produce_Sales_Budget",
                                 NULL,
                                ("Produce_Production_Budget,
                                 "Produce_Finance_Budget"),
                                 NON_SEQUENTIAL,
                                 NON_RECIPROCAL,
                                 POOLED);
}

Interdependency_View::ACF_Interdependency_View_2
{
   Interdependency_View_ID = "IDV002";
   Interdependency_View_Name = "ACF_Interdependency_View_2";
   Process_Designation = "Assess_Corporate_Finances";
   Interdependency_Attributes = ("Produce_Production_Budget",
                                "Produce_Sales_Budget",
                                "Produce_Finance_Budget",
```

```
                                                SEQUENTIAL,
                                                RECIPROCAL,
                                                POOLED);
}

Interdependency_View::ACF_Interdependency_View_3
{
   Interdependency_View_ID = "IDV003";
   Interdependency_View_Name = "ACF_Interdependency_View_3";
   Process_Designation = "Assess_Corporate_Finances";

   Interdependency_Attributes = ("Produce_Finance_Budget";
                                 ("Produce_Sales_Budget",
                                  "Produce_Production_Budget"),
                                  "Produce_Corporate_Budget",
                                   SEQUENTIAL,
                                   RECIPROCAL,
                                   POOLED);
}

Interdependency_View::ACF_Interdependency_View_4
{
   Interdependency_View_ID = "IDV004";
   Interdependency_View_Name = "ACF_Interdependency_View_4";
   Process_Designation = "Assess_Corporate_Finances";

   Interdependency_Attributes = ("Produce_Corporate_Budget";
                                 ("Produce_Sales_Budget",
                                  "Produce_Production_Budget",
                                  "Produce_Finance_Budget");
                                   NULL;
                                   SEQUENTIAL;
                                   NON_RECIPROCAL;
                                   POOLED);
}
```

7.8.5.3. The Constraint View

The 'Constraint View', originally discussed in section 7.7.6.3, details three types of constraint. These are presented in full below.

```
Constraint_View::ACF_Constraint_View_1
{
   Process_Designation = "Assess_Corporate_Finances";


   Constraint_Attributes.Producer_Activity = Produce_Sales_Budget(
                                                  "Sales_Budget",
                                                  "10OCT03",
                                                  YES);
   Constraint_Attributes.User_Activity =   Produce_Prouction_Budget(
                                                  "10OCT03",
                                                  NO,
                                                  YES);
   Constraint_Attributes.Constraint_Type = ACCESSIBILITY;
}

Constraint_View::ACF_Constraint_View_2
{
   Process_Designation = "Assess_Corporate_Finances";
   Constraint_Attributes.Producer_Activity = Produce_Production_Budget(
                                     "Prod_Sales_Data_Summary",
                                     "21OCT03",
```

```
                                              "NO");
   Constraint_Attributes.Producer_Activity = Produce_Finance_Budget(
                                              "Prod_Sales_Data_Summary",
                                              "20OCT03",
                                              "NO");
   Constraint_Attributes.User_Activity =("Produce_Corporate_Budget",
                                              "NO",
                                              "NO);
   Constraint_Attributes.Constraint_Type = USABILITY;
}

Constraint_View::ACF_Constraint_View_3
{
   Process_Designation = "Assess_Corporate_Finances";
   Constraint_Attributes.Producer_Activity = Produce_Production_Budget(
                                                  "Production_Budget",
                                                  "21OCT03",
                                                  "NO");
   Constraint_Attributes.User_Activity = Produce_Finance_Budget =
                                                 ("20OCT03",
                                                  "NO",
                                                  "NO");
   Constraint_Attributes.User_Activity =("Produce_Corporate_Budget",
                                                  "NO",
                                                  "NO);
                                                  "NO");
   Constraint_Attributes.Constraint_Type = PREREQUISITE;
}
```

7.7. CONCLUSIONS

This chapter has presented the constituent parts of the 'Co-ordinated Business Object Model' (CBOM). It has shown that by leveraging knowledge from co-ordination theory, it is possible to add richness to the business object model, when modelling tactical level processes. Specifically, this chapter has outlined that the CBOM represents a two-layer architecture. The first layer captures information from tactical level processes in four object models. These are identified as; 'Process, 'Activity', 'Resource' and 'Actor'. The second layer consists of a higher level co-ordination model, which returns information based upon the data captured in the four object models in the first layer.

Finally, in order for the concepts contained within the CBOM to be of any practical use to a tactical level manager, it is necessary to provide an application that has the ability to support the CBOM constructs in a software environment. Consequently, a prototype system needs to be developed as a tool for supporting the CBOM implementation. This shall be the primary focus of chapter eight.

CHAPTER 8

The Co-ordinated Business Object Support System (CBOSS)

8.1. INTRODUCTION

Chapter seven outlined the constituent parts of the Co-ordinated Business Object Model (CBOM). However, by itself, the CBOM is of little use to a prospective tactical level manager without an interface with which he/she can interact. Therefore, the Co-ordinated Business Object Support System (CBOSS) has been developed as a demonstrator application, with the ability to support the CBOM implementation. Furthermore, CBOSS not only supports the modelling constructs considered in the previous chapter, but it also provides the necessary functionality that is required for providing information to tactical level managers. As a consequence, CBOSS also provides an environment with which to test the CBOM. Issues regarding testing however, will be considered in further detail in chapter nine.

This chapter will outline the principle dimensions of the CBOSS system. In doing so, it will detail the systems architecture as well as outlining the salient parts of its functionality for supporting the development and analysis of tactical level processes.

8.2. THE CBOSS ARCHITECTURE

As mentioned previously, the CBOSS system is a demonstrator application. The full listing for this program is presented in appendix A. The CBOSS architecture however, is centred around a traditional two-tier systems architecture (Lhotka, 1998). The component parts of this are presented in figure 8.1 below. As is the case with a two-tier architecture, the overall system encompasses two levels that separate the 'Application layer' from the 'Database layer'.

8.2.1. Application Layer

The application level encompasses three levels. Namely; 'User Services Tier', 'Business Services Tier' and 'Data Services Tier'. These are considered in further detail below.

8.2.1.1. User Services Tier

The user services tier typically entails a graphical user interface (GUI), as part of the visual interface with which the tactical level manager will interact with. In short, it connects the user with the system. In doing so, data is captured and information that is used for decision making is presented to the user. Finally, the user services tier interacts with the rest of the system by making requests to the business and data services tier. This allows all levels of management to execute business tasks.

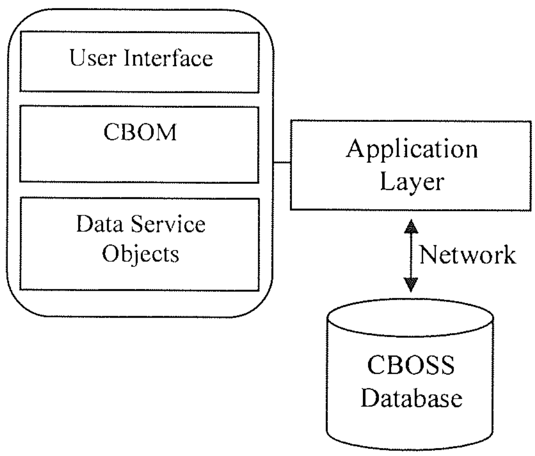

Figure 8.1. The CBOSS architecture

8.2.1.2. Business Services Tier

The business services tier executes a business task based upon requests from either the tactical manager or other business services triggered within CBOSS. This is achieved by making a request upon the data services tier and applying the formal procedures and business rules upon the requested data. The CBOM resides at the business services tier and therefore, the business rules pertaining to co-ordination are applied at this level. The business tier isolates the user from having direct interaction with the database. Moreover, as business rules within the CBOM may be prone to updates as business tasks change, the business services tier within CBOSS is composed primarily of components that are physically separate from the application logic.

8.2.1.3. Data Services Tier

The data services tier usually manages those services that are used for accessing and updating data. This tier interacts with the business services tier when requests for data are made. Within the CBOSS system, the data services allow the data structure to be modified based upon decisions made by tactical level managers. Such modifications do not effect business or user services.

8.2.2. CBOSS Database

The CBOSS database is used to hold all low level data relating to tactical level processes. Recognising a relational database structure, the CBOSS database has been developed using Microsoft Access 97.

8.3. INTERFACE DESIGN

The CBOSS interface provides the functionality for the user to interact with the CBOM. Figure 8.2 provides an example of the CBOSS main screen. The diagram points to six areas. There are considered in further detail below.

Development Area. Here, a tactical level manager would develop and construct a tactical level process. The constructs used to build a process consist of 'Activities', 'Resources', 'Dependencies' and the activity 'Uses' and 'Produces' relationship nodes.

Process Browser. The CBOSS interface contains a process browser that can be used to select an already created tactical level processes. Upon selecting the process, the representative activities and resources are displayed in the development area. This process/activity relationship is consistent with the CBOM model delineated in figure 7.2, in so much as all activities belong to higher level processes (Malone et al, 1999; Malone & Crowston, 1994; Camarinha-Matos, 2001).

Menu. The menu displays all functions relating to the CBOSS system. A full list of these are available in appendix C.

Active Database. This shows which database is currently open.

Colour Code. This allows the user to colour individual activities in order to differentiate between them. This allows different organisations or departments to be distinguished from each other.

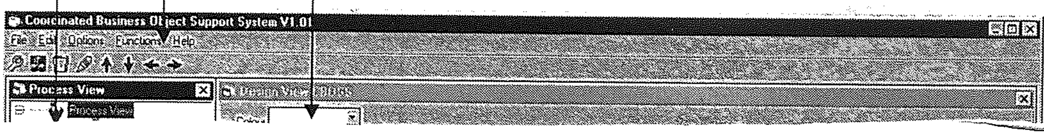

Figure 8.2. The CBOSS user interface

8.4. CO-ORDINATION CONSTRUCTS

Up to now, section C has presented the constituent parts of the CBOM. Chapter eight so far as presented the basic outline of the CBOSS. At this point in the thesis it is perhaps useful to outline what functionality in CBOM is implemented in CBOSS. In doing so this section will make reference to earlier chapters where this justifies design decisions relating to the functionality of CBOSS.

Section 6.3 outlined a notation that has been developed throughout the course of this study, which can be used for graphically representing co-ordination processes in diagrams. The notation presented in table 8.1 below, provides a description of the notation used by the CBOM in contrast to the elements that are used when representing co-ordination diagrams using the CBOSS system.

Co-ordination Element	CBOSS Symbol	CBOM Symbol
Process	Set In Process Browser	(NA)

Table 8.1. CBOM and CBOSS representative symbols

The symbols shown in the table above only form the main parts of constructing processes in the CBOSS system. As the four object models (Processes, Activities, Resources and Actors) between them contain a large number of attributes, it has been essential to build the 'drill-down' facility into the constituent CBOSS forms. In essence, if a user double clicks on an activity they can access the activity attributes, the same principles apply to resources and dependencies. For the constructs above to be used and understood by the user, it is important that they are presented to the user in a form they find easy to use. In essence, the CBOSS symbols outlined in table 8.1 are designed to draw together the salient parts of the CBOM model, whilst presenting them to the user in a form that they can understand. In short, the symbols above are designed from the attribute values outlined throughout the co-ordination model in section 7.7. However, from an academic perspective, and also to provide greater understanding for the reader and indeed to also clarify the linkage between the CBOM and it's construct representation in CBOSS, the remainder of this section will specifically outline the significance of every attribute outlined in the four object models (see sections 7.7.2 to 7.7.5), whilst outlining their purpose and corresponding representation in CBOSS.

Process. Processes were outlined in section 6.4. Significantly, Malone et al (1999) outlined the role of processes in an organisational context. Furthermore, the same author also presented the process compass as a means highlighting different sub-activities that belong to a process, along with outlining processes that the current process belongs to. The process compass can also be used to represent process generalisation and specialisation. Chapter seven outlined the role of processes in the CBOM in section 7.7.2. The CBOM, in modelling processes outlines four attributes, these are 'Process_ID', 'Process_Name', 'Activity_Ownership' and 'Process_Start_Date'. These attributes are outlined in table 8.2.

process starts. form.

Table 8.2. The Process Object In CBOSS

Activity. Activities like processes were outlined in chapter six. Specifically, section 6.4 presented an outline of activities and their relationships to processes. Chapter seven considered the role of activities within the CBOM model. Here section 7.7.3 detailed the activity object model which is part of the co-ordination model. The constituent parts of the activity model along with the individual attribute purposes and representation in CBOSS is shown in table 8.3 below.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
Activity_ID	To provide the user with an individual id for representing the activity. This is useful if an activity has the same name.	Activity box (shown in table 8.1)
[illegible]	'Produces' represents an activity that produces a resource.	This attribute is also set in the 'USES' or 'PRODUCES' check box in the Dependency Panel.

Table 8.3. The Activity Object In CBOSS

Resources. Resources were outlined in section 6.5. Throughout this section, a great deal of emphasis was placed upon the resource taxonomy presented by Dellorocas

(1996). Although not the same, the CBOM borrows some of the concepts from the work by Dellorocas. However as noted in section 6.5, Dellorocas developed a resource taxonomy based around a co-ordination perspective for integrating software components. Therefore, some of the attribute values in this model would not be appropriate in this work. As noted in table 7.3 in section 7.3 most of the attributes in the R*esource* object originate from the co-ordination domain, with the exception of Resource_ID', 'Resource_Name', 'Availability_Date', 'Position' and 'Timespan'. Both 'Resource_ID' and 'Resource_Name' are required by CBOSS in order to identify a particular resource. However. 'Availability_Date', was added as it was crucial for testing purposes, as the date the resource would be made available would be required, as would the position of the resource (*Position*), along with the amount of iterations the resource would be required within a given process (Timespan). The existence of such attributes were determined simply by the nature of tactical level processes. The *Resources* object in the CBOM was presented in section 7.7.4. Table 8.4 below outlines the purpose of the individual attribute fields pertaining to the resource object.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
	accessible.	section in the Resource Overview
Resource_Usable	Represents if the resource is usable.	Text Box in the Usability and Accessibility section in the Resource Overview

Table 8.4. The Resource Object In CBOSS

Actors. Actors were originally outlined in section 6.6. The need to have actor assignments in tactical level processes, is essential as someone or something executes a given activity. As table 7.4 shows, all of the attributes assigned to the *Actor* model were added by the author. In essence, these attributes are used to hold simple values that are used to recognise a particular actor. The *Actor* object model was originally outlined in section 7.7.5. A review of the actor attributes and their purpose and representation in CBOSS are presented in table 8.5 below.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
	Represents the Activity ID. In CBOSS this is shown	

		section in the Activities & Actors form.

Table 8.5. The Actor Object In CBOSS

All the attributes in the four object models as section 7.7.6 outlines, are used by the co-ordination model. This co-ordination model is composed of three views. Namely, *Dependency, Interdependency* and *Constraint* View.

Dependencies. The notion of dependencies between activities were originally outlined in section 6.7. The dependency types suggested by Zlotkin (1995), outlined three main dependency types. These are 'Fit', 'Flow' and 'Sharing'. A dependency model does not exist in the CBOM. Dependencies are simply processed based upon the data captured in the *Process, Activity, Resource* and *Actor* models. Section 7.7.6.1 in chapter seven outlined the parameters of dependencies within CBOM. Dependencies in CBOSS are represented in the 'Dependency Panel'. This is where the dependencies are created and deleted. Table 8.6 below outlines the dependency attributes along with a review of how they are represented in CBOSS.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
Dependency_View_ID	Assigns an ID to a dependency type.	In the Dependency View ID Text box in the Dependency View Panel.
Dependency_View_Name	Assigns a Name to a dependency type.	In the Dependency View Name Text box in the Dependency View Panel.
Process_Designation	Assigns a dependency type to the current process.	Shown in the Process Browser
Resource	Outlines which resource is part of the dependency type.	Shown in the Resource text box in the Dependency Panel.
Consumer_Activity	Represents the activity the consumes the resource.	Shown in the Activity text box in the Dependency Panel. The 'Uses' check box will be set to true.
Producer_Activity	Represents the activity the produces the resource.	Shown in the Activity text box in the Dependency Panel. The 'Produces' check box will be set to true.
Dependency_Type	Outlines the dependency type as being wither 'Fit', 'Flow' or 'Sharing'.	Shown in the Dependency Type Text box in the Dependency View Panel.

Table 8.6. Dependency Attributes

Interdependencies. Interdependencies were introduced in section 6.8. The same section outlined three interdependency types that were outlined by Thompson (1967). These were 'Pooled', 'Sequential' and 'Reciprocal'. Also presented in section 6.8 were three interdependency types as outlined by Kim, Myung & Emdad (2001) as 'Common Model', 'Pre-requisite' and 'Concurrent'. These interdependency types upon analysis translate to those presented by Thompson. Chapter seven outlined interdependencies within the CBOM in section 7.7.6.2. The attributes that are part of the interdependency model outlined in figure 7.11 are presented in table 8.7 below, along with a description as to where these attributes are represented in the CBOSS.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
Interdependency_View_ID	Represents the ID of the current interdependency.	Shown in the Interdependency ID test box in the Interdependency View Panel.
Interdependency_View_Name	Assigns a name to the current interdependency.	Shown in the Interdependency Name text box in the Interdependency View Panel.

Aston University

Illustration removed for copyright restrictions

Table 8.7. Interdependency Attributes

Constraint Types. Finally, constraint types were presented in section 6.7.4. Three constraint types were outlined as; 'Usability', 'Accessibility' and 'Pre-requisite'. These constraint types are represented in the constraint view model in the CBOM. This is outlined in section 7.7.6.3. A description of where the constraint attributes listed in figure 7.12. are represented in the CBOSS are given in table 8.8 below.

Co-ordination Attribute	Purpose In CBOSS	Representation In CBOSS
	accessible.	in the Resource Overview

Aston University

Illustration removed for copyright restrictions

Resource_Usable	Represents if the resource is usable.	Text Box in the Usability and Accessibility section in the Resource Overview
Constraint_Type	Represents the constraint type as either 'ACCESSIBILITY', 'USABILITY' or PREREQUISITE'.	Shown in the Accessibility, Usability and Pre-requisite text boxes in the Constraint Attributes section in the Constraint Analysis panel.

Table 8.8. Constraint Attributes

8.5. CO-ORDINATION MODELING

Developing tactical level processes in CBOSS, is carried out using the tool set attached to the development area. This allows the user to create processes, and input data into the CBOM. The principle dimensions of this procedure are outlined below.

8.5.1. Representing Co-ordination Models

Figure 8.3 provides a representation of the 'Assess Corporate Finances' process example presented in section 7.6. Delineating four activities and seven resources, the user can quickly create tactical level processes and assign dependency links between activities and resources.

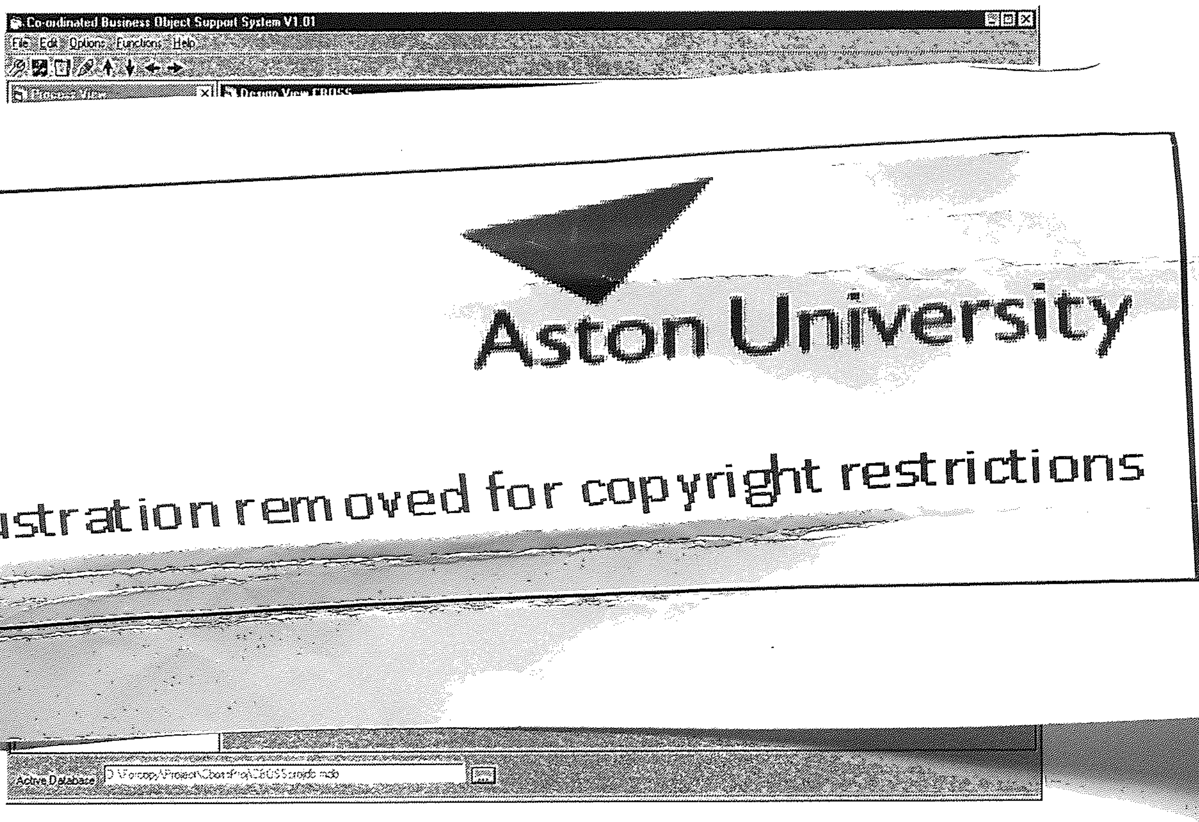

Figure 8.3. Co-ordination model

Circles next to the activity determines that the activity uses a resource. Squares on the other hand represent 'Produces' relationships. For example, the activity 'Produce Corporate Budget' produces the resource 'Corporate Budget'. Further data regarding activities and resources can be attained by the user, by simply drilling down on an activity or resource. Information regarding processes however, is shown in the process browser.

8.5.2. Process Representation

Processes are created by the user and are identified in the process browser. This is shown in figure 8.4 below. The CBOSS system displays a number of processes that can be analysed and re-modified by the tactical level manager at any one time. Upon selecting a process, the relevant parts of the process i.e. its activities, resources, relationships and dependency links are shown in the development area.

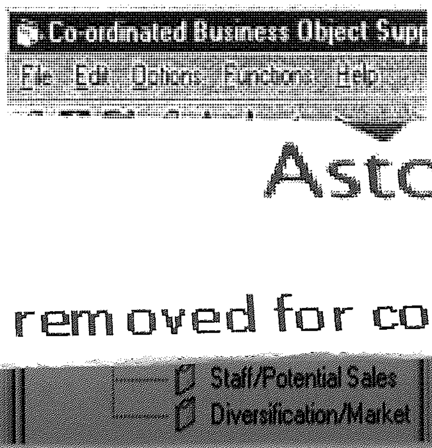

Figure 8.4. Process browser

8.5.3. Activity Representation

The data presented in figure 8.5. below, captures the principle areas of both the activities and actors models as discussed in sections 7.7.3 and 7.7.5. As activities are executed by actors, it is considered easier to present the data in these two models together on the same form. This way, it is apparent which actor is assigned to which activity. To view this data, the user simply selects an activity in the process. In essence, the user 'drills down' upon an activity in the development area. The 'Activity & Actors' form presented in the diagram above, is partitioned into six areas. The 'Activity' section details the 'Activity ID' and the 'Activity Name'.

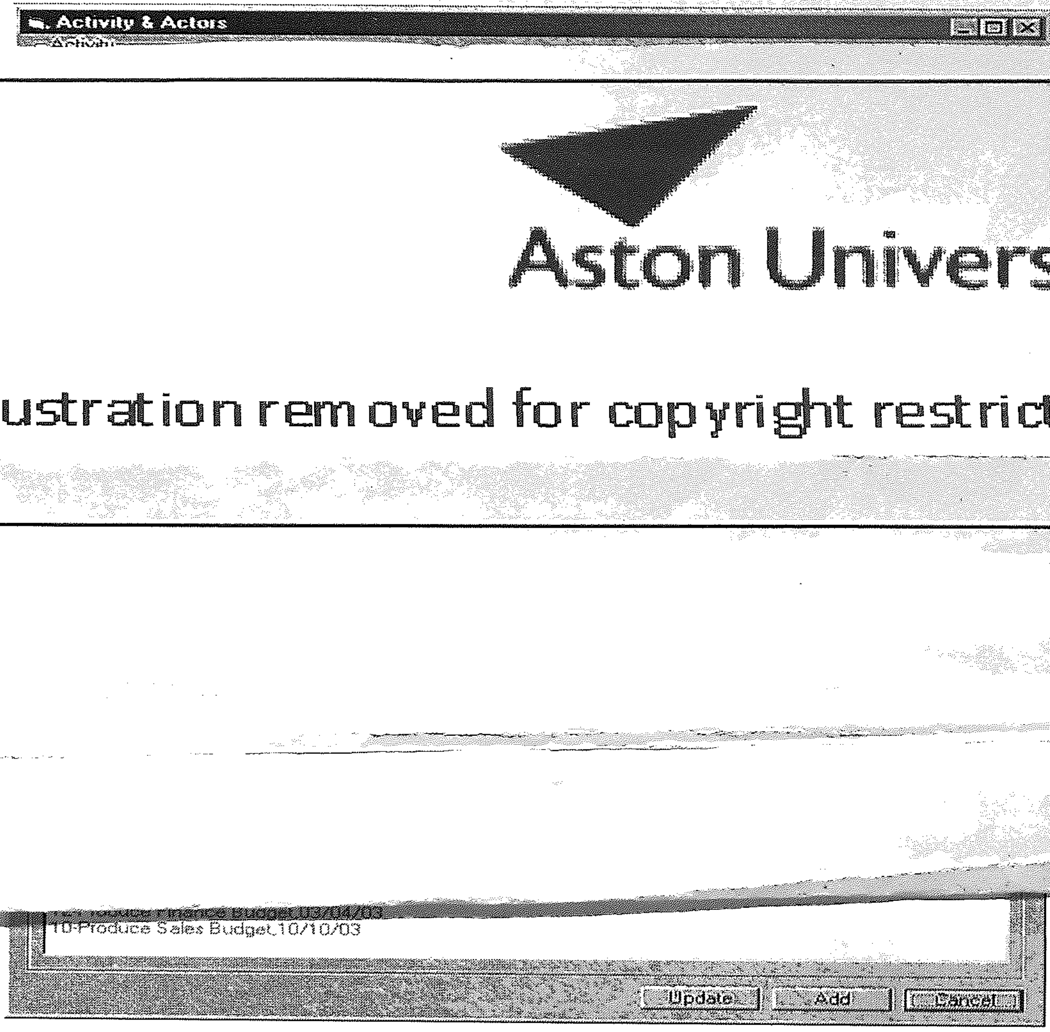

Figure 8.5. Activity and Actors in CBOSS

Consistent with the activity model presented in figure 7.4, various interdependency attributes are captured in the form. For instance, the 'Contribute and Non-Contribute' status of the activity are captured in the activity model as well as the names of other activities that are dependent upon the current activity. Furthermore, this form also highlights activities which the current activity is dependent upon. The 'Sub-Activity Details', outlines any activities in the current process that are sub-activities of the current activity. Although there are no examples stated in the figure above, it can be observed that activities such as 'Produce English Sales Budget Version' and 'Produce French Sales Budget Version' may be regarded as sub activities to the 'Produce Sales Budget'. The 'Time Details' section, represents all attribute values that are used to denote time details relating to the activity. The 'Managed' section relates to the actor who is responsible for the activity. Details regarding the actor are obtained from the actor object model. This was discussed in section 7.7.5. The 'Resource Coupling' section provides information as to which resources the current activity uses and produces. Finally, the 'Sequence Details' section, shows the sequence of execution for all activities in the current process.

8.5.4. Resource Representation

The 'Resource Overview' form shown in figure 8.6 below, summarises the main elements of the resource model. This form is partitioned into three areas. The first area outlines the resource name and corresponding identification. The second area outlines the resource details. Finally, the third area outlines the usability and accessibility state of the resource.

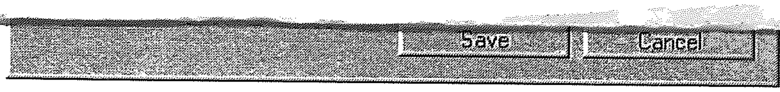

Figure 8.6. The Resource Overview

The attributes displayed in the diagram above are consistent with figure 7.8. These attribute areas were discussed in detail in section 7.7.4. The diagram above details the resource attributes that relate to the 'Sales Data' resource.

8.5.5. Dependency Representation

The 'Dependency Panel' is used to create dependency links from activities to resources. Dependency links in the CBOSS system are represented by a dependency name. It is at this stage in the design process, when the user is creating dependency links, that they assign the 'Uses' or 'Produces' status to the dependency link. This is shown in figure 8.7.

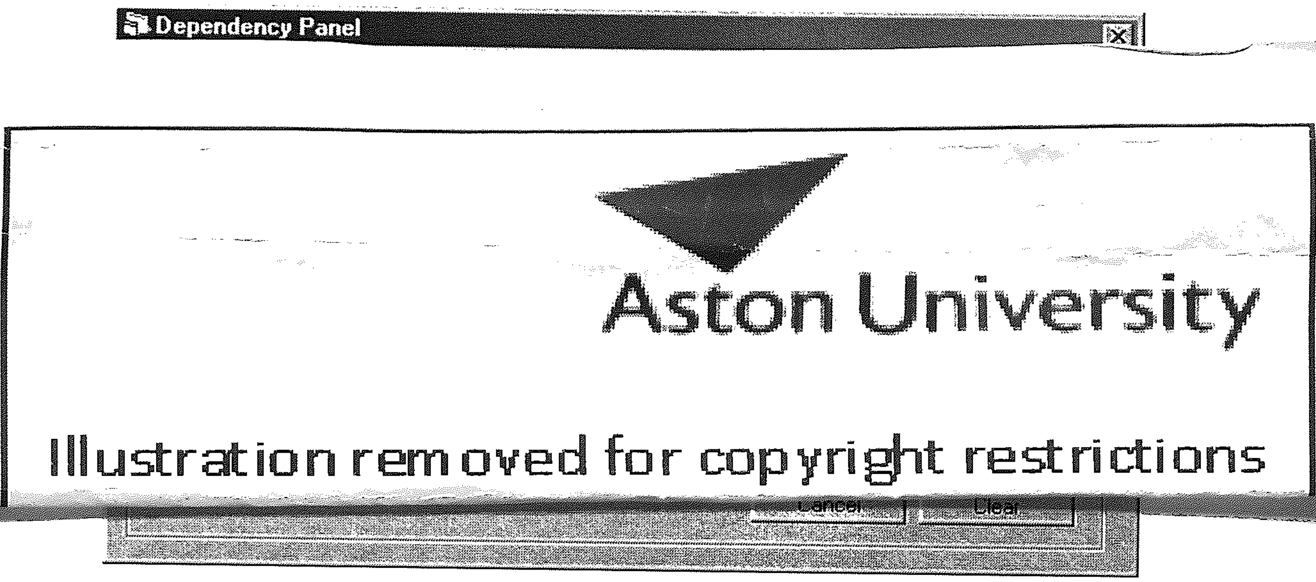

Figure 8.7. The Dependency Panel

Figure 8.9 below provides an example as to how CBOSS denotes 'Uses' and 'Produces' relationships. Here, the activity 'Produce Production Budget' produces the resource 'Production Budget', the activity 'Produce Finance Budget' then uses the activity 'Produce Finance Budget'. Consistent with table 8.1, a 'Produces' relationship is represented by a square and a uses relationship is represented by a circle.

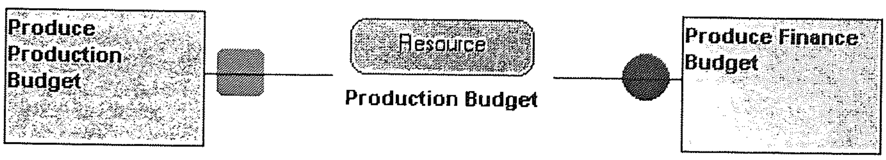

Figure 8.8. CBOSS Uses/Produces relationships

As mentioned previously, the 'Uses' and 'Produces' nodes rest upon 'Dependency links'. These are outlined by a single black line. This is also displayed in the figure above. Here two dependency link lines are portrayed. The first links the activity

'Produce Production budget' to the resource 'Production Budget'. The second links 'Production Budget' to the 'Produce Finance Budget' activity.

8.6. THE CO-ORDINATION MODEL

As presented in figure 7.2, the 'Co-ordination Model' comprises three co-ordination views. These are the 'Dependency View', 'Interdependency View' and 'Constraint View'. Information produced by these views are modelled by the CBOSS system. These views that makeup the 'Co-ordination Model', shall be considered in more detail below.

8.6.1. The Dependency View

The dependency analysis panel allows the tactical level manager to view dependency occurrences in a tactical level process. Consistent with sections 6.7 and 7.7.6.1, dependencies are detailed as 'Fit, Flow and Sharing'. An example of this, relating to the 'Assess Corporate Finances' example presented in section 7.6. can be seen in figure 8.9 below.

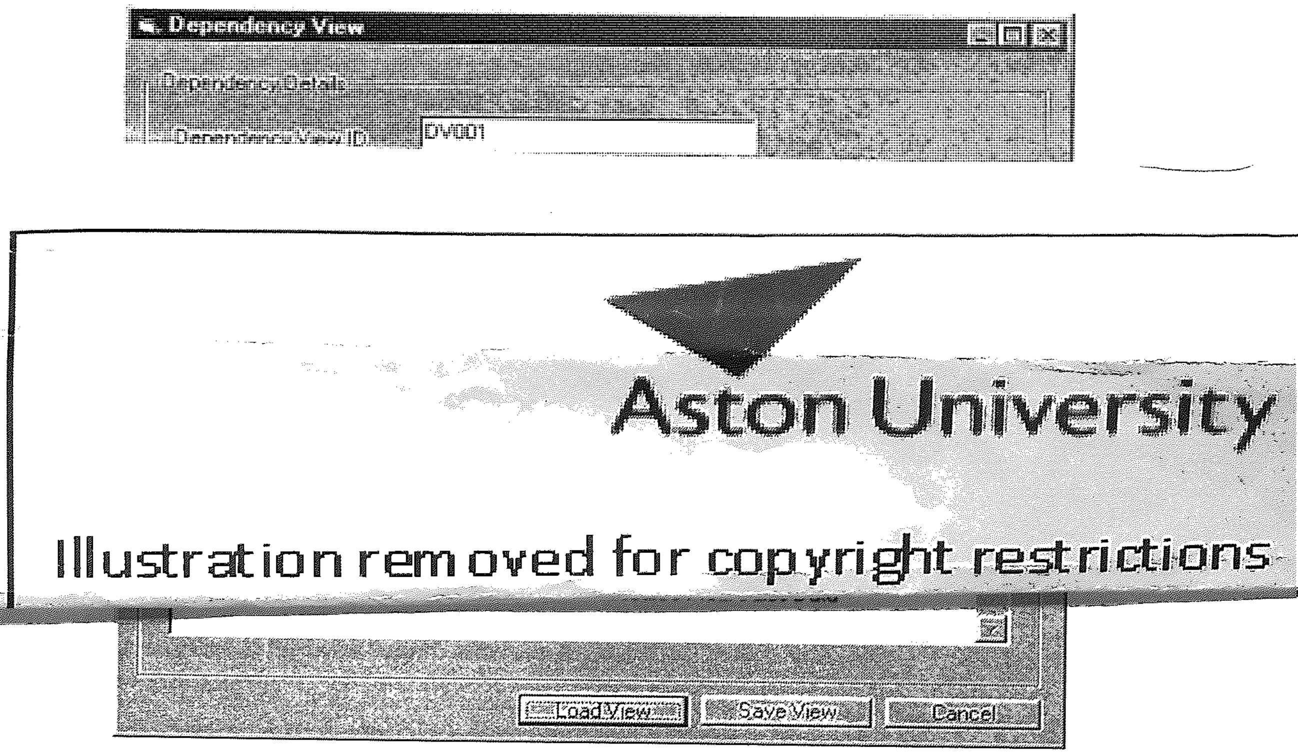

Figure 8.9. The Dependency Analysis View

Specifically, the dependency view delineates two principle groups of fields. These are 'Dependency Details', which denotes the 'Dependency ID' and 'Dependency Name'. The second group highlights the 'Dependency Analysis', which details the 'Dependency Type' and the 'Uses/Produces' relationship an activity has with a resource. In drawing parallels with Zlotkin (1995), all dependency types at their atomic level are constructed using one resource, this was demonstrated in figure 6.7. The dependency view shown below, allows the user to quickly search for individual dependency types by selecting resources. For example, the figure shows that the field 'Dependency Quick Find' is populated with the resource value 'Prod/Sales Data'. From this, the system calculates all the dependency types that exist, in this case 'Fit' and 'Flow'. Below this, all activities and their relationships with this resource are listed in the 'Uses/Produces Relationships' group.

8.6.2. The Interdependency View

The 'Interdependency View' shown in figure 8.10 below, analyses the current process and returns interdependency information based upon the data captured in the four CBOM object models discussed in sections 7.7.2, 7.7.3, 7.7.4 and 7.7.5.

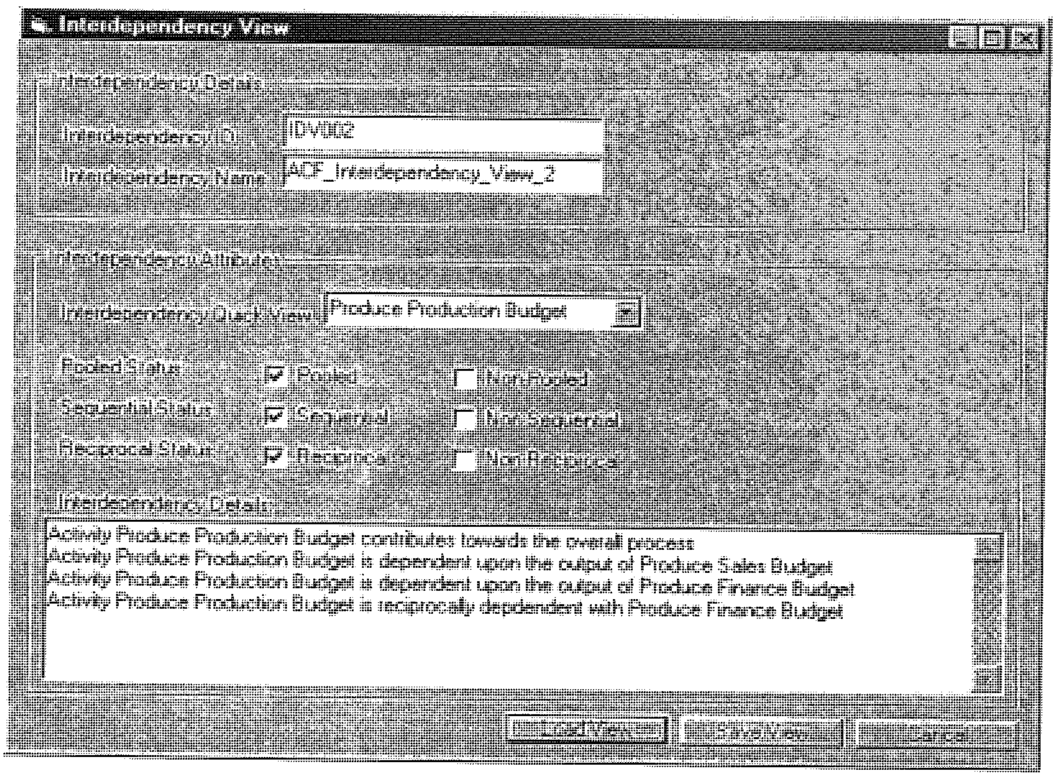

Figure 8.10. The Interdependency Analysis View

Discussed in section 7.7.6.2, the 'Interdependency View', consistent with Thompson (1967) (see figure 6.9), highlights three different interdependency types. These are, 'Pooled', 'Sequential' and 'Reciprocal'. The 'Interdependency View' shown in figure 8.10 above, is partitioned into three principle sections. Firstly, the 'Interdependency Details', denotes the 'Interdependency ID' and the 'Interdependency Name'. Secondly, the 'Interdependency Attributes' denote the interdependency type, whilst highlighting in the 'Interdependency Quick View' the current activity that is currently under review. Finally, the 'Interdependency Details', provides more information as to the activity's interdependency status. In the figure above, the current activity under review: 'Produce Production Budget' is dependent upon the output of the activities 'Produce Sales Budget' and 'Produce Finance Budget'. It also states that the 'Produce Production Budget', is reciprocally dependent with the activity 'Produce Finance Budget'.

8.6.3. The Constraint View

The 'Constraint Analysis' form, shown in figure 8.11 below, analyses all co-ordination processes and reports back to the tactical manager, any anomalies in the process regarding *Accessibility*, *Usability* and *Pre-requisite* constraints amongst activities. As updates are made to activities that belong to the process over a period of time, the constraints detected will undoubtedly change. Therefore, changes to the constraint types can be checked at any time by a tactical level manager, by using the functionality provided by this form. Detecting changes to the process and the low level data with which the CBOSS application uses, is the responsibility of the 'CBOSS Listener'. This is considered in more detail in the next section.

The 'Constraint Attributes' group, highlights issues relating to 'Accessibility', 'Usability' and 'Pre-requisite' constraints. For example, the activity 'Produce Sales Budget' uses the resource 'Sales Data', however, in this example the resource is not accessible. The activity 'Produce Production Budget' uses the resource 'Production Data' yet, this resource is not ready for use by the activity, thereby denoting a usability constraint. Finally, the CBOSS system also notifies the user that the activity 'Produce Corporate Budget' cannot execute until the resource 'Prod/Sales Data' has been produced. In any event, such notification allows the tactical manager to enquire as to the cause of these constraints in the process.

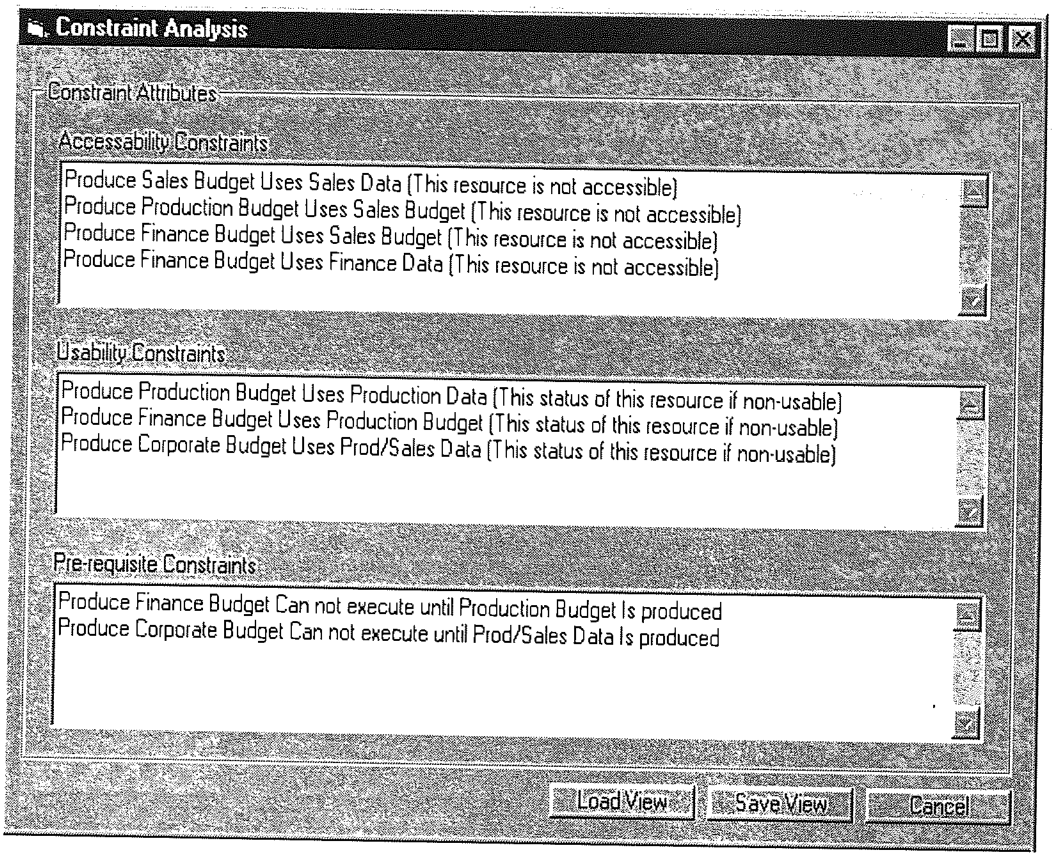

Figure 8.11. The Constraint Analysis View

8.7. CBOSS LISTENER

Once a tactical level process has been developed, the system needs to make the process 'live'. This allows the tactical level manager to keep track of real time changes throughout the process. This is achieved by using the CBOSS listener. The basic idea behind the listener, is to alert the user to changes in the process. For example, if organisation A was part of a virtual collaboration, and organisation B changed its despatch date, hence meaning that organisation A could not execute its assigned activity, then the listener, would highlight to the user that there was a problem with a particular activity. This would then allow the user, to investigate this further. In short, the listener tracks all low level data changes to the CBOM. Furthermore, in the case where anomalies such as 'Interdependencies' or 'Constraints' occur, CBOSS turns the activity red. This is highlighted in figure 8.12 below.

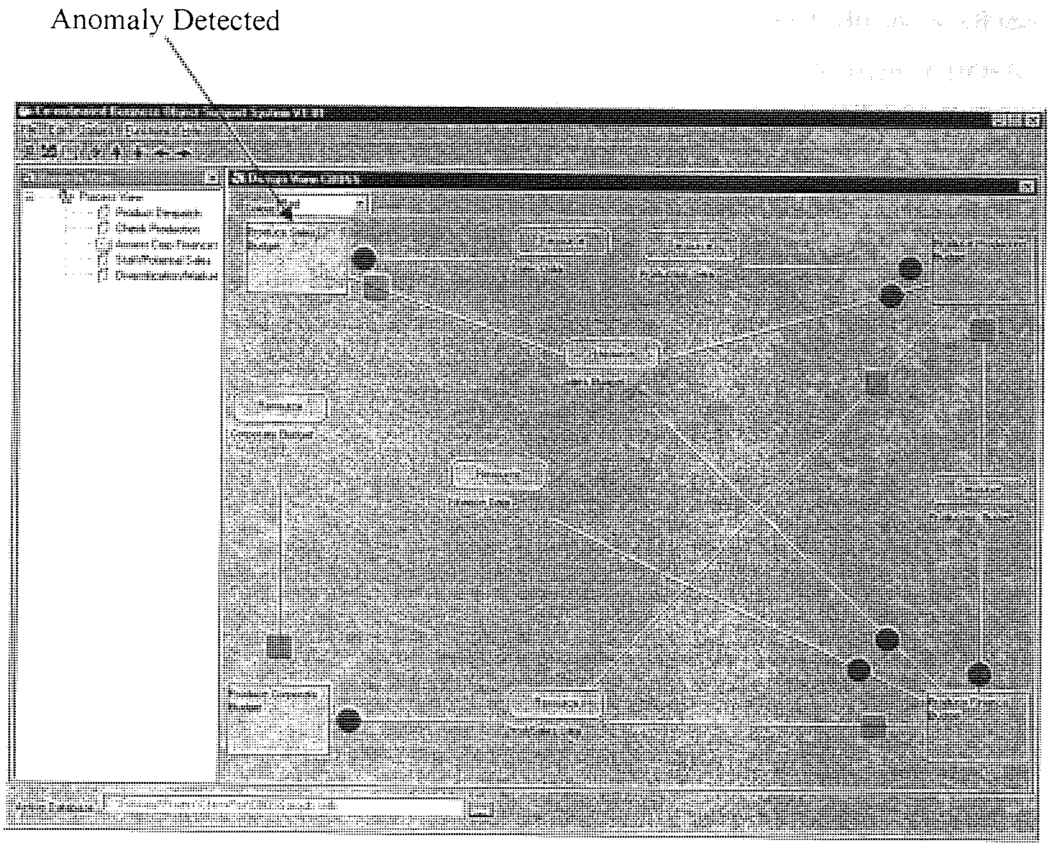

Figure 8.12. The CBOSS Listener

In this case the anomaly rests with the activity 'Produce Sales Budget'. Upon viewing this anomaly, a tactical level manager can reconstruct the process by allocating different tasks to different parties, or simply enquire as to the cause of the anomaly.

8.8. CONCLUSIONS

This chapter has presented an examination of the Co-ordinated Business Object Support System (CBOSS). The CBOSS system, as mentioned earlier supports an implementation of the CBOM. Furthermore, this chapter has targeted important areas of the CBOSS's functionality. In summary, it has presented the user interface and its modelling capabilities. Furthermore, via the use of examples it has demonstrated how the CBOSS system can be used to support tactical level decision making.

Having outlined the principle dimensions of the Co-ordinated Business Object Model (CBOM) and the Co-ordinated Business Object Support System (CBOSS), section C will conclude by detailing the results of three test cases that have been developed throughout the course of this study. The aim of these test studies are to validate the CBOM as an approach for supporting tactical level decision making. These test cases are considered in the next chapter.

CHAPTER 9

CBOM Testing And Evaluation

9.1. INTRODUCTION

So far, section C has considered the component parts of the Co-ordinated Business Object Model (CBOM) and the Co-ordinated Business Object Support System (CBOSS), as a demonstrator application for supporting the CBOM implementation. However, in order to validate the research presented so far, it is necessary to develop an environment that will rigorously test the ability of the CBOM to accurately represent tactical management processes. Such validation is consistent with objective seven, which outlined a need to test the theory development of this study; "*To test the modelling framework suggested in objective 5, in order to derive results as to the efficacy of the framework for supporting tactical management decision making*". By applying the CBOM to a series of testing scenarios, this chapter will aim to provide evidence as to whether or not the CBOM supports tactical management decision making. Specifically, this will be achieved by developing a series of three verification tests. These tests will consider three tactical level processes taken from two organisations.

9.2. EVALUATING THE CBOM

When evaluating an information system, it is not always necessary to establish if something is true or false. Indeed, the position presented by Walsham (1993), suggests that the nature of knowledge can only be judged more or less 'useful', rather than 'true' or 'false'. This becomes apparent when considering the knowledge to be gained from the CBOM, as its main function is to represent tactical level processes that are constantly changing. When analysing Walsham's view however, it is clear that a distinction is made between two fundamental research categories. Namely, 'Scientific' and 'Interpretivist' (Galliers 1985, 1992; Galliers & Land 1987). In distinguishing between these two areas, it can be observed that the scientific approach, in making assumptions that phenomena can be observed objectively and rigorously, takes the view that research is legitimated through repeatability as well as reductionism and refutability (Checkland 1981). The Interpretivist approach on the other hand, suggests that the methods of natural science to be more appropriate where human beings are concerned, as situations may be interpreted in different ways. Moreover, the Soft Systems Methodology (SSM) (Checkland, 1981; Checkland and Scholes, 1990), widely recognised as an information systems evaluation method owing to its diagnosis capabilities, supports the view suggested by Walsham (1993), by underlining the inappropriateness of scientific and engineering methods for understanding changes in organisations.

The method that has been adopted as part of the testing process outlined in this chapter, falls significantly into the interpretivist domain. From this perspective, two organisations have been used as the primary testing laboratory (Braa & Vidgen, 1995; 1996) for testing the validity of the CBOM. This has allowed technical artefacts relating to the academic disciplines, to be studied in light of their application upon the organisations that were used as part of the testing domains. In order to test the robustness of the CBOM, three test scenarios that more closely resemble the 'Case Study' method of testing have been explored. The case study is especially suitable for this testing approach, as it is relatively flexible in so far as it can be used in both scientific (Galliers, 1992) and interpretivist (Iivari, 1991) forms of research testing. Although case studies allow for the capturing and analysing of many variables in detail, it has been noted that in some circumstances, a lack of control over those variables (Galliers, 1992) can occur. The approach adopted in this study, arguably details a more softer approach to the case study. In short, it would be more accurate from this point of view to categorise the tests as 'verification tests', as they are simply verifying the validity of the CBOM to support tactical level decision making, rather than acting as a basis for directing the theory development of this study.

Deriving results from the testing phase of this study has been relatively subjective (Cronk & Fitzgerald, 1997; 1999), and hence recognised a shift from what can only be measured numerically to a more hermeneutic approach (Guba and Lincoln (1989). Such an approach has guided the evaluation process of this chapter. The evaluation process adopted throughout the course of this testing period, has therefore recognised an interpretivist approach to evaluation (Kanellis, Lycett and Paul, 1999).

9.2.1. Organisations Involved In This Testing Period

As mentioned previously, two organisations were used in the testing of the CBOM. These are detailed below.

9.2.1.1. Organisation 1

The first organisation is a software house that provides internet based software solutions to various clients. Although the organisation reflects a hierarchical structure, its operations are project driven. As a consequence of this, groups of specialised workers form temporary project teams that last generally for the length of the project. From this point of view, this organisation draws parallels with the 'Project Group' model suggested in figure 2.3. In addition, this company is only two years old and is currently expanding. In order to cope with this new expansion, the company needs to hire more staff. Consequently, this process is examined in the first verification test. Moreover, as part of a diversification strategy, the company is involved in a virtual collaboration with three other organisations, in an attempt to provide vehicle pricing software solutions to the automotive industry. This process, 'Diversification Into A New Market', will be analysed as part of the second verification test. A maximum of ten visits to 'Organisation 1' were carried out over this testing period. This was a sufficient enough period of time with which to carry out both verification tests.

9.2.1.2. Organisation 2

The second organisation involved in this testing period operates in the engineering industry, where the main focus of its activities is in the supplying of parts to the automotive industry. This organisation's structure is relatively vertical, with distinct stratified management layers. However, from closer observation, the production and sales departments are supported by the maintenance, logistics and IT departments. From this perspective, it could be argued that this organisation more closely reflects a matrix structure (McCarthy, 2002), similar to that presented in figure 2.2. As part of the testing procedure, one process entitled 'Despatch Product' has been examined as part of the third verification test. Eight visits were made to organisation 2, throughout this testing period.

9.2.2. Data Collection Methods

Chapter seven highlighted the structure of the CBOM data model. It described the foundations of the CBOM as being based upon ideas considered in chapter five regarding elements that concerned data modelling, and chapter six which explored the contribution made by co-ordination theory for modelling organisational processes. Subsequently, the formulation of the CBOM emerged as a data modelling approach for supporting tactical management decision making. In short, the CBOM defined the template for the type of data that would be required. From this perspective, the data that was collected as part of the field research, was largely driven by the data requirements defined in the CBOM.

As figure 1.4 clarified, the theory development of this study was characterised through a series of 'Understanding' 'Prediction' and 'Change'. The data that has been collected throughout the testing phase, has been used to test key areas of the CBOM. The method of collecting this data however, has been conducted over a period of time that was sufficient enough to allow for change to be administered to the development, and theoretical foundations of the CBOM if need be. Specifically, the data collection methods have involved a series of interviews and consultations with tactical level managers in both organisations. Finally, throughout the testing phase, a series of observations regarding each test has also been recorded.

9.2.3. The Test Areas

When developing the testing criteria, it has been necessary to test not only the salient parts of the system, but to also ensure that the testing validated the theory development of this thesis. However, before any form of testing could be started, it was necessary to develop criteria that could be used as a suitable yardstick for assessing the CBOM. Therefore, recognising the areas to be tested, allowed for a more thorough and valid testing procedure. From this perspective, it was important to underline that the overarching aim of this test period, was to rigorously test the validity of the CBOM. Therefore, this testing phase was centred around testing the CBOM, rather than testing other information systems issues relating to, for example, systems performance testing, user testing or 'Human Computer Interaction' (HCI). As the approach adopted in this testing phase was relatively hermeneutic, it seemed appropriate that the first test should support an exploratory interrogation of the CBOM. Consequently, it focused upon testing the ability of the CBOM to model all areas of the tactical management process under review. The second test further extended this interpretivist approach, by detailing the CBOSS as an implementation of the CBOM model, for providing the user with tactical level information. The exact tests that were applied are considered in more detail below.

9.2.3.1. The First Test

The first test details the ability of the CBOM to model all areas of the tactical management process. In short, this test can be broadly described as;

> ***'Does the CBOM model all areas of the tactical management process for all three verification tests?'***

This test determines the ability of the four CBOM object models (*Process, Activity, Resource and Actor*), to model the current tactical level process that is under review. From this perspective, it is possible to define two possible areas to this test. Firstly, when the test process has been devised, and secondly when the process is 'live' or 'active'. In order to test these areas thoroughly, the first part of this test detailed the process before it was live. In doing so, a series of test data tables are presented in appendix C. These tables represent the data gathered throughout the initial field tests for all three verification tests. Where appropriate, these tables will be referred to throughout the course of this chapter. When the process was 'live', changes to the initial low level data presented in appendix C occurred. These changes were detailed in a series of observations taken on the field visits to both organisations one and two.

9.2.3.2. The Second Test

Throughout the testing period, the CBOSS was used to formulate tactical decision making. This led to the emergence of the second test. Specifically, this test detailed the ability of the CBOSS to adequately present information based upon the current tactical level process under review. Therefore, the following question adequately describes the aim of the second test;

> ***'Can the CBOSS system be used to provide co-ordinated tactical level information based upon the tactical level process under review?'***

As the CBOSS system is a prototype, it is necessary to underline that at this stage, it is in no way intended to replace the decision making tools that were currently being used by tactical management in both organisations one and two for the completion of their everyday tasks. In essence, it is more realistic to suggest that throughout the testing period, the CBOSS system sat along side other information systems used by the tactical manager. From this perspective, other systems were still available as a 'back up'.

9.2.4. The Testing Procedure

The testing procedure adopted in this thesis, sequentially tested all three processes, then analysed and produced summaries based upon the results that have been derived as a consequence of the testing. Figure 9.1 below delineates the scope of the testing procedure.

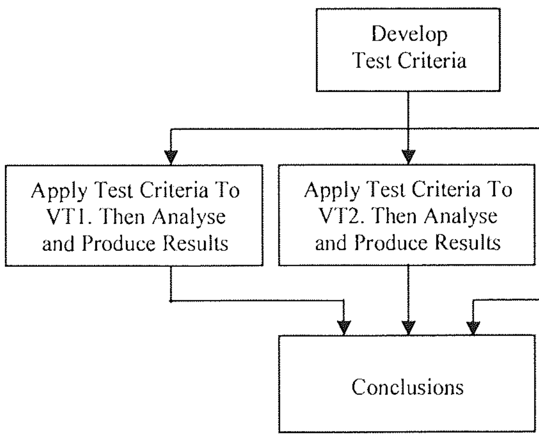

Figure 9.1. The testing procedure

The remainder of this chapter will consider the three verification tests and the resultant findings that have been derived from them.

9.3. VERIFICATION TEST 1 (STAFF/SALES POTENTIAL)

The first process to be tested, considered the task of hiring staff based upon potential sales data. The purpose of this process, is to enable the Personnel manager to estimate the number of staff to hire, based upon data gathered from the market research. Owing to a drive by strategic level management to increase revenue, it was decided that an increase in staff was required. Specifically, the field study captured three primary activities that were required for hiring these extra staff. These were, 'Analyse Market Research Data', 'Assess Sales Potential' and 'Hire Staff'. These activities were administered by the various heads of department that were responsible for their execution. In addition, contained within this process, four resources were identified. These were 'Data Sample From Public', 'Analysed Data From Public Sample', 'Potential Sales Data' and 'Amount of Extra Staff Required'. Figure 9.2 below describes a 'Process Model Level 1' of the 'Staff/Sales Potential' process.

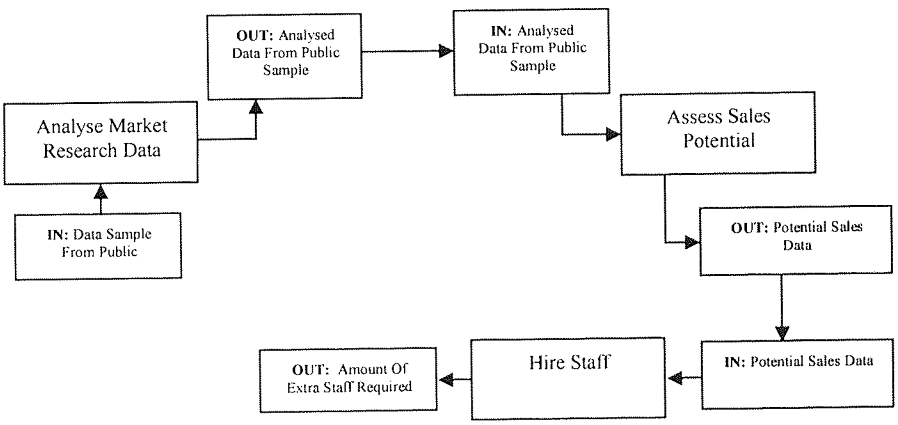

Figure 9.2. Process Model Level 1: Staff/Sales Potential

The organisation from which this process is taken, is relatively hierarchical in nature and therefore would be consistent with the model proposed by Anthony (1965) (see figure 2.1). However, although the process is carried out within a single organisation, each activity is the responsibility of three separate managers who reside at the tactical level within the organisation. From this perspective, the process can be viewed as being relatively horizontal in nature. These three individual activities will all have 'sub-activities' attached to them. Therefore, the achievement of these activities will rely upon the successful execution of these sub-activities. However, in modelling this process, these sub-activities will not be displayed in further diagrams as they are the responsibility of the individual organisations. In short, it is the composite tactical activities that represents these lower level activities that will be modelled. In order to analyse this process in further depth, figure 9.3 below provides a level two breakdown of the corresponding activities of the 'Staff/Sales Potential' process.

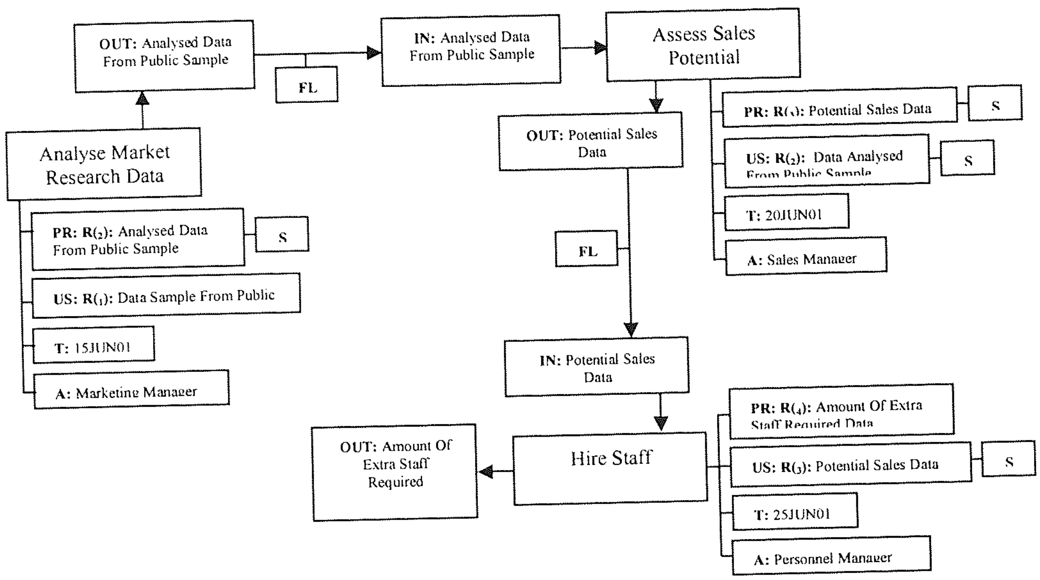

Figure 9.3. Process Model Level 2 : Staff/Sales Potential

In adopting the diagrammatic approach presented in chapter seven, it details two shared resources, namely; 'Analysed Data From Public Sample' and 'Potential Sales Data'. It also shows the relevant actor and time constraints that are attached to the three activities. In addition, figure 9.4 below, provides a 'Co-ordination View' of the 'Staff/Sales Potential' process.

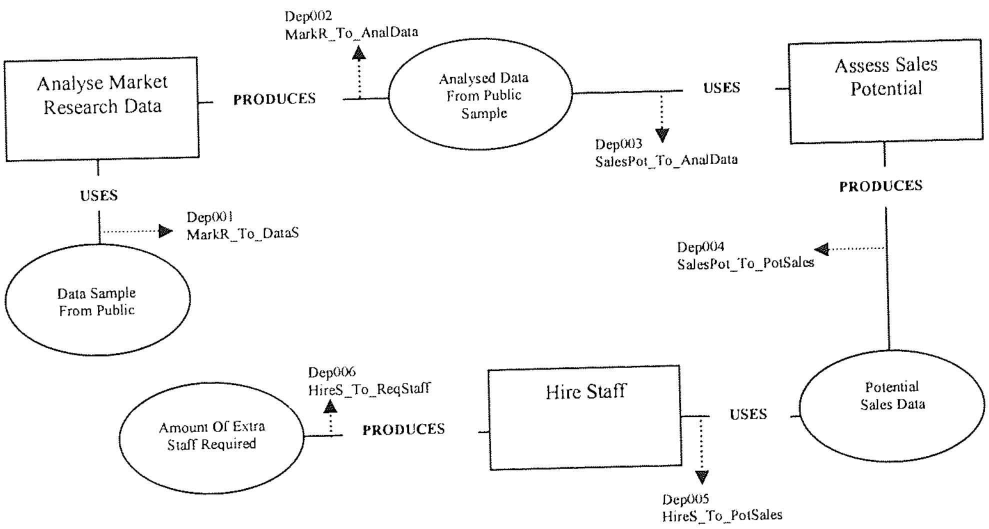

Figure 9.4. Co-ordination View: Staff/Sales Potential

Denoting dependency links and uses / produces relationships between activities and resources, figure 9.4 displays six dependency links in total. By studying the diagram, it is clear that dependency links 'Dep002' and 'Dep003' link the activities 'Analyse Market Research Data' and 'Assess Sales Potential' to the resource 'Analysed Data From Public Sample', thereby signifying a flow dependency. The same dependency type occurs between dependency links 'Dep004' and 'Dep005' which links the activities 'Assess Sales Potential' and 'Hire Staff' to the resource 'Potential Sales Data'. Further analyses of dependency occurrences relating to this process however, will be considered in more detail in the testing phase of this section in 9.3.2. The remainder of this section will detail the test results of the two verification tests outlined in 9.2.3.

9.3.1. Application Of The First Test

As detailed in section 9.2.3.1 above, this section will test the modelling ability of the CBOM in light of the tactical level process under review. From this perspective, appendix C details the test data tables for all three tests that are contained within this study. Specifically, appendix C.1 details the specific data that refers to the 'Staff/Sales Potential' process. It should be noted however, that these tables provide a snapshot of the process at a particular moment in time. From this perspective, the tables do not provide any evidence that the CBOM modelled all areas of the tactical level process when the process was active. Therefore, further testing necessary to validate this test, required the monitoring and observation of the process when it was deemed to be active or ongoing. As a consequence of this, the initial field visits regarding this process were monitored over a three month period from the 15th June 2001 to the 20th September 2001. Consequently, in order to assess if the CBOM can indeed model all areas of a changing tactical level process, the following observations were recorded.

Observation 1. Throughout the three month period, issues arose with the collection of data by the activity 'Analyse Market Research Data'. Consequently, the data was not made available to the activity 'Assess Sales Potential' until the 21st June. Taking this into account, the resource 'Analysed Data from Public Sample' was late by one day. Therefore, this resource was neither accessible nor usable. This however, had little effect upon the data modelling issues to which this test refers, as the 'Resource Accessible' and 'Resource Usable' flags were simply changed to the value 'NO'. Furthermore, in order to overcome this problem, it was agreed that the activity execution date for both the, 'Assess Sales Potential' and 'Hire Staff' activities should be changed to the '22nd and 27th June'.

Observation 2. The 'Hire Staff' activity, involved various operational level sub-activities, such as 'Placing Advert', 'Interviewing Employees' and 'Selecting Staff'. As these activities are operational level in nature, they did not require the focus of the tactical level manager, in this case the 'Personnel Manager'. It was issues that concern the overall process 'Hire Staff' with which the tactical level manager was primarily concerned. Consequently, as the 'Hire Staff' activity was executing, instructions from head office required a change in procedure. In so far as the hiring of staff was to be deferred until a later date. This raised interesting points regarding the way CBOM models not only this process but all tactical level processes. For instance, the CBOM was required to model an adjustment to the process that could not be conceived as being part of the 'Staff/Sales Potential' process. However, this intervention nonetheless had an effect upon the process. Therefore, in order to model this, two possible courses of action could be taken. Firstly, the CBOM could embellish the 'Staff/Sales Potential' process by adding an extra activity to the process. Namely, 'Wait for Head Office Approval'. Alternatively, instead of treating this intervention by head office as an activity, it could simply leave the process in its current state with the 'Hire Staff' activity highlighted red by the CBOSS listener (see section 8.7).

Depending upon the perspective, it possible to suggest that both approaches are correct. For example, it may be argued that the latter is the more appropriate approach, as the intervention was not part of the 'Staff/Sales Potential' process. Also, the argument may extend to suggest that the intervention by the head office, may not be classified as a tactical-type activity. In this case, the CBOM would fail to model such an intervention. The former approach on the other hand, would suggest that a new activity should be produced. Therefore, as the CBOSS functionality extends to being able to accomplish this, it may be argued that this is the only approach that can be taken by the user. The resulting effect of this would be to create a new process with a new specification.

9.3.2. Application Of The Second Test

As chapter eight clarified, the CBOSS system allows for the development of tactical level processes, as well as providing information based upon those processes, when they are implemented within an operational capacity (see section 8.7). However, in order to collect suitable evidence for this test, it was necessary to investigate the contribution made by the CBOSS, in support of the CBOM model for the 'Staff/Sales Potential' process. From an operational point of view, the CBOSS provides a graphical depiction of the tactical level process by allowing the user, in this case the tactical level manager, to model the 'Staff/Sales Potential' process, by using the development tools provided. An example of this can be seen in figure 9.5 below.

The diagram below provides the 'Personnel Manager' with a co-ordinated view of the process. The 'Uses / Produces' relationships are displayed along with activities and resources that are associated to this process.

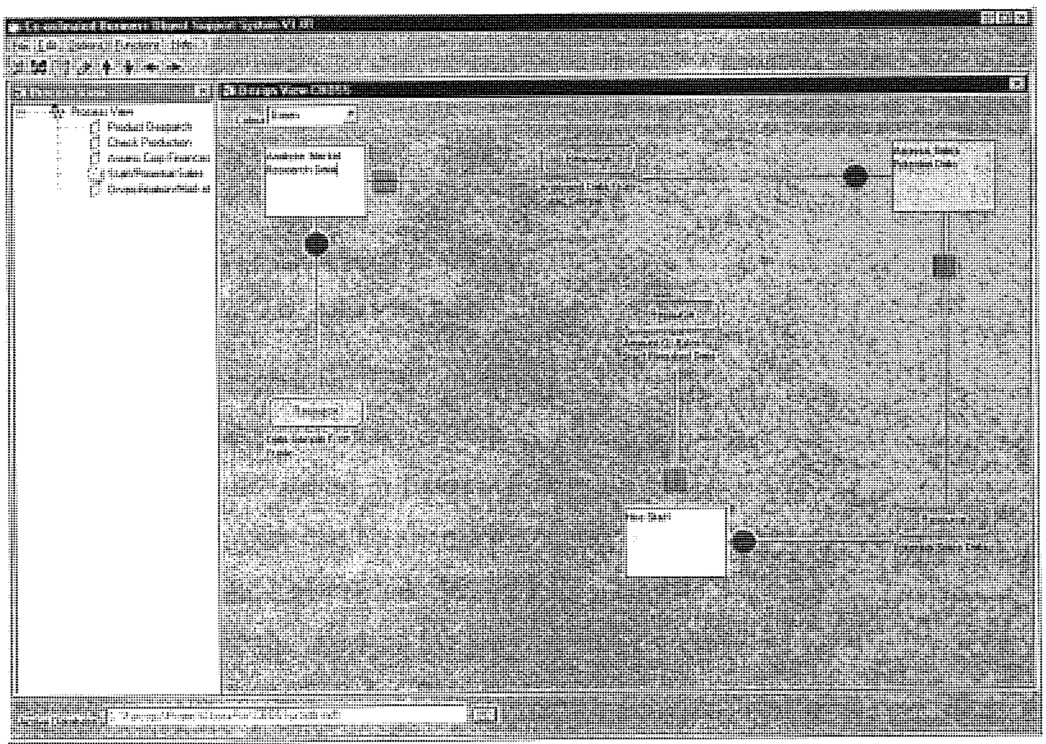

Figure 9.5. Staff/Sales Potential (CBOSS Diagram)

In addition, the co-ordination functions that have been added to the CBOSS system, allow the user to analyse 'Dependencies', Interdependencies' and 'Constraints' by making the appropriate selections from the CBOSS menu (see appendix B).

Throughout the testing period, the CBOSS system was used extensively to map the operational activities associated to the 'Staff/Sales Potential' process. When concerns regarding the activity 'Analyse Market Research Data' became apparent, CBOSS notified the user by highlighting the 'Analyse Market Research Data' activity with a warning. Figure 9.6 provides an example of this.

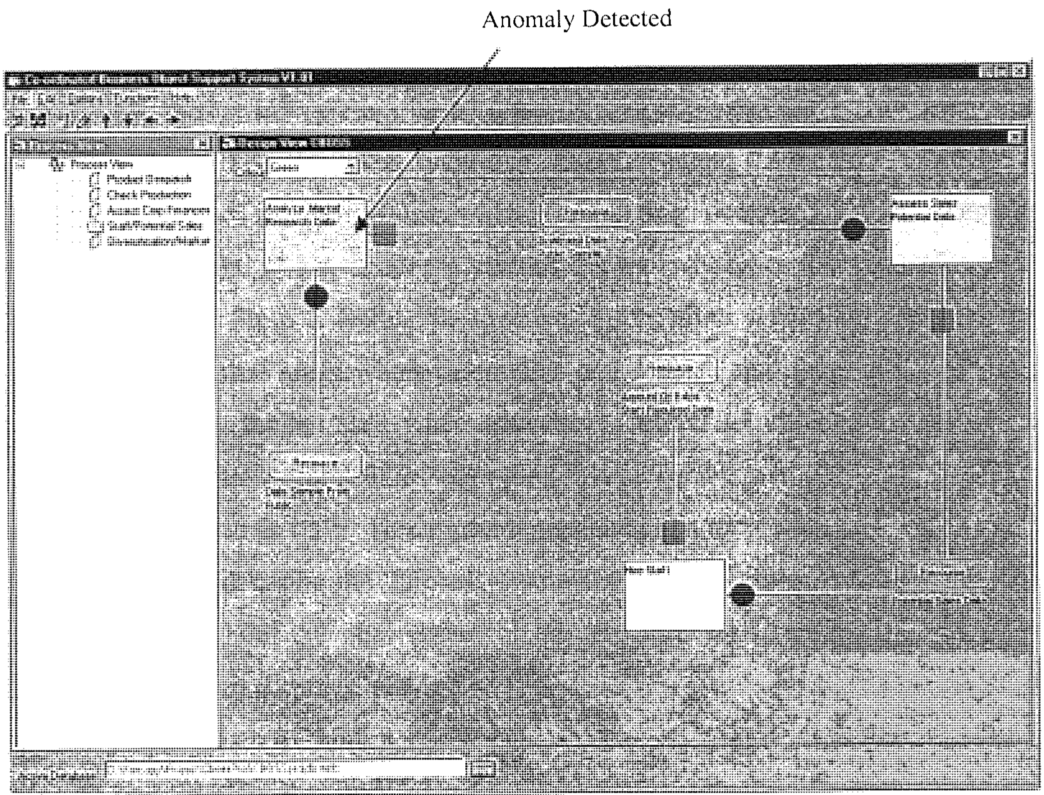

Figure 9.6. Anomaly Detected in Staff/Sales Potential

Consistent with observation 1, this activity could not execute as there were issues regarding the collection of the initial market research data. Consequently, the co-ordination model delineated in figure 9.6 above, brought this problem to light and hence notified the user that an alternative course of action, regarding the execution of this activity would be needed. In order to overcome this problem, the 'Activity Execution Date' for the 'Analyse Market Research Data' activity was adjusted to the 22^{nd} June. The 'Assess Sales Potential' activity was changed to the 27^{th} June. Screen displays of these activities are presented in appendix D under sections D.1.1 and D.1.2. The 'Activity Execution Date' for the third activity, 'Hire Staff', shown in D.1.3 was not adjusted as it was agreed that this activity had sufficient time with which to execute. The updated CBOSS screen display for the resource 'Analysed Data From Public Sample' is shown in appendix D section D.1.4. Under test conditions, the 'Resource Overview' was consulted mainly for the 'Resource ID', 'Resource Name', 'Availability Date', 'Resource Accessible' and 'Resource Usable' fields. Information regarding the 'Shareability', 'Usability', 'Tangiability', 'Position' and 'Timespan', although useful for defining the makeup of the resource, were, by themselves rarely consulted. Other resources that refer to this process are shown in appendix D.1.5 to D.1.7.

In addition, further information based upon the co-ordination constraints discussed in both chapters six and seven regarding 'Dependencies', 'Interdependencies' and 'Constraints', also proved useful when formulating decisions. For example, appendix D.1.8 and D.1.9, detail the various dependency types that naturally occur in this process. From a testing perspective, information regarding dependencies highlighted potential areas where anomalies could occur and therefore acted as a point of reference for the user, when establishing which activities to monitor more carefully. The 'Interdependencies' regarding this process, are displayed in appendix D.1.10, D.1.11 and D.1.12. Three interdependencies in total where detected by the CBOSS system, detailing the 'Pooled', 'Sequential' and 'Reciprocal' values for each activity. Throughout the observation of this process, the interdependency values listed in the screen displays, seemed to be used more as supporting information by the user, in so much as the 'Interdependency View' was rarely consulted. The user seemed only to view the 'Interdependency View' for curiosity and general interest. The 'Constraint View' presented in D.1.13 on the other hand, seemed to be of more benefit to the user, as it provided critical up to date information regarding the current state of the activities in the process

In drawing conclusions from this test, it can be argued that the CBOSS provided relevant support to the user. In doing so, it provided modelling and operational support in terms of changes that occurred to the process. However, as mentioned in observation two in section 9.3.1, interventions, such as those from head office, that are not considered activities, need to be modelled as separate activities. Alternatively, changes need to be made to the process in order to reflect these changes in the outside environment. From this perspective, it is possible to argue that such an intervention from head office, is in itself, part of the process and that the original process was inaccurately specified. Therefore, the original model of the process, was originally inaccurate and subsequently was in need of updating.

9.4. VERIFICATION TEST 2 (DIVERSIFICATION INTO A NEW MARKET)

The second verification test, similar to the first test, was also carried out in 'Organisation 1'. As part of a drive to expand its operations, 'Organisation 1' undertook a virtual collaboration with two other organisations. These organisations, for the purposes of this test these were named 'Organisation 1(A)' and 'Organisation 1(B)'. The collaboration was centred around extending the current outsourcing operations into the automotive industry. Organisation 1(A), took responsibility for the activity 'Assess Market Suitability'. This organisation specialises in collating market data for organisations. Such data is typically used for reshaping business strategy. This organisation ceased to be part of the collaboration after it had produced its initial findings regarding market suitability. Organisation 1(B) was responsible for the activity 'Assess Sales Potential', this organisation was already established in the automobile industry as a supplier of up to date automotive car price data for the UK market. It did not however, have the software development capabilities of 'Organisation 1', and was therefore forced to partake in a collaboration with another organisation. Before any firm commitment was made towards furthering the partnership, a series of activities, with the intent of uncovering information as to the viability of the venture was conducted. These activities formed part of the composite process 'Diversification Into A New Market'. An example of this is shown in figure 9.7 below.

Although both organisations are in the UK, the business practices were conducted over the internet, with the possible formulation of a team to oversee the efficient running of the collaboration. As this new organisation had no new or specially adapted headquarters in the form of physical premises, this outlined the basis for a virtual organisation. Evidence in support of this can be derived from the view posited by Burn, Marshall and Wild (1999) (see section 2.4.3). In doing so, this particular type of virtual organisation would map more closely to a 'Value-Alliance' model, whereby one organisation co-ordinates various participants in the collaboration, rather than a 'Market Alliance' model, where the virtual collaboration is driven by the market.

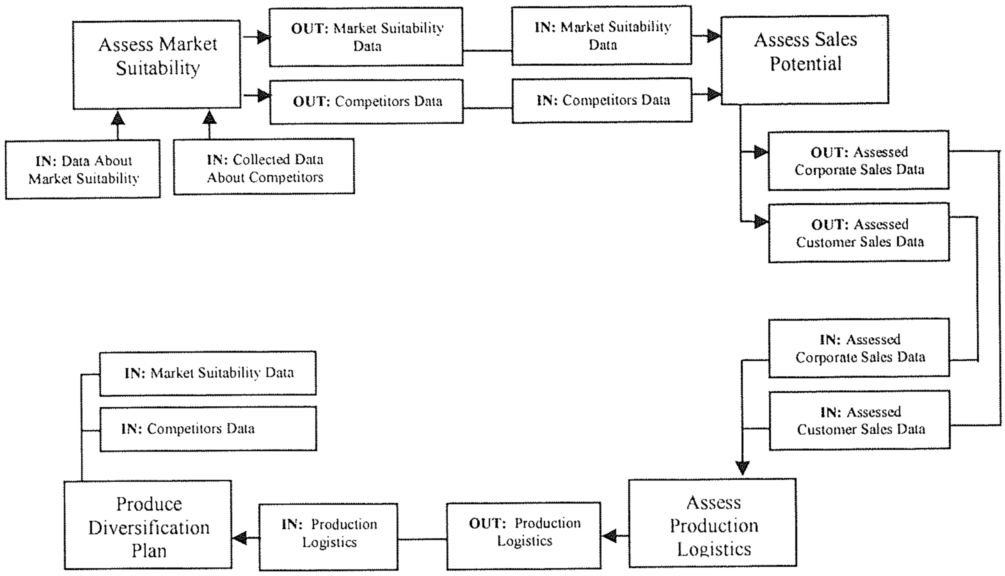

Figure 9.7. Process Model Level 1: Diversification Into A New Market

As mentioned previously 'Organisation 1(A)', was responsible for the activity 'Assess Market Suitability' whilst 'Organisation 1(B)' took responsibility for the activity 'Assess Sales Potential'. Therefore, 'Organisation 1' was responsible for the activities 'Produce Production Logistics' and 'Produce Diversification Plan'. Figure 9.7 above, details the existence of eight resources.

The existence of a fit dependency can be highlighted at the 'Assess Sales Potential' activity. For example, execution of the 'Assess Production Logistics' activity, is dependent upon the 'Assess Sales Potential' producing the 'Assessed Corporate Sales Data' and 'Assessed Customer Sales Data' resources at the same time. Further information regarding the 'Uses / Produces' state of the resources, along with the various dependency types and resource flows are presented in the 'Process Model Level 2: Diversification Into A New Market' in figure 9.8 below. This diagram also outlines the time and actor details. Here four managers, that can be considered to operate at the tactical level are assigned as the managers of one activity each. In this test, there were no multiple actor assignments. Therefore, one manager was in control of one activity.

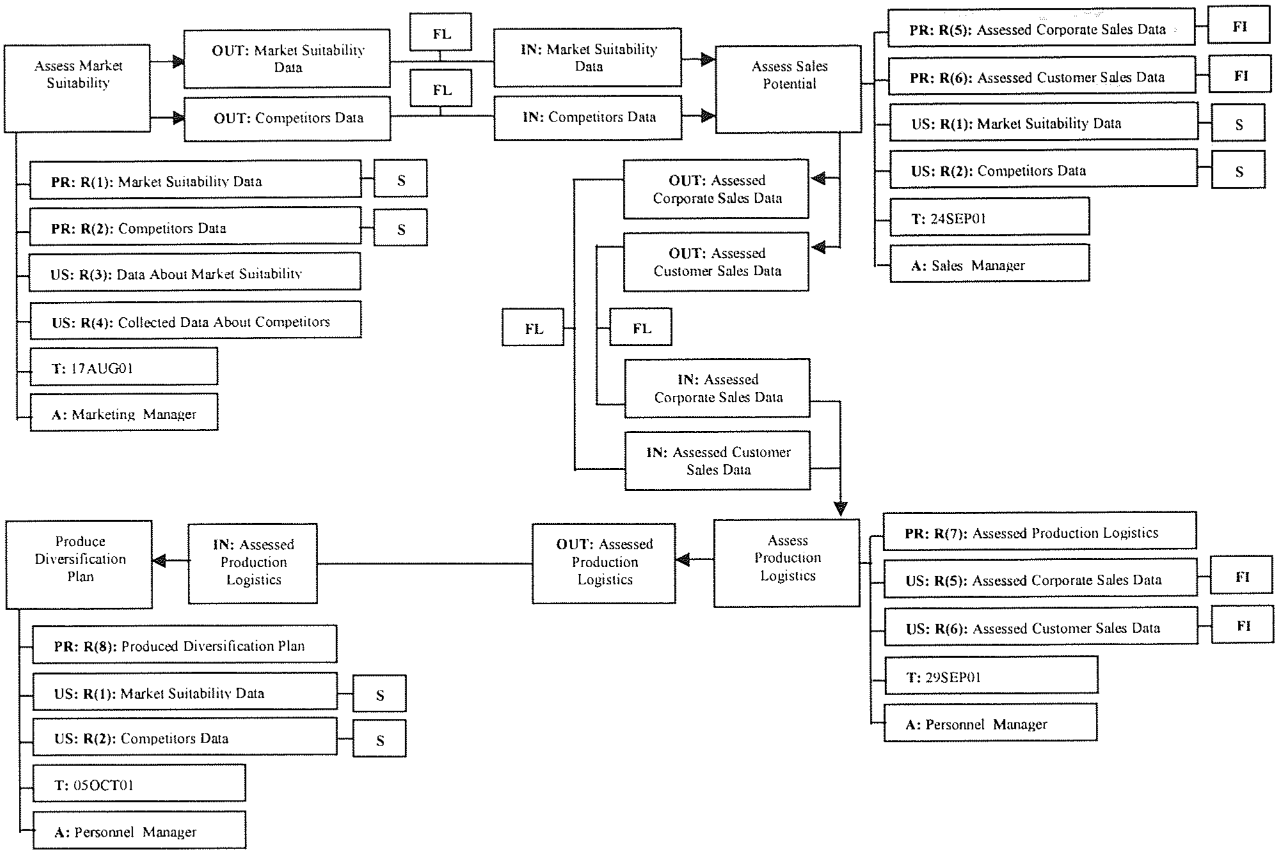

Figure 9.8. Process Model Level 2 : Diversification Into A New Market

The overall cycle time for this process is a little over one month. However, similar to the first process considered in section 9.3, the above diagram provides a snapshot of the process at a point in time. This diagram also displays two resources that are shared. These are, 'Market Suitability Data' and 'Competitors Data'. This type of diagram however, does not detail the dependency links that exist between the various processes. This however is represented in figure 9.9 below, which outlines a 'Co-ordination View' of the 'Diversification Into A New Market' process. Specifically, this diagram highlights the activities, resources and the corresponding dependency links that connect activities to resources. Fourteen different dependency links exist between activities and resources in this process, which together form to make various dependency types. These however, shall be considered in more detail throughout the remainder of this section. The dependency links shown in figure 9.9 have associated names. These names are modelled as 'Dependency_Link_Name' in CBOM, and were highlighted in further detail in section 7.7.3.

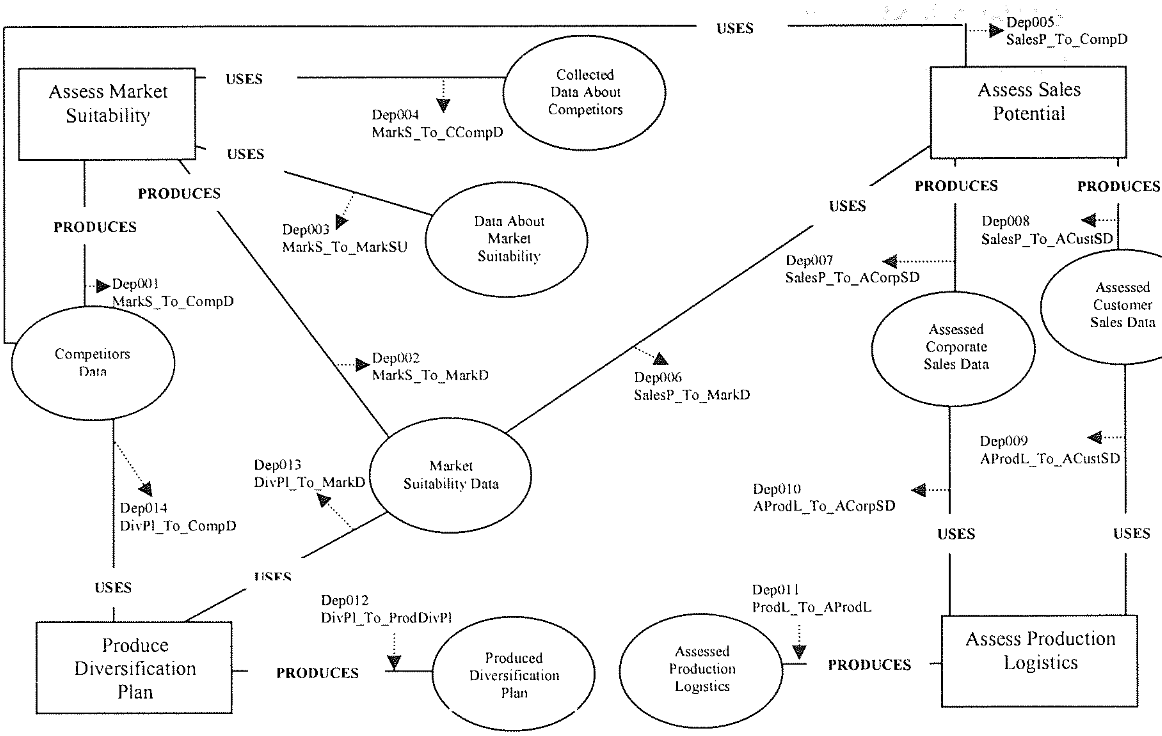

Figure 9.9. Co-ordination View : Diversification Into A New Market

The testing for this process will be presented throughout the remainder of this section.

9.4.1. Application Of The First Test

The data gathered throughout the testing period for this process is shown in appendix C.2. As noted in 9.3.1 above, the data contained in appendix C, was collected at the start of the process. In short, it is a snapshot of the state of the process and its associated activities, resources, actors and dependency links. This verification test, similar to 9.3.1, consisted of a series of field visits to Organisation one. This test however, overlapped 'verification test 1', and it was therefore possible to collect data for both tests at the same time. The precise testing period for this process was between the 17th August and the 24th December 2001. The relevant observations that were made throughout this period are detailed below.

Observation 1. Before the eventual process was modelled on the CBOSS, the user expressed a wish to be part of the design process. This allowed the user to experiment with different design patterns. Furthermore, it also gave the user the opportunity to analyse dependency and interdependency patterns in the different process designs. This uncovered a significant finding relating to the dependency and interdependency views. As noted in the first verification test, the user did not rely upon these views when the process was 'active'. Therefore, it seems that at this early stage, the evidence suggests that these views are more useful when designing the process, than being used as a tool for analysing the 'live' process. It is however, important to note that this observation has been derived from one test, and in order to further validate this assumption, much more 'user-type' testing would need to be completed.

Observation 2. Organisation 1(A), can be regarded as an organisation that is outside the virtual partnership that exists between 'Organisation 1(B)' and 'Organisation 1'. During the execution of the 'Assess Market Suitability' activity, some problems with the data production where encountered. This however, did not effect the completion date of this activity

Observation 3. Throughout the execution of the activity 'Assess Sales Potential', there were issues with the initial data collection owing to a delay in starting the activity. This was not owing to any recognisable problems with the activity 'Market Suitability Data', but simply the actors who were to carry out the 'Assess Sales Potential' activity, were unavailable to start on the specified activity execution date. Consequently, the resources 'Assessed Corporate Sales Data' and 'Assessed Customer Sales Data' could not be produced on time. This therefore impacted upon the execution of the activity 'Assess Production Budget'. In order to compensate for this, it was decided to extend the activity execution date of this activity.

Observation 4. Throughout the course of the project, it was necessary for 'Organisation 1(A)' to be consulted, when organisations' 1 and 1(B) were interpreting the data produced by the 'Competitors Data' and 'Market Suitability Data'. This therefore, detailed minor iterations in the process.

9.4.2. Application Of The Second Test

In order to assess the ability of the CBOSS to provide tactical level information to the user, it was necessary to construct the process, using the tools attached to the CBOSS. Figure 9.10 below, denotes the relevant activities and resources that comprise the process 'Diversification Into A New Market'. Specifically, the activities are shaded in order to outline which organisation owns which activity. The names of the organisations are attached to the activities in order to explicitly outline this. In this case, the CBOSS uses this facility to highlight the existence of a virtual organisation. Consistent with the text outlined in the introduction to section 9.4 above, there are three recognisable organisations. In the diagram below, 'Organisation 1' is responsible for the activities 'Assess Production Logistics' and 'Produce Diversification Plan', 'Organisation 1(A)' is responsible for the activity 'Assess Market Suitability' and 'Organisation 1(B)' for 'Assess Sales Potential'. The figure below, consistent with figure 9.7, 9.8 and 9.9, shows the relevant activities and resources that belong to this process. Furthermore, it also displays the uses and produces relationships that exist between the resources and activities.

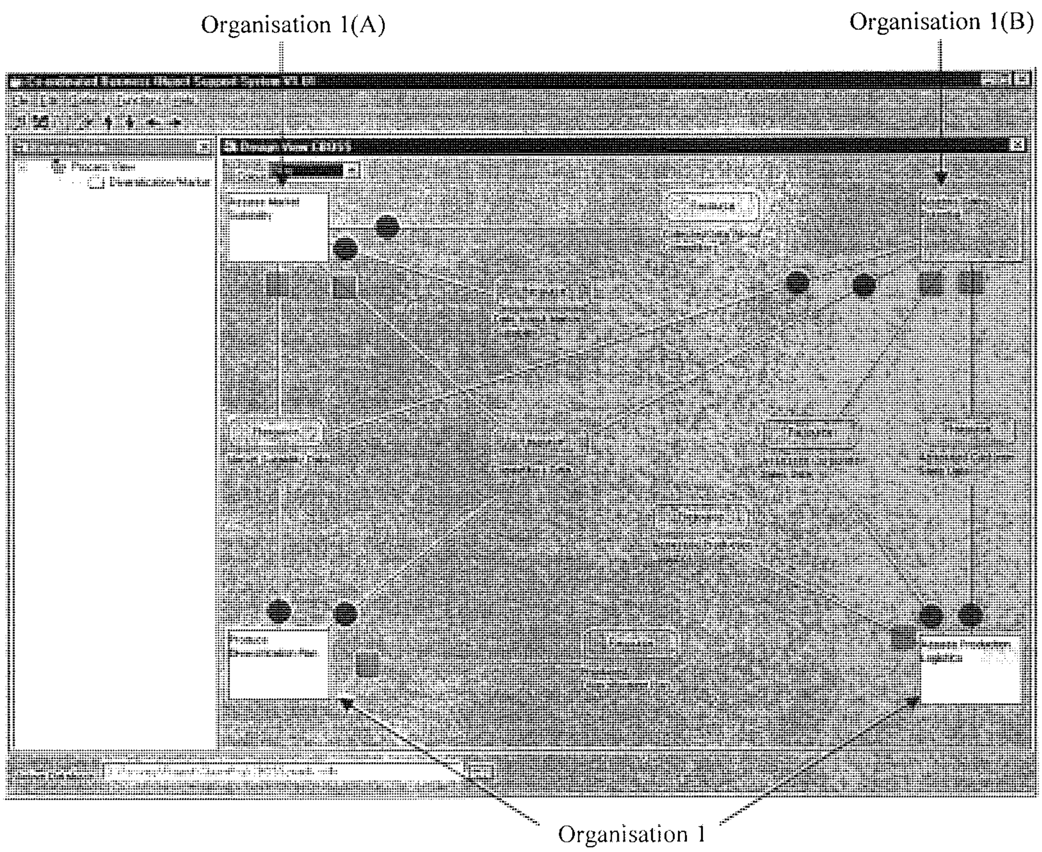

Figure 9.10. Diversification Into A New Market (CBOSS Diagram)

Observation three, detailed an issue with the activity 'Assess Sales Potential'. This was attributed to problems with the initial data collection. The CBOSS denoted this by highlighting the activity with a warning. This is shown in figure 9.11 below.

Anomaly Detected

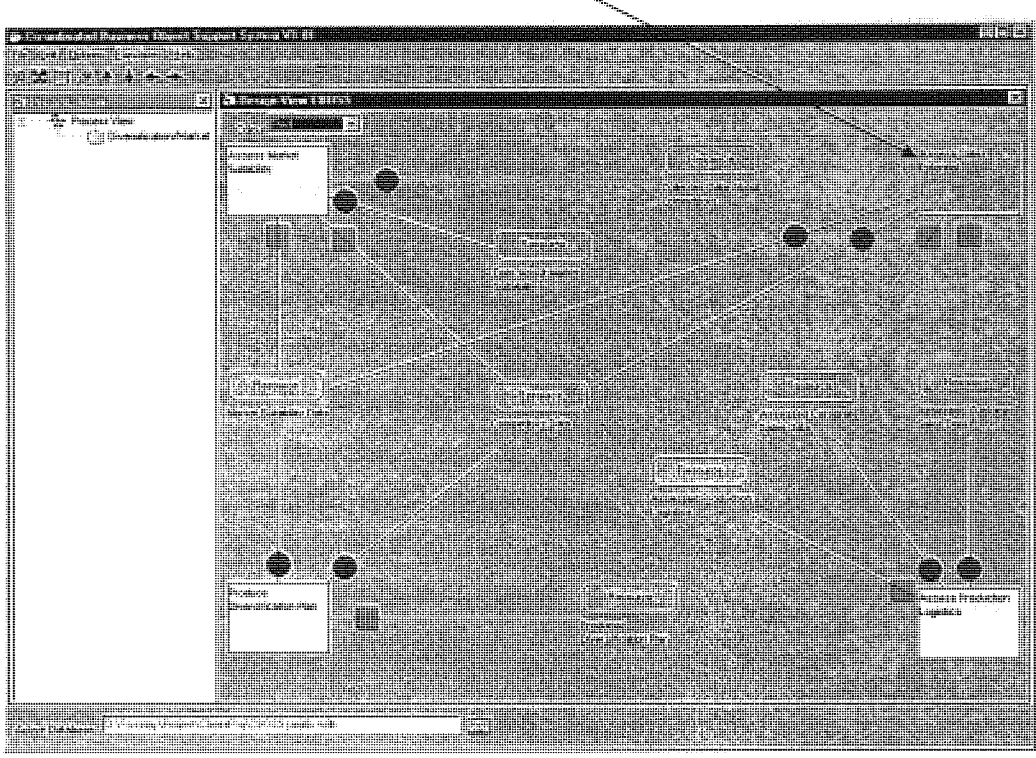

Figure 9.11. Anomaly detected in Diversification Into A New Market

The screen displays for this process are presented in appendix E. Specifically, E.1.1 to E.1.4 show the state of the activities associated to this process at the time the initial data collection was taken. The associated resources are displayed in E.1.5 to E.1.12. Significantly, the resources that are used and produced by the activities in this process belong in different locations, as the organisations involved in the process are part of a virtual collaboration. For example E.1.5 denotes that the resource 'Market Suitability Data' is positioned at 'Organisation 1(A)', whereas the resource 'Produced Diversification Plan' is positioned at 'Organisation 1'. In short, the 'Position' attribute, simply denotes the place or organisation, where these resources where positioned.

Attached to this process are four dependency patterns. The screen displays are shown in appendix E.1.13 to E.1.16. The first view displayed in E.1.13, denotes a flow and sharing dependency at the resource 'Market Suitability Data'. The same dependency types also exist at the 'Competitors Data' resource. This is shown in E.1.14. A flow dependency exists at the resource 'Assessed Corporate Sales Data' (see E.1.15). The same dependency type also occurs at the 'Assessed Customer

Sales Data' resource (see E.1.16). As noted in observation one in section 9.4.1 above, the user was heavily involved in the design phase of this process. Therefore, throughout this period, design and redesign of the process, in order to find the most efficient sequence, was conducted. For example, completing the assessment of the sales potential firstly, or concurrently along-side the assessment of the market suitability activity were amongst the issues that arose. Therefore, the views listed in E.1.13 to E.1.16, were considered helpful by the user when establishing the most efficient method of producing the diversification plan. Significantly, the same can also be observed with the 'Interdependency View' patterns that are detailed in E.1.17 to E.1.20. Here, information regarding the sequential and reciprocal patterns, allowed the user to establish which activities were reliant upon others in the system. Furthermore, it also forced the user to consider the contributory nature of each activity, and therefore determine if an activity should be designed as a sub-activity of another. Finally, the 'Constraint View' in appendix E.1.21, highlighted the various constraints that occurred over a period of time within the process. Although these are subject to change as the low level data changes, this view was of more benefit to the user when the process was running, in contrast to the dependency and interdependency views which were largely consulted throughout the design phase.

In drawing conclusions from this test, the evidence would suggest that the CBOSS provided support for tactical management decision making. Significantly, this test also highlighted the use of the design features of the CBOSS. Although this was not an area that was specifically tested, it highlighted two important aspects. Firstly, it allowed the user to design and redesign the process. In doing so it was possible to insert sample data, and thereby determine where in the process, 'constraints' may occur. Secondly, it allowed the user to recognise where different dependencies and interdependencies would occur, based upon the activities that were used in the process.

9.5. VERIFICATION TEST 3 (DESPATCH PRODUCT)

The final verification test in this chapter detailed the process of despatching products. Organisation 2, as detailed in 9.2.1.2, was used as the primary testing laboratory for this test. Three activities belong to this process. These are categorised as; 'Assemble Product', 'Despatch The Product' and 'Update System'. This is shown in figure 9.12 below.

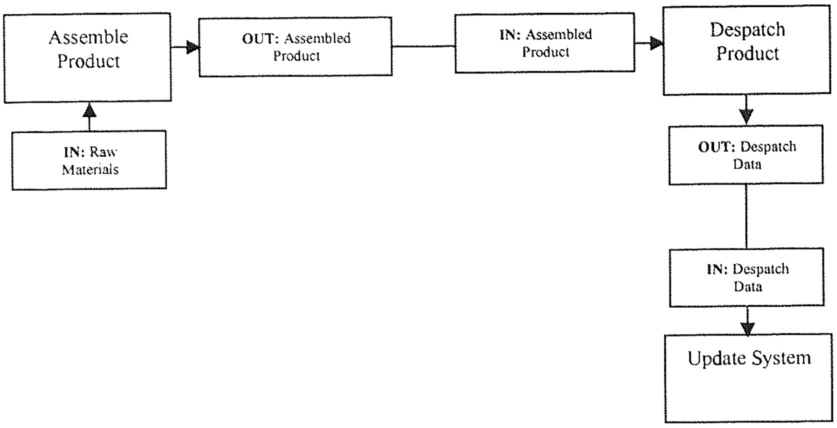

Figure 9.12. Process Model Level 1: Despatch Product

Specifically, the activity 'Assemble Product' consisted of various operational level sub-activities. From the perspective of testing the CBOM however, it was decided that these activities should not be included in the overall process for two distinct reasons. Firstly, it was possible to group these activities into composite activities, and secondly, by drawing parallels with research presented by Jaques (1989) and Anthony (1965) (see chapter two), it became clear there is a boundary between the tactical and operational levels in the organisation.

The process shown in figure 9.12 above, is a recurring process that runs indefinitely. However, it became clear throughout the field-study period, that this process was prone to functional problems with the late arrival of assembled products to the despatch department. Consequently, the despatch department could not process any products until the system had been updated. Therefore, anomalies that occurred with the activity 'Update System', had the potential to cause disruption to the 'Despatch Product' activity. From a testing perspective, this provided a good opportunity to test the modelling capabilities of the CBOM, and the functional robustness of the CBOSS. As part of the operational procedure of this process, one cycle is completed in one day. Therefore, the despatch data is updated onto the system at the end of each working day. In addition to the activities, three resources are represented in this process. Namely, 'Assembled Product', 'Raw Materials' and 'Despatch Data'. These are shown in figure 9.13 below.

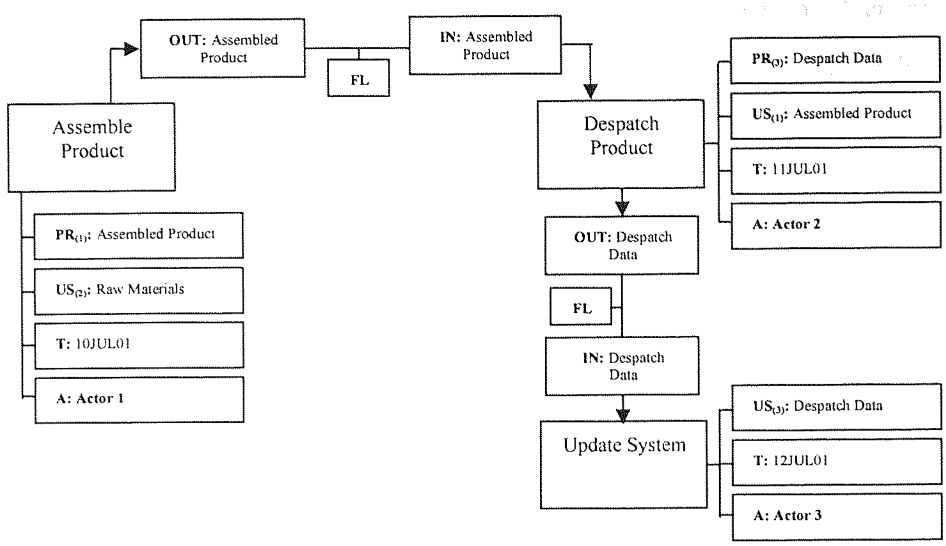

Figure 9.13. Process Model Level 2: Despatch Product

Furthermore, there are five separate dependency links in this process. These are shown in figure 9.14 below.

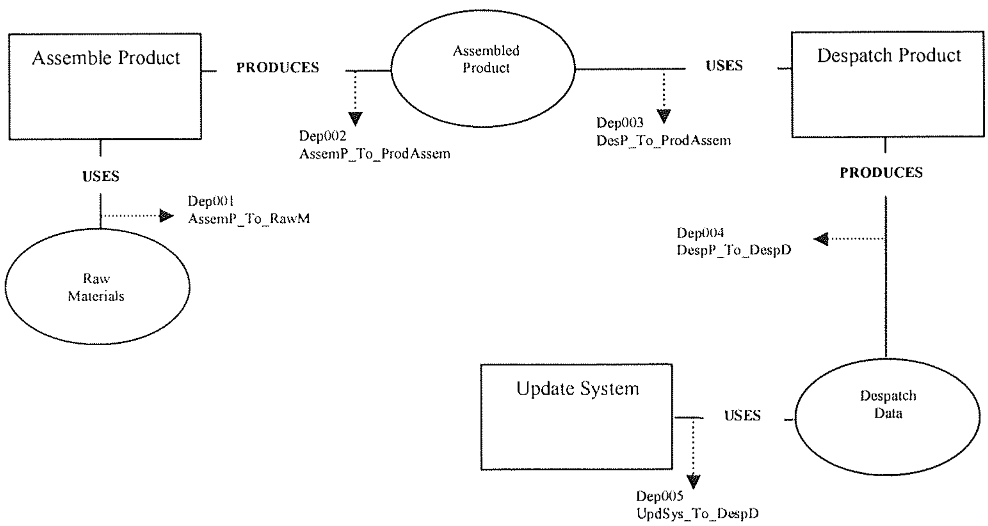

Figure 9.14. Co-ordination View: Despatch Product

Both tests, as detailed in section 9.2.3 above, will be applied to this process. These test areas are considered in greater detail below.

9.5.1. Application Of The First Test

Similar to the verification tests in sections 9.3 and 9.4 above, the initial collection of data from the field research is presented in appendix C.3. This verification test details a process that is ongoing, in so much as it loops perpetually. The process completes one loop at the end of each day, when the system is updated. Therefore, this process was observed during five cycles over a five day period. Over this time period, two anomalies occurred that resulted in delaying the completion of the process. These were recorded in the observations presented below.

Observation 1. As mentioned previously, the activity 'Assemble Product' consists of a variety of activities. Throughout the five day testing period, various issues were observed that had a direct impact upon the efficient running of this activity. Machine breakdowns and staff issues were amongst the most common. However, as these sub-activities belonged to the 'Assemble Product' activity, it was this activity that the CBOSS highlighted as an anomaly. Therefore, as the CBOM did not record these sub-activities, the user did not directly have the information detailing the exact cause of the problem to hand. Consequently, as the user could not find the specific problem, he therefore felt that the system should detail more information about the operational level activities. This however leads to an interesting observation regarding the construction of the CBOM. The theory states that organisational layers and their corresponding objectives and processes are separated into operational, tactical and strategic levels (Jaques, 1989; Anthony, 1965). CBOM, as part of its design, models the management processes at the tactical level. However, upon the evidence gathered in this verification test, it would appear that, it may be useful for the manager to have the ability to drill down to a further level if in some circumstances that would add to the management decision making capabilities.

Observation 2. The second observation detailed issues that arose with the activity 'Update System'. Throughout the five day period a problem occurred with this activity. The observer however, was informed that problems with this particular activity occurred on a regular basis and for a variety of reasons. In this particular instance, a problem occurred with the activity 'Despatch Product', owing to the 'Despatch Data' resource being unavailable. Consequently, this was relatively simple for the CBOM to model, and therefore the relevant low level data was amended. Furthermore, the activities where the anomalies occurred, were highlighted to the user by the CBOSS.

9.5.2. Application Of The Second Test

The 'Product Despatch' process, comprising its activities, resources and uses / produces relationships, are presented in the CBOSS screen display in figure 9.15 below.

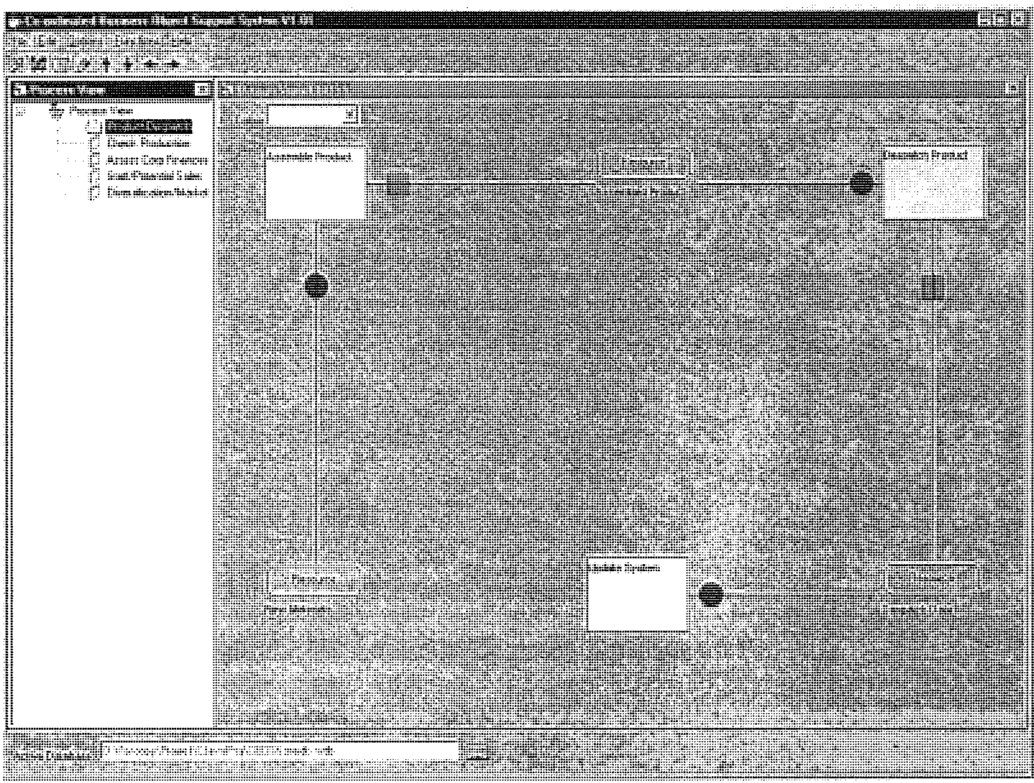

Figure 9.15. Despatch Product (CBOSS Diagram)

Further data concerning activities and resources, that were detailed in appendix C, were also inputted prior to the process becoming live. Throughout the five day period, various tests concerning the CBOSS functionality were applied in support of this test area. Specifically, appendix F details the various screen displays that were gathered. These will be referred to throughout the remainder of this section where appropriate.

The first and second observation in 9.5.1 above, detailed anomalies with the 'Assemble Product', 'Despatch Product' and 'Update System' activities. Consequently, figure 9.16 below shows a CBOSS screen display that highlights anomalies with all the activities in this process.

Anomalies Detected

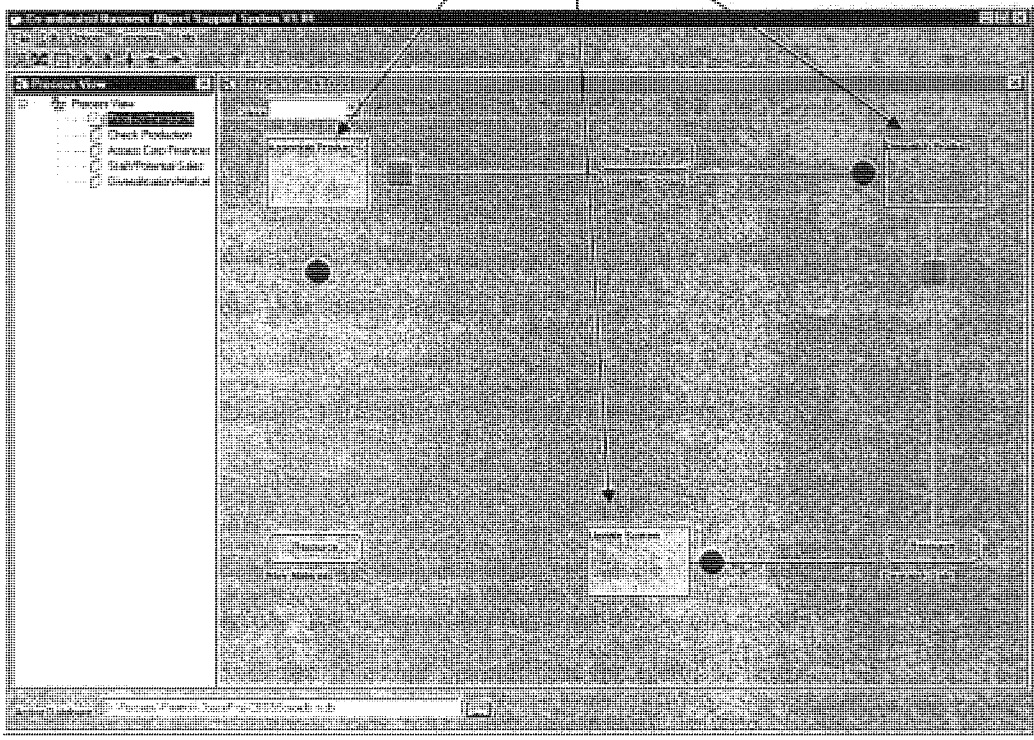

Figure 9.16. Anomaly Detected in Despatch Product

These anomalies usually occurred as a result of breaks in production. However, they did not last for a sustained period of time. In short, they were temporary delays that effected the overall product despatch at the end of each working day. Here, the delay was remedied by either moving staff to areas where problems occurred, or in the case of broken-down machinery, simply notifying maintenance staff. This however, was not scheduled by the manager who was using this system, but by designated operational level staff.

The 'Resource Accessible' and 'Resource Usable' flags shown in appendix F.1.4, F.1.5 and F.1.6, point to all resources being both usable and accessible. These screen displays are consistent with a process that is running efficiently. However, when anomalies with this activity became apparent, these flags changed to show the accessible and usable status of the resource. In addition, two dependency views outlined in appendix F.1.7 and F.1.8, detail two flow dependencies within the process. This part of the CBOSS functionality however, was rarely consulted. The same was observed for the three 'Interdependency Views' outlined in F.1.9, F.1.10

and F.1.11. It seemed that the likely reason for this, was owing to the user not requiring information regarding these areas, as the process was relatively uncomplicated. In short, as the findings of verification test two suggested, information regarding these views seemed to be more appropriate when designing and re-designing a new process. Finally, the 'Constraint View' in appendix F.1.12, also details a snapshot of the current state of the constraint view attributes for the process at a particular point in time. Here it outlines the existence of a prerequisite constraint at both the 'Assemble Product' and 'Despatch Product' activities. This area of the CBOSS functionality was used more often by the user, owing to it changing and providing up to date information regarding constraints between activities in the process.

In drawing conclusions to this test, it can be assumed that the CBOSS provided the necessary functionality for supporting tactical decision making. However, it became apparent over the five day period, that the specific information detailed in the activities and resources form was rarely consulted. Indeed, the same can also be observed as regards consulting the co-ordination views. In short, the main screen that displayed the current state of the activities was used more frequently than any other. Two main reasons can be targeted for this. Firstly, as there were only three activities and three resources, the process itself was relatively uncomplicated. Therefore, the user could determine the precise course of action based upon the actor who is responsible for the activity. Secondly, as the tactical level manager already had a high degree of familiarity with this process, remedying recurring problems, more often than not, did not require the use of a computer based information system.

9.6. SUMMARY OF THE TESTING PERIOD

Although presented in greater detail throughout the preliminary sections in this chapter, table 9.1 below summarises the overall results of the three verification tests presented in this chapter. For brevity, and ease of understanding, three ratings, detailing, 'Acceptable', 'Acceptable with some issues' and 'Major Issues' have been created in order provide an overall representation of the results of all three tests. Therefore, 'Acceptable', details that the test gained acceptable results that would point to the test criteria being met. Secondly, 'Acceptable with some issues' details that the test provided acceptable results yet there were some issues uncovered throughout the testing period. Finally, a rating equal to 'Major Issues', details that major issues arose from the test.

	Test 1	**Test 2**
Organisation 1		
Tactical Process 1	Acceptable with some issues	Acceptable
Tactical Process 2	Acceptable with some issues	Acceptable
Organisation 2		
Tactical Process 3	Acceptable with some issues	Acceptable

Table 9.1. Summary of the test results

It is important to point out that the above table is simply a snapshot that presents the overall results of the process. However, it does provide a guide as to the success of the test. Apart from testing the robustness of the CBOM in terms of modelling tactical management information, this testing period also brought to light various areas that were not specifically tested for. In themselves they can be considered fairly significant. As a result of this, these areas will be considered in further detail.

Process Changes. During the first verification test, an intervention from the company's head office, highlighted a need to change the process 'Staff/Sales Potential'. Throughout the initial design phase, this intervention was not anticipated. Therefore, when this occurred, it required changes to the process. The CBOSS functionality however supported these changes. Furthermore, it also highlighted the ability of the CBOSS to represent changes to a process, that occurred as a result of changes from the outside environment.

Development Issues. The second verification test detailed the process named 'Diversification Into A New Market'. Unlike the processes analysed in verification tests one and two, the user was heavily involved in the design stage of the process, as well as its testing phase, when it was considered to be 'active'. Although this was not specifically outlined as an area for testing, this activity, outlined the use of the CBOSS functionality for constructing processes. Furthermore, it also allowed the user to represent the process, with the activities that they wanted to track. As opposed to the author constructing the process for the purposes of testing. It could be argued that this had various advantages, owing to the user being more experienced than the author and subsequently having a greater knowledge of the environmental factors. Finally, this development phase, also highlighted alternative ways that the dependency and interdependency views could be used. For example, as the user entered different data into the CBOSS, the CBOM produced different interdependency and dependency occurrences. From this perspective, the CBOSS could be viewed as a tool that supported the development of processes, as well as being a tool that provided analysis about them. Further issues that were highlighted under this area, detailed the lack of use of various fields in the resources form. Specifically, the 'Shareability', 'Divisible', 'Concurrency', 'Usability' and 'Tangiability' fields were seldom referred to throughout the three verification tests.

Organisational Issues. Consistent with the research presented in chapter two, the organisation consists of various layers. Consequently, the CBOM, as a tool for supporting tactical level decision making, reinforced the need for the user to separate those tasks that were tactical in nature, from those that could be deemed operational or strategic. For example, at the operational level, decision making may consist of low level activities, this was shown in the third verification test (Despatch Product). Here, the 'Assemble Product' activity consisted of further operational level activities, namely; 'assemble product fittings' and 'place the assembled products in boxes'. Whereas at the tactical level, these activities were grouped into a composite process activity named 'Assemble Product'. In short, it was not necessary for the tactical level manager to be burdened with minutiae regarding the lower level activities, but to be presented with the overall picture as to the state of the product assembly. It could therefore be observed, that the CBOSS separates these functional layers, and apart from supporting tactical level information, required the user to model processes that were tactical level in nature.

9.7. CONCLUSIONS

The objective of this chapter has been to rigorously test the CBOM. This has been achieved by using three processes that were tactical type in nature. The three tests were conducted in two organisations that were situated in the engineering and software industries. This testing phase has presented various contributions towards understanding how effectively the CBOM supported tactical management decision making.

The next chapter will conclude this thesis. In doing so it will draw upon work presented throughout this study as well as assessing the research contribution made by this body of work.

CHAPTER 10

The CBOM Requirements Revisted

10.1. INTRODUCTION

Section 7.2 outlined four points that together form the underlying requirements for establishing the CBOM. In essence, this acted as a criteria list of things the CBOM should do. With this chapter being the last chapter in section C, it is perhaps worth revisiting these four points in order to establish if indeed this criteria has been met. It became clear throughout the testing phase, that the success of the requirements listed in section 7.2 hinged upon the testing phase outlined in chapter nine. In essence, it could be argued that a detailed analysis of section C would provide the user with enough information as to whether or not the four CBOM requirements have been met. Therefore, it is not the purpose of this section to simply reiterate what has already been written in previous chapters, but to simply provide supporting text that answers specifically, whether or not the CBOM functions as the requirements suggest it should.

10.2. THE REQUIREMENTS REVISITED

To recap, the requirements outlined in section 7.2 entailed the following:

- ***To model tactical management information.***
- ***To support the modelling of tactical level processes..***
- ***To provide a model that can be implemented into a fully functional information system.***
- ***To support the tactical management decision making process.***

The remainder of this section will consider whether or not these requirements have been met. In doing so, it will draw upon evidence presented in section C.

To model tactical management information. For the CBOM to be of any use, it was a requirement that it supported the information needs of the tactical level manager. The CBOM was responsible for modelling such information based upon the architecture of its modelling constructs. Chapter seven noted that tactical level information is processed by the second tier of the co-ordination model, by interpreting the data inserted into the four object models. From a functional perspective, evidence gathered from chapter nine proved that the co-ordination model performed as its design suggested in chapter seven, and also provided the tactical level managers' who were involved in the study with information that was appropriate for their tactical information needs. This is for two reasons, firstly, there were no problems arising regarding the functionality of the co-ordination model in the two test environments. Secondly, the results from the test yielded relatively successful results. Therefore, in the view of the author, the evidence presented throughout section C proves that the CBOM has the ability to present the user with tactical level information based upon data regarding their tactical level tasks.

To support the modelling of tactical level processes.. Most significantly, section 9.4 provided evidence that the CBOSS could be used by the user as a tool to construct, test and reconstruct tactical type processes. Indeed, prior to the testing period it was not assumed that this type of activity would be employed in the creation of tactical level tasks. Nonetheless, it could be argued that it is more accurate, as the tactical level manager in organisation one had a greater knowledge of the tasks he executed than the author. From this point of view, the toolset included in the CBOSS, which was a representation of the CBOM constructs outlined in table 8.1, was adequate enough for this requirement to be achieved. Furthermore, as chapter nine suggests, the modelling of tactical level processes throughout the entire testing period proved fruitful, as no problems were highlighted in these areas. Consequently, from a scientific perspective, there is no evidence to suggest that the CBOM did not do anything but provide the constructs for the accurate modelling of tactical level processes.

To provide a model that can be implemented into a fully functional information system. Chapter eight was primarily devoted to outlining the role of the Co-ordinated Business Object Support System (CBOSS). Specifically, the same chapter outlined the architecture of the CBOSS as being built around a standard two–tier architecture. In essence, the CBOSS represented an implementation of the CBOM. By implementing the CBOM into a fully functional information system, allowed the principles encapsulated in the CBOM to be tested in a real world environment. Before this could be applied however, it was necessary to translate the CBOM constructs to CBOSS constructs, therefore providing an environment with which the user could interact. This was delineated in table 8.1. However, having made this translation, it was necessary to test in some way whether or not the CBOSS provided an information system with which the user could interact.

From a systems perspective, the model functioned as it was intended to throughout the testing period. The implementation of the software was successful and the co-ordination constructs that were represented in the CBOSS were available for the tactical manager to use. Although it should be noted that such functionality was already a prerequisite before the testing phases could begin. Throughout the testing period the users entered data via the CBOSS interface. This data was processed by the co-ordination model and information based upon co-ordination constructs was returned to the user. All three tests presented examples of where the CBOSS returned such information (see appendices D, E and F). Consequently, it became clear towards the end of the testing period, that all the co-ordination constructs outlined in table 8.1 had been adequately represented. Therefore, the author is relatively satisfied that the CBOM was successfully implemented into a fully functional information system and that this CBOM requirement that this section addresses has been met.

To support the tactical management decision making process. The first point in the CBOM requirements listed above, outlined the role of tactical level information. From an information systems perspective, information is used to support the decision making capabilities of the management level it relates to. Evidence gathered throughout the testing period outlined in chapter nine, also suggests that the CBOSS can be used to support the tactical level decision making process. In the CBOM, the decision making process is supported by the various screen views that are presented by the CBOSS (see appendices D, E and F). Throughout the testing phase, some of these screens where used more than others, as the users seemed to consider some parts of the co-ordination model more useful than others. Regardless of this the functions where still available for use. The results from the testing phase in chapter nine suggested that the CBOM supported the tactical decision making process. Therefore, the author is relatively satisfied that the requirement in the list above has been met.

10.3. CBOM AND OTHER INFORMATION MODELS

By forwarding the CBOM as a model for supporting the tactical management decision making process, does to some extent, assume that other approaches are not suitable or indeed appropriate. In essence, as figure 1.4 outlines, the role of this research is to determine a certain amount of understanding through interpretation of the current academic literature. From this, it will bring about, or in the case of this thesis, suggest through intervention, a different approach. The CBOM, by the end of this thesis will be presented to the reader as an object model that combines co-ordination constructs, that can be used to support tactical management decision making. However, in ending this section, it is perhaps necessary to answer two fundamental questions that the reader may require an answer to at this point in the thesis. Firstly, how the CBOM relates to other information models and secondly, why it is more suitable for supporting the decision making needs of tactical management. These areas are reviewed below:

10.3.1. How The CBOM Relates To Other Information Models

Information models from an information systems perspective, details the accurate representation of real world entities. This has certainly been the underlying aim of many data modelling approaches (Codd, 1970; Chen, 1976; Rumbaugh et al, 1991). From a design perspective, the CBOM has followed very similar lines to other information models in terms of its design, as it has followed this common aim. Indeed an information system that did not accurately reflect the real world entities it was supposed to represent, may be prone to operational inaccuracy. In addition, other similarities to be outlined between the CBOM and other information models, represent the motive to support the decision making capabilities of the user. In the case of the CBOM of course, this is the tactical level manager.

10.3.2. Why Is The CBOM More Appropriate Than Other Information Models?

In essence, the short answer to this question, would be that other information models do not specifically support the tactical management decision making process as well as the CBOM. In essence, the CBOM takes the perspective that information from the mapping of tactical level processes, based upon an object model that uses co-ordination constructs, provides a more appropriate platform than other models from which tactical level managers can derive information based upon tactical level processes. In order to support the argument presented in this thesis, an extensive testing period has been carried out in chapter nine. The results of these tests pointed to the CBOM as a modelling approach that returned positive results when applied to tactical level tasks. Other information system models, that support management tasks, such as management information and decision support systems, are generally built around supporting a particular management function. Furthermore, as is the case with such systems, they are simply reporting tools that supply the manager with reports in the form of spreadsheets, pie charts and graphs etc. Such reports generally reflect things such as the end of day sales figures, or the profit that has accrued over the working week. In understanding this point, as regards the operations of these types of systems, outlines where the CBOM is different. From this perspective, it is this difference that in essence, provides the answer to this section. In short, it highlights why the CBOM is more appropriate. For clarity, these differences are listed below.

- The CBOM represents tactical management processes.
- The CBOM supports tactical management decision making using co-ordination constructs.

The CBOM represents tactical management processes. It can be ascertained that the difference between the CBOM and the conventional DSS or MIS is that the CBOM represents a modelling approach that specifically supports the tactical management function regardless of the industry of the actual process. In short it is not limited like many other information system to being customised to a particular market or industrial sector.

The CBOM supports tactical management decision making using co-ordination constructs. By implementing the CBOM with the CBOSS, has allowed various software design principles to be applied to the way the user interacts with the model. The approach adopted in the CBOSS is to allow information to be returned to the tactical manager based upon co-ordination constructs. The reason that co-ordination was adopted, was owing to its flexibility for representing organisational processes. In addition, the CBOSS also allowed up to the minute changes regarding the CBOSS system to be reported to the manager. From this perspective, the co-ordination approach has not been used in this way before.

10.4. CONCLUSIONS

In concluding this section it can be ascertained that it was the seeming lack of a functional capacity within the standard DSS and MIS, that determined that the CBOM is more appropriate than other information systems models. The next chapter will conclude this thesis.

CHAPTER 11

Conclusions

11.1. INTRODUCTION

This thesis has presented the Co-ordinated Business Object Model (CBOM) as a possible approach for supporting tactical management decision making. Underpinning the development of the CBOM, is a research approach that, as outlined in section 1.4, leverages knowledge from subject areas such as co-ordination theory, business object modeling, organisational structure and management decision making.

The initial understanding of the problem stated in section 1.2 outlined a general acceptance that there is a need for change, in so much as an improvement in the quality of tactical management information is required. In order to address this, this thesis outlined the use of co-ordination theory as a method for supporting the analysis of tactical level processes. In drawing a close to this study, this chapter will consider the architecture of this thesis in light of the problem area stated in section 1.2. It will also detail the contribution made by this body of work. In addition, this chapter will also consider any possible future work that may lead from this research, before drawing any final conclusions.

11.2. THE THESIS REVIEWED

This thesis has been developed over three sections. Section A was comprised of three chapters. The overall emphasis of this section was to detail the role of the tactical level manager, the environment in which they reside, information systems used for making tactical level decisions and the information needs that managers at the tactical level in the organisation require. In doing so, chapter two - the first chapter in section A - introduced both the role of the tactical level manager and the environment in which management at this level reside. The distinguishing characteristics of this environment focused upon traditional hierarchical structures, as well as more modern structures such as horizontal systems and autonomous work units. These structures targeted the significance that they have upon tactical management decision making. Furthermore, this chapter also pointed out, that regardless of how vertical or horizontally aligned these structures are, organisational tasks still have to be achieved, and as these tasks in some circumstances will cross departmental levels, specific employees will be assigned tasks based upon their degree of specialisation. Leading on from this, chapter three delineated the wider perspectives of management information systems, and their role in the production and dissemination of information for supporting the tactical management information need. In short, the purpose of this chapter was to further extend the information presented in chapter two, by examining how tactical level management make tactical level decisions. Chapter four, the final chapter in section A, narrowed the specific focus of this study, and detailed the constituent areas of tactical management information. In order to achieve this, it considered two primary areas. Namely; 'The Scope of Tactical Management Information', which detailed the wider perspectives of tactical level information such as its characteristics, as well as its corresponding level of abstraction along with its use and application. Secondly, it considered various approaches and suggested architectures, which provided an exploration into various current approaches, and technologies used for delivering information to the management decision making process.

Section B further developed the research content. In doing so, it considered two main areas. The first in chapter five, extended the information systems theme discussed in section A. It directed the emphasis of the research towards investigating various modeling approaches used in the development of information systems. Such models included conventional data models and object oriented approaches. Although this chapter surveyed a number of modeling techniques that are employed in the design of an information system, it was found that these tools, by themselves, offer little scope for providing the tactical level manager, with a model with which to make tactical-type decisions. From this perspective, a more high level approach to business object modeling was required. Consequently, chapter six introduced the role of co-ordination theory, as developed by Malone and Crowston (1994). In doing so, it outlined the role of the co-ordination framework, as well as its ability to provide an approach for representing tactical level processes. In addition, this chapter, also examined the academic literature regarding co-ordination. Chapter six also detailed significant embellishments made to the co-ordination approach, by various authors. By collating and assessing the distillation of these approaches, it was possible to represent the salient parts that could be used by this thesis for representing tactical level processes. These, along with a suitable notation were presented in table 6.1. As the co-ordination theory framework is centred around the analysis of organisational processes, the approach presented in chapter seven - the introductory chapter to Section C - begun to detail an information systems model that specifically supports tactical management decision making.

Using knowledge leveraged from chapter five regarding the business object model, with the co-ordination theory constructs presented in chapter six, chapter seven presented the 'Co-ordinated Business Object Model (CBOM). Presented in two layers, the CBOM was developed throughout the course of this study. It's principal aim was to provide an object model that could be used to present tactical level information. Chapter eight detailed a prototype implementation of the Co-ordinated Business Object Support System (CBOSS). The aim of the CBOSS, was to support the CBOM implementation. The CBOSS, via a graphical user interface (GUI), provided the user with the functionality to interact with the CBOM. In this way, it allowed the user to develop and analyse tactical level processes. In order to provide an environment in which the user could create such processes, the CBOSS system represented all the constructs detailed in the CBOM model. Specifically, table 8.1 presented the CBOM and corresponding CBOSS notation. The CBOSS also presented a system with which the CBOM could be rigorously tested. Chapter nine detailed the procedure for testing the CBOM. This consisted of two tests, applied to three tactical-type processes, that were taken from two organisations.

11.3. CONTRIBUTION OF THE RESEARCH

The overarching aim of this thesis, as stated in section 1.5, has been to find a better way of generating information for supporting tactical management decision making. In order to achieve this, it has been necessary to gather information from key areas in the research literature. In order to better present the information gathered, it was decided to group related chapters into corresponding sections. Therefore, references to these sections and chapters will be made when reviewing the research objectives. The objectives that have guided the research focus of this study were highlighted in section 1.5. The objectives that are detailed in this section, will be reviewed in light of the research contribution made throughout this project.

OBJECTIVE 1 - *To explore the tactical management role in the organisational structure. This will be achieved by;*

OBJECTIVE 1.1 - *Investigating the hierarchical structure of organisations.*

All employees within the organisation reside at a particular level in the organisational hierarchy. This is a fairly common view of the organisational structure and seemed a suitable place to start when investigating this topic. Specifically, the focus of chapter two highlighted the hierarchical structure of the organisation. Here, it presented research that outlined that all organisations represent a certain degree of hierarchy. Consequently, the amount of layers in an organisation, will represent how 'vertically' or 'horizontally' aligned the organisation appears. The research literature details an abundance of work that highlights various forms of organisations. However, such forms all seem to revolve around a hierarchical structure that contains a degree of hierarchy. Therefore, it was decided to focus upon the hierarchical structure of the organisation, rather than investigating the many individual forms that have been posited by various authors. This objective proved to highlight the hierarchical structure of organisations. In doing so, it provided a necessary grounding in understanding the type of organisation within which tactical level managers operate. The next objective, specifically targeted the tactical management role in the organisation.

OBJECTIVE 1.2 - *Exploring the tactical management role in the organisational structure.*

Chapter two noted that the hierarchical structure of the organisation is split into three distinct areas. These are 'Strategic', 'Tactical' and 'Operational'. All employees in an organisation carry out roles that equate to the position they hold in the organisation. The aim of this research objective however, was to outline the role of the tactical level manager in the organisation. By surveying the research literature, it was clear that an abundance of sources emerged relating to work that detailed the strategic and operational management roles. Relatively few sources however, seemed to concentrate upon the role of the tactical level manager. In order to fully understand the role of the tactical level manager, it became apparent, that it would be necessary to investigate the roles of strategic and operational level management, as they interact to some extent with tactical level management. The various sources that supported this investigation were available and consequently, were highlighted in chapter two.

OBJECTIVE 2 - ***To investigate the information needs of tactical management.***

Having outlined the role of the tactical manager and the organisational structure in which they operate, it emerged that different management levels have different information needs (see chapter four). This was strongly highlighted in the literature, as much work into the different information systems that are used to support management decision making have been presented. However, in order to expose the information needs of the tactical manager, various definitions outlining the meaning of information have been investigated. Such definitions proved useful when developing an understanding as to the nature of tactical level information in an organisational context. The primary focus of these definitions highlighted the use of information for knowledge generation.

The research findings relating to this objective, highlighted the role of information not only in the classical organisational structure but also in the virtual organisation. As the concept of the virtual organisation was relatively prominent in the research literature, it was considered necessary to investigate the information needs of tactical level management. However, having outlined the role of tactical level information, it seemed appropriate to investigate the various systems that would deliver that information. This was the focus of the next research objective.

OBJECTIVE 3 - ***To investigate the role of information systems that are used to support tactical management tasks.***

By examining the research literature, it became clear that an abundance of research into the various types of computer based information systems was available.

Consequently, the point of departure signified by this objective, outlined specifically, various computer based information systems approaches that could be used for delivering tactical management information. Consequently, chapter three outlined the role of Management Information Systems (MIS), and Decision Support Systems (DSS), as possible approaches for delivering tactical information. The research literature details the role of both the MIS and DSS as tools for bridging the gap between low level data and meaningful information. In addition, they also detail their use as systems that help to reduce the problem of information overload. Consequently, this research objective outlined an investigation into the information systems that could be used by tactical management in supporting their allocated tasks.

Having reviewed this, it became clear that various information systems were already available for supporting management tasks. Following a thorough investigation, three distinct areas were identified as;

- Collaborative Approaches
- Enterprise Resource Planning (ERP)
- Integrative Technologies

These areas were explored in detail and as a consequence, many areas that are relevant to this study were uncovered. For example, various collaborative approaches such as the Light Weight Directory Access protocol (LDAP) and Workflow Access protocol (SWAP) were considered. The use of Enterprise Resource Planning (ERP) and Enterprise Application Integration (EAI) systems, as a means of improving the information supply chain between all decision makers in an organisation were also examined. Integrative technologies such as CORBA, DCOM as well as the Distributed Transaction Processing standards, which are defined by the Open Group (XA, XCPI-C, TX, TxRPC, XATMI, Reference Model V.3), were detailed as standards that aim to develop further integration within the distributed computing framework. The distillation of the findings pertaining to this research objective detailed the unsuitability of these approaches for supporting tactical management decision making. Therefore, a deeper investigation into the modeling constructs that support information systems needed to be undertaken.

OBJECTIVE 4 - *To investigate modeling approaches for supporting tactical management decision making.*

Chapter five highlighted various modeling methods that are incorporated into supporting an information systems functionality. Upon examining the literature, it became clear that many modeling approaches were available. Therefore, in order to conduct a suitable review of the literature, this chapter focused upon both static and dynamic modeling methods. A review of the static methods, highlighted conventional data models such as the Relational data model (Codd, 1970) and the Entity Relationship data model introduced by Chen (1976). Further extensions to these approaches such as the named Relational Model Tasmania (RM/T) Codd (1979), and the Extended Entity Relationship Model (EERM) were also considered. In addition to this, an investigation into Object Oriented Analysis (OOA) (Coad & Yourdon 1991), and the Object Modeling Technique (OMT) Rumbaugh (1991) also underlined various object oriented approaches to data modeling.

The dynamic data modeling methods that were investigated, detailed a broad review of various approaches. Here it was outlined that static methods do not support dynamic aspects to data modeling, and therefore an analysis of the models that support the analysis of the object's states, processes, events and behaviour would be required. Finally, the notion of the 'Business Object' was examined as a modeling approach for supporting tactical management decision making. Here, a distinction was made between the conventional object and the business object. Moreover, although it was discovered that the business object maintains a great deal of flexibility for modeling business processes, it was concluded that the constructs used in object oriented approaches, revolved around events changing states and behaviour. Consequently, they do not provide any functionality for analysing tactical level processes in any greater depth, than other dynamic modeling approaches. From this perspective, a higher level approach to returning information based upon tactical level processes was needed. This approach detailed the use of co-ordination theory.

Co-ordination theory (see chapter six) as presented by Malone and Crowston (1990; 1994), details an approach for co-ordinating organisational processes. In doing this, the co-ordination framework outlines the modeling of activities within a process.

Furthermore, it places the emphasis on analysing the dependencies that exist between activities in a given process. From this point of view, this study explored the possibility that the co-ordination framework, could add richness to the business object model for supporting tactical level processes. Therefore, the next step was to develop a modeling framework that translated the modeling constructs of co-ordination theory, into an object modeling environment that could be used to provide tactical level information.

OBJECTIVE 5 - ***To develop a modeling framework for supporting tactical management decision making.***

In order to validate the theory development of this study, the 'Co-ordinated Business Object Model' (CBOM) (see chapter seven) has been developed. The development of this model centred around providing tactical level information, that is appropriate for supporting tactical management tasks. In essence, this model has shown that the modeling constructs developed as part of the co-ordination theory framework, can be translated into an object modeling environment. Such an approach has not been considered before in the literature, and from this perspective the CBOM adds richness to the business object model for supporting tactical level information. Therefore, this research objective allowed the theory development of this study to be grounded into a software application that could be tested. In order to test the efficacy of the CBOM, it was necessary to develop a system that could support the CBOM implementation. This was achieved by developing the 'Co-ordinated Business Object Support System' (CBOSS).

OBJECTIVE 6 - ***To develop a software tool for demonstrating the modeling framework suggested in objective 5.***

The 'Co-ordinated Business Object Support System' (CBOSS) (see chapter eight) has been developed as a prototype application that supports the CBOM implementation. The CBOM constructs along with the CBOSS functionality, realise an information system that can be used to support tactical level decision making. The CBOSS was designed around a typical two-tier architecture, that separated the 'Application Layer' from the 'Database Layer'. This particular architecture figures quite prominently in the research literature. Upon implementing the CBOM, the CBOSS could be used as a vehicle for testing the robustness of the CBOM. This was categorised in the final research objective.

OBJECTIVE 7 - *To test the modeling framework suggested in objective 5, in order to derive results as to the efficacy of the framework for supporting tactical management decision making.*

The research literature outlines the importance of testing the robustness of an information system. Various sources were used to outline this in chapter nine. By testing the CBOSS in a real organisational environment, allowed for a thorough analysis of the robustness of the CBOM. The testing procedure used in order to achieve this research objective, outlined three tactical type processes from two organisations. The results suggested that the CBOM has the ability to support tactical management decision making. Therefore, this objective outlined not only the efficacy of the CBOM but also the robustness of the CBOSS application.

Finally, in order to conclude this section, section 1.5 detailed that the overall aim of this study could be summarised by the following question:

> ***"Is there a better way of generating information for supporting tactical management decision making?"***

Now that this thesis is in its closing stages it is useful to recap and assess if indeed the research contained in this study has suggested a better way of generating information for supporting tactical management decision making. In essence, it can be concluded that this thesis in its preliminary stages highlighted issues with the supply of tactical level information. Such issues outlined the reliance of the tactical level manager upon report based tools that did not provide a deep or sufficient enough view of the organisation, or simply that the manager had to rely upon aggregated operational level data. From this point of view, the thesis set out a research approach that was consistent with the five stage research approach suggested by Susman (1983) (see figure 1.3). Here a comprehensive review of the academic fields in both the tactical management and information systems domain was completed. Based upon this initial research, this study developed the Co-ordinated Business Object Model (CBOM). The CBOM leveraged knowledge from both object orientation and co-ordination theory. In short, it identified parts of the co-ordination theory approach that could be used to support the behavioural aspects of the business object model. From this point of view, co-ordination theory, has not been used in this way before. In support of this approach, the verification tests provided evidence that the concept of the CBOM does provide "a better way of generating information for supporting tactical management decision making?" Therefore, the author would argue that development of the CBOM, as an object modeling approach, can be regarded as the main contribution made by thesis. However, further refinements to the work presented throughout this thesis have become apparent. These are detailed in the section below.

11.4. FUTURE WORK

At this point it is usual to underline any advancements that can be made upon the work presented in this study. Various ideas that became both apparent and implicit to the author throughout this thesis regarding any future work are detailed below.

11.4.1. Extensions to the CBOM

The purpose of the CBOM was to provide a modeling approach that supported tactical level decision making. From this perspective, the CBOM modeling constructs do not allow the user to model operational activities that belong to tactical level activities. Moreover, the constructs do not extend to modeling the strategic level objectives with which tactical level activities belong. Although this does not effect the modeling of tactical level processes, it is an area that is worth further investigation, as it allows the user to state more supporting information regarding the tactical level process and the strategic and operational level processes to which it relates. As detailed in chapter nine, it became clear that such extensions to the CBOM model would provide the manager with more detailed information.

11.4.2. Transfer From Research To Development

The results of the testing procedures outlined in chapter nine, were satisfactory, and although it was not part of the scientific process of testing the system, the feedback from the mangers in both organisations was very complementary. Therefore, it is the author's intention to further develop the CBOSS into a more robust piece of software. In order to do this the author will make enquiries for exploiting various avenues of funding from the business development unit that is attached to Aston University, and whose purpose is to help in the exploitation of new ideas formulated from research. Furthermore, as part of the transfer from research to development, the author also recognises that the profile of the CBOSS information system, needs to be enforced on a wider scale, therefore, a presence at a number of conferences with the intention of presenting not only the practical uses of the CBOSS but also any further developments to be made, should also be noted as an area of future work.

11.4.3. Extensions To The Testing Period

Throughout the testing phase of this work, it became clear the overall testing phase could have been made more robust, and consequently such improvements could be used as a basis for future work. These areas that have been identified as the following:

- A more rigid introduction of the co-ordination constructs.
- Front end user testing should be carried out in further tests, in order to assess the manager's capability to use the CBOSS GUI.
- Further testing should be carried out with organisations in virtual environments.
- A more robust testing procedure should be implemented that tests the CBOSS in a larger variety of organisational structures.
- More mission critical testing of tactical level processes.
- A longer period of time should devoted to further testing periods.
- Evaluation of the CBOM and CBOSS design.

Owing to the weight and importance of the points listed above, each will be considered in further detail below.

A more rigid introduction of the co-ordination constructs.

Towards the start of the testing process it became clear that before the user could use the CBOSS for analysing and developing tactical level processes, they had to learn and understand the co-ordination constructs that were contained in the

CBOSS. Therefore, as the testing procedure outlined in chapter nine only required the author to deal with a relatively few number of tactical managers, explaining the concepts was a relatively straight forward task as the author could provide individual tuition. However, it is not hard to envisage that if the testing laboratory became larger, i.e. a hundred or more organisations then the author would not have been able to have finished the testing procedure within the required time frame. Therefore, although the above points in this area did not effect the testing procedure carried out in chapter nine, it is crucial that any further testing addresses it, as it would be crucial that the user, in this case a tactical level manager, became aware of the co-ordination constructs in order to allow for a smooth and efficient start to the testing procedure.

Front end user testing should be carried out in further tests, in order to assess the manager's capability to use the CBOSS GUI.

The author has arrived at this conclusion for a number of reasons. The first reason was adequately outlined by Hall (2001), 'It is well accepted that people [in this case of this study this would be tactical level managers] have problems using new technology such as consumer products, systems and software'. With reference to this statement, it was observed that the users in both organisations had to learn not only how to use the system i.e. the CBOSS GUI functions but to also understand the co-ordination constructs that were part of the systems functionality. Consequently, as the user had to learn how to use the front end, a valuable amount of data regarding how the user interacts with the CBOSS system could have been captured. Such data would further have outlined not only naturally occurring deficiencies with the usability of the CBOSS, but would have also had an impact in future development of the CBOSS system. Although, the author recognises that time constraints were prevalent throughout the testing of this research, in hindsight, a 'user-centred' design schedule along with an evaluation and testing procedure would have been beneficial.

Further testing should be carried out with organisations in virtual environments.

Verification test two tested a process in a virtual environment. Such testing proved useful for testing how the CBOM model and the CBOSS system held up in an environment that was in the virtual domain. However, as the method of doing business online becomes more embedded into the organisations operational tasks, so growth of the virtual organisation will become more prevalent. As the CBOSS system supports the analysis of tactical level processes in the virtual environment, the author feels that a rich source of data could be derived from specifically testing the CBOSS system in more virtual environments. By doing this, more data regarding any possible differences between the analysis of tactical level processes in a virtual environment as opposed to a static physical environment could be derived. Moreover, further test results would also contribute value to the test already carried out in verification test two.

A more robust testing procedure should be implemented that tests the CBOSS in a larger variety of organisational structures.

Section A provided an extensive review of organisational structures along with the tactical management function in those organisations. However, in order to thoroughly test the functionality of the CBOSS system it could be argued that it should be tested in a larger variety of organisational types. For example, a testing procedure in line with the research presented in section A, should test a number or processes in horizontal and vertical organisation only and derive testing results as to how the CBOSS performed in these specific organisational structures. This would then provide a valuable source of data regarding how the CBOSS performed in a vertical or a horizontal structure overall. This is in contrast to the testing plan devised in chapter nine, which simply tested the processes in separate organisational structures. From this perspective, there is no data with which to compare the performance of the CBOSS in vertical or horizontal structures only.

More mission critical testing of tactical level processes.

The CBOSS listener throughout the testing period provided the tactical level manager with up to date information regarding the state of every activity within the process. However, the processes that were contained within the testing procedure, with the exception of verification test three (Despatch Product), were processes that were executed over a long period of time. Therefore, in order to more rigorously test the CBOSS tool, more tests on processes that were mission critical, in so much as they executed within a shorter time period, would provide valuable data regarding the robustness of the CBOM when operating in more stressful conditions.

A longer period of time should devoted to further testing periods.

The criteria devised to test an information system will by and large be dependent upon what is actually being tested. For example, owing to the time factors involved in testing the three verification tests, it was feasible to test verification test three (see section 9.5) in a day, whilst verification test one (see section 9.3) and three (see section 9.4) took slightly longer than was required. Therefore, although the author is relatively satisfied with the testing procedure stated in chapter nine. It could be argued that if more organisations were incorporated into the testing procedure, then a more valuable source of data could be captured and hence used for future further developments of the CBOSS system. Various problems owing to time constraints, prevented the implementation of a longer testing period.

Evaluation of the CBOM and CBOSS design.

The testing plan devised in chapter nine provided a good insight as to how the theory development of this thesis performed in a real life organisation. However, prior to testing the CBOM there was no coherent evaluation of the design process itself. From this, is should be noted that although this did not effect the actual results derived from the tests, it could be argued that by evaluating the design process of both the CBOM and CBOSS prior to testing them, may have led to a more efficient design. Therefore, it is recommended that if any future testing is to be done, with any modified version of the CBOM or CBOSS, then an evaluation of the design changes should be carried out before testing.

11.4.4. Further development of The CBOSS

Throughout the testing period, various performance issues regarding speed of processing and process design with the CBOSS became apparent. These were partly owing to the software platform used to build the CBOSS, as well as heavy processing required to run the GUI. Although this did not effect the testing period, further embellishments to the CBOSS design should be considered if further testing is to be planned. Consequently, various software packages that allow for the development of portable and secure software such as Perl/TK, or Java should be reviewed as part of any future development to the CBOSS.

11.5. CONCLUSIONS

This thesis has presented an approach for supporting tactical level decision making. The development of the CBOM and its constructs have been developed over a period of three years. This has provided sufficient time to develop, improve and prove the effectiveness of the CBOM under test conditions. In addition, there has been sufficient time to synthesise the research conducted over this time period, in order to furnish a current scientific position with which the author is relatively satisfied. The research contained within this thesis, has shown that by encapsulating co-ordination constructs in a business object model, can offer bridges for supporting tactical management decision making. Indeed, the development of the CBOM for outlining such an approach, has in effect, detailed the main contribution presented in this thesis.

References

Abinavam, S. (1998), "San Francisco Concepts and Facilities", *International Technical Support Organisation*, February, IBM, SG24-2157-00.

Ackoff, R. L, (1967), "Management Misinformation Systems", *Management Science*, 14(4), pp.147-157.

Ahituv, N. (1980), "A Systematic Approach Towards Assessing the value of an Information System", *MIS Quarterly,* 4(4), pp.61-75.

Ahituv, N. and Neumann, S. (1990), *Principles of Information Systems for Management,* Third Edition, Wm. C. Brown Company Publishers, Dubuque, Iowa.

AllFusion (2003), AllFusion, [Online]: http://www3.ca.com/Solutions/Product.asp?ID=255

Alshawi, A. (1991), "Semantic and Organisation Considerations in Database Conceptual Modelling: The Semantic Conceptual organisational Model", PhD Thesis, The University of Aston.

Anthony, R. N. (1965), *Planning and Control Systems: A framework for Analysis*, Graduate School of Business Administration, Harvard University.

Ardell, Bill. (1993), "The Critical Word", *Business Quarterly,* Spring, pp.118 - 120.

Argenti, J. (1974), *Systematic Corporate Planning*, Nelson, London.

ARIS (2000), The ARIS product Suite and Trade, [Online] Available: http://www.idsscheer.com

Ashby, W. Ross. (1956), *An Introduction to Cybernetics*, Chapman & Hall, London.

Baskerville, R. and Wood-Harper, A. T. (1996), "A critical perspective on action research as a method for information systems research". *Journal of Information Technology*, 11, pp.235-246.

Beer, S. (1967), *Cybernetics and Management*, (Second Edition), London: English Universities Press.

Beer, S. (1979), *The Heart of The Enterprise*, Wiley.

Benett, Gordon. (2001), "LDAP: A Next Generation Directory Protocol", 27th Sep, Available Online: http://www.intranetjournal.com/foundation/ldap.shtml

Benyon-Davies, P. (1996), *Database Systems*, Macmillan Press, Basingstoke.

Best, D. P. (1996), *The Fourth Resource: Information and Its Management,* Aslib/Gower, Aldershot, Hants.

Booch, G., Rumbaugh, J. and Jacobson, I. (1998), *The Unified Modeling Language User Guide*, Addison-Wesley, Reading, MA.

Boulding, K. E. (1956), *The Image*, Ann Arbour, MI: University of Michigan.

Braa, K. and Vidgen, R. (1995), Action Case: Exploring the middle kingdom in information systems research methods, In: Kyung, M., & Mathiassen, L., editors, *Proceedings of the Third Decennial Conference on Computers in Context: Joining Forces in Design*, August, pp.14-18. Aarhus University, Denmark.

Braa, K. and Vidgen, R. (1996), An IS research framework for the organisation as laboratory. In: Kyung, M., & Mathiassen, L., editors, *Computers In Context: Joining Forces In Design*, MIT Press.

Brachman, R. J. and Levesque, H. J. (1985), *Readings in Knowledge Representation*, Los Altos, CA: Morgan Kaufmann.

Braman, S. (1989), "Defining information: An approach for policy-makers", *Telecommunications Policy*, 13(3), pp.233-242.

Brockie, S. and Golder, P. A. (2000), "A co-ordination theory approach for developing a tactical management support system", *in Proc. The 6th International Conference on Information Systems, Analysis and Synthesis ISAS,* Orlando Florida, pp.142-147.

Brockie, S. and Golder, P. A. (2001), "Supporting Tactical Level Co-ordination In The Virtual Organisation", *in Proc. International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, L'Aquila, August 6 - August 12,* Scuola Superiore G. Reiss Romoli.

Brown, W. (1971), *Organizations*, Heinemann.

Bryan, Lowell. (2000), "Managers are becoming nodes", *BusinessWeek Online*, August 28th, Available: http://www.businessweek.com/2000/00_35/b3696034.htm

Burk, C. F. & Horton, F. W. (1988), *Infomap: A Complete Guide to Discovering Corporate Information Resources*. Englewood Cliffs, New Jersey, Prentice Hall.

Burn, J. M., Marshall, P., & Wild, M. (1999), "The emperor has no clothes-time to address the virtual organisation," in *Proc. Managing Information Technology Resources in Organisations in the Next Millennium. 1999 Information Resources Management Association International Conference,* 1999, pp.219-26.

Burns, T. and Stalker, G. (1961), *The Management of Innovation*, Tavistock, London.

Camarinha-Matos, Luis. M. (2001), "Execution system for distributed business processes in a virtual enterprise", *Future Generation Computer Systems*, 17, pp.1009-1021.

Catalysis, (2003) , [Online] Available: http://www.catalysis.org/

Checkland, P, B. (1981), *Systems Thinking, Systems Practice*. Wiley. Chichester.

Checkland, P and Scholes, J. (1990), oft Systems Methodology in Action, Wiley, Chichester.

Checkland, P. (1991), *From Framework through Experience to Learning: the essential nature of action research.* In: Nissen, H. -E., Lein, H. K., & Hirscheim, R., editors.

Information Systems Research: *Contemporary Approaches and Emergent Traditions*. Elsevier Science Publishers, North Holland.

Chen, P. (1976), "The Entity Relationship Model – Towards a Unified View of Data", *ACM Transactions on Database Systems*, 9(4), Dec. 1984, pp.616-645.

Coad, P. and Yourdon, E. (1991), *Object Oriented Analysis*, Second Edition, Yourdon Press.

Codd, E. F. (1970), "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM*, 13(6), pp.377-387.

Codd, E. F. (1979), "Extending the Database Relational Model to Capture More Meaning", *Communications of the ACM*, 4(4), pp.397-434.

Coulson-Thomas, C., and Coe, T. (1991), *The flat Organisation: Philosophy and Practice*, British Institute of Management Report.

Cronk, M. C. & Fitzgerald, E. P. (1997), "A Conceptual Framework for Furthering Understanding of IT Business Value and its Dimension", *Proceedings of the Pacific Asia Conference on Information Systems (PACIS '97)*, 1-5 April, Brisbane, pp.405-16.

Cronk, M. C. & Fitzgerald, E. P. (1999), "Understanding IS Business Value: Derivation of Dimensions", *Journal of Logistics and Information Management*, vol. 12, no. 1.

Crowston, Kevin. (1989), "Towards A Co-ordination Cookbook: Recipes For Multi-Agent Action", PhD Thesis, Massachusetts Institute Of Technology.

Crowston, Kevin (1994), "A Taxonomy of Organisational Dependencies and Co-ordination Mechanisms", MIT Technical Report 174, 3718-94, Available Online: http://ccs.mit.edu/papers/CCSWP174.html.

Davis, William. S. (1995), *Management Information and Systems, An Introduction To Business Information Systems*, West Publishing Company, St. Paul, MN.

Dawson, S. (1996), *Analysing organisations*, Basingstoke, Macmillan Business.

Debreu, G. (1959), *Theory of value: An axiomatic analysis of economic equilibrium*, New York: Wiley.

Dellarocas, Chrysanthos. Nicholas. (1996), "A Co-ordination Perspective on Software Architecture: Towards a Design Handbook for Integrating Software Components", PhD Thesis, Massachusetts Institute of Technology.

Dichter, S. F. (1991), The Organisation of the '90s', *The McKinsey Quarterly*, No.1, pp.145-155.

Dittrich, J. (1991), Groupware Report: "Requirements, State of the art and analysis of group support systems". Technical Report (GMD Focus Berlin).

Dopson, S. and Stewart, R. (1993), "Information Technology, Organisational Restructuring and the Future of Middle Management", *New Technology, Work and Employment*, Vol. 8, No.1, pp.10-20.

Drucker, P. F. (1988), "The Coming of The New Organisation", *Harvard Business Review*, January-February, Vol. 66, No. 1, pp.45-53.

Drucker, P. (1991), "The New Productivity Challenge", *Harvard Business Review*, November-December, Vol. 69, Iss. 6.

D'Souza, Desmond F. and Alan C. Wills. (1998), *Objects, Components and Frameworks with UML: The Catalysis Approach (OBT)*, Addison, Wesley Paperback - 9 December.

Elmasri, R., Hevner, A. and Weeldreyer, J. (1985), "The category concept: An extension to the entity-relationship model", *Data & Knowledge Engineering*, 1 (1), pp75-116.

Emery, J. C. (1969), *Organisational Planning and Control Systems: Theory and Technology*, New York: Macmillan.

Flowers, S. (1996), *Software failure: management failure.* Chichester, UK: John Wiley.

Fox, M. S. (1981), An organisational view of distributed systems, *IEEE Transactions on Systems, Man and Cybernetics*, 11(1), pp.70-79.

Galbraith, J. R. (1977), *Organisation Design*, Reading, MA: Addison-Wesley.

Galliers, R. D. (1985), "In search of a paradigm for information systems research", In: Mumford, E. Hirscheim, R., Fitgerald, G., & Wood-Harper, A. T., editors, Research methods in Information Systems, *Proceedings of the IFIP WG 8.2 Colloquim*, 1-3 September, 1984, Manchester Business School, Elsevier, Amersterdam.

Galliers, R. D. (1992), *Choosing Information Systems research Approaches*, In Galliers, R. D., editor, Information Systems Research: issues, methods and practical guidelines, Blackwell Scientific, Oxford.

Galliers, R. D. and Land, F. F. (1987), "Choosing Appropriate Information Systems Research Methodologies", *Communications Of The ACM*, 30 (11): pp.900-902.

Gayialis, Sotiris P. Tatsiopoulos, Ilias P. (2004), "Design of an IT-driven decision support system for vehicle routing and scheduling", *European Journal of Operational Research,* Volume 152, Issue 2, 16 January, Pages 382-398.

Georgakopoulos, D., Schuster, H., Cichocki, A. and Baker, D. (1999), "Managing process and service fusion in virtual enterprises", *Information Systems*: Special Issue on Information Systems Support for Electronic Commerce, Vol. 24, No.6, pp.429-456.

Glazer, R. (1993), "Measuring the Value of Information: The Information-Intensive Organisation". *IBM Systems Journal*. Vol.32, No.1, pp.99 - 110.

Golder, P. A. (1997), "CORD an Object Model Supporting Statistical Summary Information for Management Decision Making", PhD Thesis, The University of Aston.

Gomes, Rui. Figueiredo, Antonio, Dias, de. (1999), "Types of Business Objects Used to Model an Information Architecture During Information System Planning", in *Managing IT Resources In Organisations, IRMA International Conference.*

Gorry, Antony, G. and Scott-Morton, Michael. (1989), "A Framework for Management Information Systems", *Sloan Management Review*, Vol. 30, No.3, pp.49-61.

Guba, E. & Lincoln, Y. (1989), *Fourth Generation Evaluation*, London: Sage.

Guynes, C. S. Vanceek, M. T. (1996), "Critical Success Factors in Data Management", *Information and Management*, pp.201-209.

Hall, Roger. R. (2001), "Prototyping For Usability Of New Technology", *International Journal Of Human Computer Studies*, Vol 55, pp.485-501.

Handy, C. (1993), *Understanding Organisations*, 4th Edition, Penguin, London.

Handy, C. (1995), *The Age of Unreason*, 2nd edition. Arrow Business Books.

Harris, Andrew. (2000), "Hiring Middle Management: External Recruitment or Internal Promotion", *Canadian HR Reporter*, Toronto; Apr 10.

Hattrup, G. (1993), "How to Establish the Proper Span of Control for Managers", *Total Quality Management Journal*, Nov/Dec, pp.26-29.

Hayashi, Naoki. and Herman, George. (2002), "A Co-ordination-Theory Approach to Exploiting Process Alternatives for Designing Differentiated Products", MIT Technical Report: 218, 4250-02.

Hayashi, Naoki and Herman, George. (2003), "A Coordination-Theory Approach to Exploring Process Alternatives for Designing Differentiated Products", MIT Sloan School of Management, MIT Technical Report: 218, 4250-02, Available Online: http://ccs.mit.edu/papers/pdf/wp218.pdf

Hewitt, C. (1986), "Offices are open systems", *ACM Transactions on Office Systems*, 4(3), pp.271-287.

Hicks, James. O. Jr. (1995), *Management Information Systems*, West Publishing Company. Pub.

Huberman, B. A. (Eds), (1988), *The Ecology of Computation*, Amsterdam: North Holland.

Iivari, J. (1991), "A Pragmatic Analysis of Contemporary Schools of IS Development", *European Journal of Information Systems*, (1:1), 1991, pp.249-272.

Jacobson, I. (1996), *The Object Advantage – Business Process Reengineering with Object Technology*, Addison-Wesley Publishing Company.

Jaques, E. (1989), "In Praise of Hierarchy", *Harvard Business Review*, pp.127-133.

Kanellis, Panagiotis. Lycett, Mark. and J, Paul. (1999), "Failure, Identity Loss and Living Information Systems", *Information Systems Concepts (ISCO)*, 1999, pp.189-212.

Kaye, D. (1995), "The Importance of Information", *Management Decision*, Vol.33, No.5, pp.5 -12.

Kettley, P. (1995), *Is Flatter Better? Delayering The Management Hierarchy*, The Institute For Employment Studies, Report 290, pp.1-61.

Kim, Gyeung-min, J., Myung, Rohae. and Emdad, Ali, F. (2001), "A simulation study on co-ordination strategies: decision cycle-time perspective", *The Journal of Systems and Software*, 57, pp.61-72.

Kim, Hee-Woong. (2000), "Business Process Versus Co-ordination Process in Organisational Change", *The International Journal of Flexible Manufacturing Systems*, 12 (2000), pp.275-290.

Kingdon, D. R. (1973), *Matrix Organisation*, London, Tavistock.

Knight, A. V. and Silk, D. J. (1990), *Managing Information: Information Systems for Today's General Manager*, McGraw Hill Book Company (UK) Ltd, London.

Koh, J. and Kim, Y. G. (2003), "Knowledge sharing in virtual communities: an e-business perspective", *Expert Systems with Applications*, In Press, Corrected Proof, Available online 14 August.

Laudon, Kenneth. C. and Laudon, Jane. Price. (1996), *Management Information Systems*, Fourth Edition, Prentice Hall (Upper Saddle River, N.J.).

Laudon, Kenneth. C. and Laudon, Jane. Price. (1998), *Management Information Systems*, Fifth Edition, Prentice Hall (Upper Saddle River, N.J.).

Lhotka, Rockford. (1998), *Visual Basic 6 Business Objects: Enterprise Design and Implementation*, Wrox Press Ltd, Birmingham, UK.

Lucey, Terry. (1995), *Management Information Systems*. 7th edition, DP Publications, London.

Ludwig, H. and Wittingham, K. (1999), "Virtual Enterprise Coordinator–Agreement-Driven Gateways for Cross-Organisational Workflow Management", in *Procs of the International Conference on Work Activities Co-ordination and Collaboration (WACC'99)*, Feb 22-25, San Francisco, CA, USA.

Lunn, K. (2002), *Software Development With UML*, Palgrave Macmillan, Paperback - 13 December.

Lyytinen, K. and Hirscheim, R. (1987), "Information failures - a survey and classification of the empirical literature", *Oxford Surveys in Information Technology*, 4, pp.257-309.

Malone, T.W. and Crowston, K. (1990), "What is co-ordination theory and how can it help design cooperative work systems?", *Conference on Computer-Supported Cooperative Work*, April, 112, 3402/3183.

Malone, Thomas. W. and Crowston, K. (1994), "The Interdisciplinary Study of Co-ordination", *ACM Computing Surveys*, 26(1), pp.87-119.

Malone, Thomas. W. et al, (1999), "Tools for Inventing Organisations: Toward a Handbook of Organisational Processes", *Management Science,* March, 45(3) pp 425-443.

Malone, T. W., Crowston, K. G., & Herman, G. (2003), *Organizing Business Knowledge: The MIT Process Handbook*. Cambridge, MA: MIT Press.

Malone, Thomas. W. and Laubacher, Robert. (2003), "Temporary assignments and a permanent home: A case study in the transition to project-based organizational practices", MIT Sloan School of Management, MIT Technical Report: 220, 4323-02, December, Available Online: http://ccs.mit.edu/papers/pdf/wp220.pdf

March, J.G., Simon, H.A. (1958). *ORGANIZATIONS*, New York, NY: Wiley.

Maturana, Sergio., Ferrer, Juan-Carlos and Barañao, Francisco. (2003), "Design and implementation of an optimization-based decision support system generator", *European Journal of Operational Research*, In Press, Corrected Proof, Available online 22 January.

McCann, J. E. & Ferry, D. L. (1979), An approach for assessing and managing inter-unit interdependence. Academy of Management Review, 4(1): pp.113-119.

McCarthy, Brendan. (2002), General Manager, BMW Hams Hall, Interview conducted 14th September 2000.

McConnel, S. (1997), The OMG/CommerceNet Joint Electronic Commerce White-Paper, EC/97-06-09, Object Management Group, July.

McLean, E. R. and Soden, J.V. (1977), *Strategic Planning for MIS*, Wiley-Interscience, New York.

Mentzas, Gregory. N. (1993), "Co-ordination of joint tasks in organisational processes", *Journal of Information Technology*, 8, 139-150.

Miller, M. S. and Drexler, K. E. (1988), "*Markets and computation: Agoric open systems*", In B. A. Huberman (Ed.), "*The Ecology of Computation*" (pp.133-176), Amsterdam: North-Holland.

Mintzberg, H. (1979), *The Structuring Of Organisations*, Third Edition, Prentice-Hall, Englewood Cliffs, N.J.

Mintzberg, H. and Quinn, J, (1988), *The Strategy Process: Concepts, Contexts and Cases*, Third Edition, Prentice-Hall, Englewood Cliffs, N.J.

Morden, Tony. (1993), *Business Strategy and Planning, Text and Cases,* McGraw-Hill International (UK) Limited.

Nonaka, I. (1991), "The Knowledge-Creating Company", *Harvard Business Review*, November-December, pp.96-104.

Norfolk, D. (1997), "Data Management", *Information Age*, June, pp.44-53.

Pan, Shan L. and Leidner, Dorothy E. (2003), "Bridging communities of practice with information technology in pursuit of global knowledge sharing", *The Journal of Strategic Information Systems,* Volume 12, Issue 1, March, Pages 71-88.

Papazoglou, M., Riebbers, P. and Tsalgatidou, A. (2000), "Integrated Value Chains and their implications from a business and technology standpoint", Decision Support Systems Journal, Vol. 29, No. 4, pp.323-342.

Parsons, G.L. (1983), "*Information Technology: A New Competitive Weapon,*" *Sloan Management Review*, pp.3-14.

Partridge, C. (1996), *Business Objects Re-engineering for RE-use*, Butterworth Heinemann, Oxford.

Pennings, J. (1974), "Differentiation, Interdependence and Performance in Formal Organizations". *Paper presented at American Sociological Association Annual Conference*, Montreal, Carnegie-Mellon University.

Porter, M. E. (1985), *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: Free Press.

Ramackers, G. (1996), "BPR with extended use cases and business objects", *in Object World '96 UK Conference*, Queen Elizabeth Conference Centre, London.

Rational Rose (2003), Rational Rose [Online] Available: http://www.rational.com/products/rup/index.jsp?SMSESSION=NO

Reynolds, George. W. (1995), *Information Systems for Managers*, West Publishing Company. Pub.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991), *Object Oriented Modelling and Design*, Prentice/Hall International.

Sauer, C. (1993), *Why information systems fail: a case study approach, information systems series*. Henley-on-Thames, UK: Alfred Waller.

Scheer, A. W. Habermann, F. (2000), "Making ERP Success". *Communications of the ACM,* Vol.43, No.4, April, pp.57-61.

Schoderbek, P. P., Schoderbek, C. G. and Kefalas, A. G. (1990), *Management Systems. Conceptual Considerations*, Fourth Edition, Richard D Irwin Inc, USA.

Senn. A, (1990), *Information Systems in Management*, 4th Edition,Wadsworth Publishing Company, California.

Shapiro, Benson. P. (1977), "Can Marketing and Manufacturing Coexist?", *Harvard Business Review*, Vol 55, pp.104-114.

Singh, B. (1992), "Interconnected Roles (IR): A Co-ordinated model", Tech, Rep. CT-84-92, Microelectronics and Computer Technology Corp, Austin Tex.

Smith, R. G. and Davis, R. (1981), "Frameworks for co-operation in distributed problem solving", IEEE Transactions on Systems, Man and Cybernetics, 11(1), pp.61-70.

Stamelos, Ioannis., Angelis, Lefteris., Morisio, Maurizio., Sakellaris, Evaggelos and Bleris, George. L. (2003), "Estimating the development cost of custom software", *Information & Management, Volume 40, Issue 8, September, Pages 729-741.*

Stefik, M. and Bobrow, D. G. (1986), "Object-oriented programming: Themes and variations", *AI Magazine*, *6,* 4 (Winter), pp.40-62.

Sumner, Mary. (2000), "Risk factors in enterprise-wide/ERP projects," *Journal of Information Technology*, Vol.15, No.4, pp.317-327.

Susman, G. (1983), *Action Research: a sociotehnical system perspective.* In: Morgan, G., editor, *Beyond Method: strategies for social research*, Sage, Newbury Park, CA.

Sutherland, J. (1995), "The Object Technology Architecture: Business Objects For Corporate Information Systems", Symposium for VMARK Users, Albuquerque, USA, Source: http://www.itac.net/users/jsuth.

Swash, G. (1998), UK business on the internet. *New Library World*99, 1144, pp.238-242.

Swenson, Keith. (1998), "Simple Workflow Access Protocol", 7/16/98,
Available Online: http://www.isr.uci.edu/events/twist/wisen98/presentations/Swenson/

Tanniru, M. R. and Jain, H. K. (1989), "Knowledge-base GDSS to support reciprocally interdependent decisions", *Decision Support Systems,* Vol. 5, No. 3 pp.287-301.

Thomas, J. (1957), "Effects of facilitative role interdependence on group functioning", *Human Relations*, 19: pp.347-366.

Thomas, G. H. and Langemo, M. (1992), "A New Look at Information Systems and Records Management", *Arma Records Management Quarterly*, Vol. 26, No. 3, July, pp.3-4, 6-7.

Thompson, J. D. (1967), *Organisations in Action*. McGraw Hill. New York.

Victor, B. & Blackburn, R. S. (1987), "Interdependence: An alternative conceptualization", *Academy of Management Review*, 12(3), pp.486-498.

Visual Modeller, (2003), [Online]: http://msdn.microsoft.com/vstudio/

Van den Bogaard, Michel. A. Speklé, Roland F. (2003), "Reinventing the hierarchy: strategy and control in the Shell Chemicals carve-out", *Management Accounting Research,* Volume 14, Issue 2, June, pp.79-93.

Von Bertalanffy, L. (1950), The theory of open systems in physics and biology. *Science*, 111.

von Martial, F. (1989), "Multiagent plan relationships", *In Proceedings of the Ninth Workshop on Distributed Artificial Intelligence*, Rosario Resort, Eastsound, WA, pp.59-72.

Walsham, G. (1993), *Interpreting Information Systems in Organisations*, Wiley, UK.

Weber, Max. (1922), *Wirtschaft und Gesellschaft*, Tuebingen, J.C.B. Mohr, part III, chap. 6, pp.650-78.

Wegner, P. (1987), "Dimensions of object-based language design", In *Proceedings of the Conference on Object-Oriented Systems, Languages, and Applications (OOPSLA '87)*, pp.168-182, Orlando, Fla.: ACM.

West, L. A. and Courtney, J. F. (1993), "The Information Problem in Organizations: A Research Model for the Value of Information and Information Systems," *Decision Sciences*, Vol. 24, No. 2, March/April, pp.229-251.

Wetherbe, J. C. (1994), *Systems Analysis and Design*, West Publishing, St. Paul, MN.

Whang, Seungjin (1995), "Co-ordination In Operations: A Taxonomy", *Journal Of Operations Management*, 12, p. 413-422.

Wheatley, M. (1992), *The Future of Middle Management*. London: British Institute of Management.

Wiener, N. (1948), *Cybernetics or Control and Communication in the Animal and the Machine*, Wiley, New York.

Wiener, N. (1954), *The Human Use of Human Beings,* Boston, MA: Houghton-Mifflin.
Williamson, O. E. (1975), *Markets and Hierarchies*, Free Press, New York.

Wilson, A. David. (1993), *Managing Information. Managing Information.* Institute of Management. Butterworth-Heinemann. Oxford.

Workflow Analyzer. (2001), Avalable: http://www.metasoftware.com/

Workflow Management Facility. (1998), Revised Submission, OMG Document Number bom/98-06-07, Object Management Group, July.

Wu, Jen-Her., Chen, Yi-Cheng., and Lin, Hsin-Hui. (2003), "Developing a set of management needs for IS managers: a study of necessary managerial activities and skills", *Information & Management,* In Press, Corrected Proof, Available online 12 September 2003.

Wyner, George. M. Lee, Jintae (2003), "Defining Specialization for Process Models", MIT Sloan School of Management, MIT Technical Report: 216, 4159, Available Online: http://ccs.mit.edu/papers/pdf/wp216.pdf

Yeo, K. T. (2002), "Critical failure factors in information systems projects", *International Journal of Project Management*, 20, pp.241-246.

Yoshioka, Takeshi. and Herman, George. (1999), "Genre Taxonomy: A knowledge Repository of Communicative Actions", MIT Tech Report: 209, 4093, Available Online: http://ccs.mit.edu/wp_toc.html

Zeffane, R. (1992), "Organisational Structures: Design in the Nineties", *Leadership and Organisational Development Journal*, Vol. 13, No. 6, pp.18-23.

Zlotkin, G. (1995), "Co-ordinating Resource Based Dependencies", *MIT Center for Co-ordination Science Unpublished working paper*, March 1995.

Zlotkin, G. and Wyner, G. (1995), Resource Use and Co-ordination Presentation given at the MIT Center for Co-ordination Science, April 28, 1995.

APPENDIX A

CBOSS Program Listing

A.1. INTRODUCTION

For brevity, the CBOSS program files can be browsed on the CD-ROM attached to this thesis. It should be noted that the application is not bug free, but was sufficient enough to be used in a testing environment. It is advised however, that the latest up to date files should be downloaded from the following URL:

http://www.cbom.pwp.blueyonder.co.uk

APPENDIX B

CBOSS Menus and Other Screen Displays

Pg 257 - 292