USING STRUCTURED ANALYSIS AND DESIGN TECHNIQUE (SADT) FOR SIMULATION CONCEPTUAL MODELLING

By

Fahim Ahmed

A Doctoral Thesis

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

August 2015

Dedicated to my parents for their unconditional love and prayers

ACKNOWLEDGEMENTS

First of all, thanks to GOD for giving me the strength and ability to complete this thesis. Next, I would like to express my gratitude to my parents whose moral support made this journey possible.

I am extremely grateful to my supervisors Professor Stewart Robinson and Dr. Antuela Tako for giving me the professional and personal support whenever I required. Without their feedback and encouragement, it would have been impossible to complete this research.

I would like to thank my colleagues at Loughborough University who helped in piloting and testing of the frameworks during this research. I am also thankful to all the administrative staff at Loughborough Business School who was extremely helpful in providing their support. I am thankful to Professor Jyin liu, Dave coates, Alan French, Malcolm King and Rupal Rana for their assistance in the teaching modules. I am also grateful to Dr. Ryan Leduc (McMaster University, Canada) and Professor Bengt Lennartson (Chalmers University, Sweden) for their help and support during early phases of my postgraduate career. I want to acknowledge the support of all my friends and family during this research, most of whom are living far and away.

Last, but not the least, I would like to thank my wife for her continuous support throughout this journey, and standing by me during the difficult moments of this research.

ABSTRACT

Conceptual Modelling (CM) has received little attention in the area of Modelling and Simulation (M&S) and more specifically in Discrete Event Simulation (DES). It is widely agreed that CM is least understood despite its importance. This is however, not the case in other fields of science and engineering (especially, computer science, systems engineering and software engineering). In Computer Science (CS) alone, CM has been extensively used for requirements specification and some well-established methods are in practice. The aim of the thesis is to propose a CM framework based on the principles of software engineering and CS. The development of the framework is adapted from a well-known software engineering method called Structured Analysis and Design Technique (SADT), hence it is called SADT CM. It is argued that by adapting approaches from CS, similar benefits can be achieved in terms of formality, understanding, communication and quality. A comprehensive cross-disciplinary review of CM in CS and M&S is undertaken, which highlights the dearth of standards within M&S CM when compared to CS. Three important sub-fields of CS are considered for this purpose namely, information systems, databases and software engineering. The review identifies two potential methods that could be adopted for developing a M&S CM framework. The first method called PREView was found unsuitable for M&S CM in DES domain. Hence, the thesis concentrates on developing the framework based on SADT. The SADT CM framework is evaluated on three-in depth test cases that investigate the feasibility of the approach. The study also contributes to the literature by conducting a usability test of the CM framework in an experimental setting. A comprehensive user-guide has also been developed as part of the research for users to follow the framework.

Table of Contents

1 Introduction

This introductory chapter presents the context of the research undertaken in the area of Conceptual Modelling (CM) for Discrete Event Simulation (DES) during the course of the Ph.D. work. It provides background of the research, objectives, contribution, and outline of the thesis.

1.1 Background of the research

Modelling and Simulation (M&S) has become an important part of organisations' planning and decision making policies in today's world. Some examples include but not limited to enterprise level matters, operations management, cost analysis and resource optimization. An important phase of M&S which has started to gain attention in recent years is CM, and its outcome a conceptual model. One of the major challenges in conducting research in this area is due to the lack of consensus on the perception of conceptual models. Although it is now widely agreed that CM is the most important part of simulation study; it is still least understood and there is no agreement on the processes that define CM (Law, 1991; Robinson, 2008a; Robinson, 2008b). After consulting the literature on CM within M&S, it is concluded that the most accepted view about conceptual model in the simulation community is from Robinson's definition which is adopted in this thesis and stated here:

"The conceptual model is a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model."

(Robinson, 2008a)

The uncertainty and lack of consensus in the area of CM within M&S has inspired this research. There exist no procedures and/or literature that has given due attention to this area despite its agreed importance. In this respect, among the disciplines in science and engineering, M&S stands alone as the only sub-field with no established CM procedure. The need to consult other disciplines for their perspectives on CM is equally important so as to highlight any approaches that could be adopted for CM within M&S. The need to develop CM in M&S provides the rationale of this research.

1.2 Rationale of the research

Although the definition of conceptual model given in Robinson (2008a) is followed and adopted in most literature, there is a strong lack of rigorous and concise approaches to develop such a conceptual model. There are two possible reasons for that:

- Firstly, CM is placed on the creative or even artistic side of simulation projects (Robinson, 2013), so the need for structured (or rigorous) approaches has never been seen as important or even possible.
- There is very little communication on this subject between M&S and other disciplines (computer science, systems engineering, control engineering, etc.) (Arthur and Nance, 2007; Ahmed et al., 2014), which has led M&S into methodological isolation for CM unlike other communities.

It is true that CM is highly influenced by creativity, intuition and art but this does not justify less research in structured approaches for CM. This is because, to completely get the benefits from a conceptual model as stated in Robinson's definition, there is a need for precise and structured methods to develop such conceptual models. These structured methods may not need to be developed from scratch, since there are various disciplines (especially, computer science, systems engineering and software engineering) that perform CM with structured methodologies that might be adopted within M&S. Also, despite the differences in the perception of a conceptual model among each field, the fact is, there is much to learn from the overlapping elements of these fields and a joint venture is necessary to highlight the importance of such adoption for CM within M&S. This learning curve should start from a deep comparative study within these areas and feasible approaches for adoption should be highlighted to benefit CM within M&S. It is also true that whatever perception these communities hold of CM, they have given due attention to this area realizing its benefits. If different perceptions are studied, the closest to M&S can be identified and the practices can be adopted with some adaptation.

This thesis advocates the need for adoption of structured approaches from Computer Science (CS). The reason for selecting CS is the level of commonalities among CS and M&S; especially when it comes to the early phases of the projects in both areas. However, this claim itself is one of the major findings presented in this research as there is no comprehensive study before to address this gap. More detailed discussion is presented in chapter 2. Also, it is conclusive through the review of literature that CM is more researched in CS than in M&S, as it is discussed in chapter 2.The paucity of research in CM within M&S is a major concern especially when there exists similarities between the activities in the early stages of projects within M&S and CS. The scope of this research is DES. The idea is to explore the practices of CM in DES as this mode of simulation is more in-line with the structured methods from CS that are used for requirements specification process.

1.3 Research Objectives

The aim of the research in this thesis can be stated as follows:

"To explore how CM approaches in CS could be adopted and adapted for M&S CM"

In the light of this objective, the main research questions are as follows:

- ***RQ1: Which part of CS is most appropriate for adaptation to M&S CM?***
- ***RQ2: Which methods from CS could best be adopted for M&S CM?***
- ***RQ3: How should these methods be adapted for M&S CM?***

In the following, we briefly present the research objective related to each research question above:

- ***RQ1: Which part of CS is most appropriate for adaptation to M&S CM?***

This research question is related to the objective of conducting a comprehensive cross-disciplinary review between CS and M&S on the topic of CM. This review is presented in Chapter 2. The review is analysed and the key results are summed up to decide the most relevant field for adaptation of methods within CM for M&S. The study is undertaken with respect to the four major components of a conceptual model, definition of CM, purpose of CM, CM process, and conceptual model testing. These dimensions help to identify the part of CS most appropriate for adaptation to M&S CM.

- ***RQ2: Which methods from CS could best be adopted for M&S CM?***

The objective of this research question is to identify particular methods from CS that could best be adopted for M&S CM. The justification of selecting particular methods from CS for adopting into M&S CM is expected to follow from the review between CS and M&S on the topic of CM (Chapter 2). The detailed discussion on selection of the methods from CS is given in research methodology in section 3.2.

- ***RQ3: How should these methods be adapted for M&S CM?***

The objective for this research question is to test the feasibility and usability of the developed framework. The methodology for evaluating the adapted M&S CM framework is detailed in chapter 3 (section 3.3).

1.4 Related work and main contribution

During the last decade some attention was given to the area of CM by the research community. A separate stream was organized at the Operational Research Society Simulation Workshop in 2006 (SW06) for CM that represents the highest number of papers on this topic (Robinson, 2008b). A CM day was organized after the end of Operational Research Society Simulation Workshop in (2006 (SW6), 2008 (SW8), 2010 (SW10) and 2012 (SW12) at which various academics and practitioners were gathered to give their thoughts on this topic. The only full length book on this topic was published in 2011 (Robinson et al., 2011) where authors have discussed CM frameworks, objectives and implementation from a range of perspectives. The only other book which dedicates a handful of pages to this topic is from Robinson (2014). Some other works (Kotiadis et al., 2013; van der Zee and van der Vorst, 2007; Montevechi and Friend, 2012; Balci and Ormsby, 2007; Robinson, 2008b) are either related to specific domains or mostly descriptive in nature.

There are some important works in the area of military simulation which discusses the use of simulation conceptual model and some essential standards have been established to assess the quality and interoperability among different conceptual models. Simulation Interoperability Standards Organization (SISO) was started in 1996 with the cooperation of US Defense Modelling and Simulation Office (DMSO). Previously, it started with a small conference with the aim of developing standards among network simulation and to develop means for exchange of information between different technological groups, however, it soon transformed into Distributed Interactive Simulation (DIS) and then finally became known as SISO (https://www.sisostds.org, assessed Dec, 2015). SISO/DIS was involved in developing IEEE standards for DIS (IEEE 1278) and Higher Level Architecture (HLA) IEEE 1516 (https://www.sisostds.org, assessed Dec, 2015). However, HLA [available on DMSO website: http://www.msco.mil/] focused mainly on interoperability among distributed simulations within US defense community, europe and elsewhere, but similar standard approaches were not developed for documenting a simulation conceptual model (Pace, 2000; Tolk and Muguira, 2003). Within the military domain, there are various other platforms which have contributed towards developing standardized procedures towards interoperability of the simulation models. These include Simulation Interoperability Workshops (SIWs), Society for Computer Simulation (SCS) conferences and Military Operations Research Society (MORS). In November, 1999, under Department of Defense (DoD), Joint Technical Architecture (JTA) selected Integrated Computer Aided Manufacturing Definition (IDEF) as standards for data and process descriptions (Pace, 2000). However, as described in In Tolk and Muguira (2003), these standardizations were limited to implementation level . They have argued that most of the above mentioned standards have not targeted the composablity at the cocneptual levels and were mainly concerned with issues of data aggregation , object models etc. Tolk and Muguira (2003) propose a layered model called Levels of Conceptual Interoperability Model (LCIM) to bridge the gap between implementation methods and conceptual models. The primary focus of this work is the military domain which specifically deals with C4ISR (command,control,communications, computers,intelligence, surveillance, and reconnaissance) .

This thesis aims to contribute in the literature on the topic of M&S CM by comparing different approaches and perspectives on CM from CS and by providing an insight to the dearth of standards (if any) within M&S CM. The three main sub-fields of CS studied for this purpose are Information Systems (IS), Databases (DB) and Software Engineering (SE).

The main contribution of this thesis is the development of a CM framework through adaptation of a well-known SE method, called SADT (Structured Analysis and Design

Technique). Although it is argued that SE has closest potential to be adapted for M&S CM framework, but the review suggests that some well-established standards, objectives and frameworks for CM already exist in all the sub-fields of CS, details of which are reported in chapter 2. CM is widely used in SE during the early phases of systems development to gather and communicate domain information for use in the design phase. It has been proposed in this research that rather than *reinventing the wheel*, the integration of methods from these two fields would allow M&S to benefit from the CS methods. In recent years, a few studies have suggested such an adaptation of CS methods within (conceptual) modelling for M&S (Arthur and Nance, 2007; Turnitsa et al., 2010; Tolk and Turnitsa, 2012; Guizzardi and Wagner, 2012; Tolk et al., 2013), however, a step-by-step guidance or framework that could be useful for other modellers (especially less experienced ones) remains largely unaddressed.

1.5 Thesis outline

This thesis is structured as follows:

- **Chapter one: Introduction**

This introductory chapter presents the background and rationale of the research. Research objectives and research questions are also presented. The chapter lists some related work and the main contributions expected from this thesis.

- **Chapter two: Conceptual Modelling in Computer Science and M&S**

This chapter provides a cross-disciplinary review between computer science and M&S on the topic of CM. The three sub-fields of computer science are discussed: information systems, databases and software engineering. This comparison has been made with respect to four components of CM: definition, purpose, process and testing.

- **Chapter three: Research Methodology**

This chapter describes the research methodology for each stage of this research. An overview of the methodology is provided with respect to the research questions identified in this chapter.

- **Chapter four: PREmod: framework for CM**

This chapter presents a CM framework developed in this research called PREmod which is inspired from software requirements engineering method, PREView. The adaptation of the method is explained with simple examples. Application of the framework on semi-real test cases is provided along with pilot testing. Results from the testing are reported and any issues with the framework are also discussed. The PREmod framework was not successful in tackling all aspects of CM but it provided useful lessons in adaptation of CS methods into M&S CM. The detailed discussion is presented in chapter 4.

- **Chapter five: SADT CM: framework for CM**

This chapter describes the framework for CM adapted from the software requirements engineering method SADT (Structured Analysis and Design Technique). The framework is called SADT CM. Different steps of the framework are explained in detail along with step-by step guidelines for modellers to use the framework. The application of the framework on a TV manufacturing case study is provided in the chapter and results are discussed.

- **Chapter six: Testing of SADT CM**

This chapter reports on the detailed testing of the CM framework SADT CM presented in chapter 5. The testing procedures involved pilot testing, modifications, final test and analysis of the results for the framework.

- **Chapter seven: Discussion and Future work**

This chapter summarises the research presented in this thesis along with a brief discussion about the research outcomes, limitations of the study and identification of future directions for the research.

2 Conceptual Modelling in Computer Science and M&S

The term *Conceptual Modelling* (CM) has now been widely used in many areas of science and management. The interpretation and framework adopted by all such communities, however, differs from each other. In this *review*, we aim to collect different perspectives that exist currently to support the idea of CM with reference to Computer Science (CS) and M&S disciplines. In CS, CM is a pre-development phase for systems design when the requirements and objectives are being understood. However, within CS, there exist well-defined objectives, frameworks and quality evaluation methods for CM, and these have been in use for many decades. In this chapter, we present a cross-disciplinary review of CM within the major sub-fields of CS (Information Systems, Software Engineering and Databases) and make a comparison with M&S. The main aim of this work is to highlight the need for a well-defined process for CM within M&S along the lines of that in CS. The objective of this review is as follows: to perform a comparative study of CM in the fields of CS and M&S, thus highlighting the dearth of standards within M&S CM when compared with CS. This involves a comparison of four components of CM across four fields:

- Definition of CM
- Purpose of CM
- CM Process
- Conceptual Model Testing

Our discussion of these four components of a conceptual model is based around certain characteristics which forms the basis of comparison for this study. These characteristics can broadly be assigned to these four components as follows:

- Definition of CM: level of formality, scope
- Purpose of CM: understanding, communication, abstraction
- CM Process: theoretical basis, formal methods, modelling languages
- Conceptual Model Testing: quality standards and criteria

The chapter concludes by summarising the comparison of CM across the fields and outlining the next phase of the research. Individual summaries of the four characteristics across each sub-field of CS and M&S are also reported in subsequent sections.

2.1 Conceptual modelling in Computer Science (CS)

Within CS, CM can be used for multiple purposes from an informal diagram in the early phases of a project to a very formal and machine readable document. In this section, we discuss CM within three major fields of CS that are defined below:

- Information Systems (IS)

IS represents a major sub-field within CS and deals with the management of information during system development. This can include the study of information flow for the design of IS in terms of hardware, software, people or enterprise (Datri et al., 2008).

- Databases (DB)

DB are primarily used to store and retrieve the information within an organisation's information management system; however, a DB system is also proven to have an effect on the long-term goals and other aspects of corporate management such as processes, customers, suppliers etc.(Mineau et al., 2000).

- Software Engineering (SE)

SE deals with the development of a software system as a sub-component of a large CS project or standalone application. SE in particular is a discipline that deals with development, documentation and maintenance of software systems (Thayer and Dorfman, 1997).

In recent years, SE has gained the status of a major sub-field within CS due to the rapid increase in the development of complex software systems.

2.1.1 Information Systems (IS)

In this section, we present different perspectives on CM within IS according to the four characteristics: definition of CM, purpose of CM, CM processes and testing of conceptual model.

2.1.1.1 IS: Definition of CM

According to Mylopoulos (1992), "conceptual modelling is the activity of formally describing some aspects of the physical and social world around us for the purpose of understanding and communication". This definition also agrees with the view of Siau (2004), who defines CM as a process of formally documenting a problem domain for the purpose of understanding and communication among stakeholders. Shanks et al. (2003) describe conceptual model to be a representation (typically graphical) of an individual or a group's perception of a real world domain. This also follows from the definition of conceptual model by Wand and Weber (1990), that conceptual model represents some aspects of the real world as perceived by a human.

In Summary, a consensus exists on some aspects of the CM definition. Mainly, CM is agreed to be a formal process of describing and documenting a problem domain (physical and social world), for understanding and communication purposes. The important point to notice here is the emphasis on a "formal" procedure that completes the definition of a CM. Naturally, it can only be the outcome of a formal process that the product (conceptual model) is communicative and shares some common understanding. Vague and ambiguous models lack the notion of understandability. Also conceptual model is agreed to be an abstraction of the problem domain as it is used to describe certain features relevant to some aspects of the world. The scope of CM is the information about a real world domain.

2.1.1.2 IS: Purpose of CM

Various researchers have presented the purpose of the CM within IS field and highlight the importance of a well-developed conceptual model in the early phases of systems development. Conceptual model has been identified as of extreme importance for the assistance of users and analysts during the whole lifecycle of a project. Also it has been classified as a phase where the problem domain is captured in a way which can serve as a bridge for users and developers of a model. Wand and Weber (1990) sees a conceptual model as an integral part of the requirement analysis for IS development. According to them, conceptual model should serve four purposes: a) communication between developers and users b) helping analysts understand a domain c) provide input for the design process d) documenting original requirements for future reference. Also, according to March and Allen (2009), a conceptual modeller must analyse the domain to identify the important objects and rules, these must be understood in the context of system design and social constraints. Kung and Solvberg (1986) suggest that a conceptual model supports communication between stakeholders and Rolland and Cauvat (1992) have described the purpose of a conceptual model as a bridge between user's requirements and systems design. In the view of Davies et al., (2006), a conceptual model is developed and used during the requirements analysis phase of IS development. Such methods are mostly graphical and they are used to represent both static (e.g. entities) and dynamic phenomena (e.g. processes) in some domain.

From the above discussion, we see an agreement in the research community on the objectives that should be achieved through CM in terms of users' requirements and communication between stakeholders. The documentation and communication of requirements are two other purposes of CM reported in the studies.

2.1.1.3 IS: Process of CM

CM processes in IS can broadly be assigned into two categories: theoretical (based on theories from the social and natural sciences, e.g. ontology, linguistics and cognition) and modelling (based on modelling languages, e.g. Unified Modelling Language (UML) and Entity-Relationship (ER) models). The theoretical aspect is based on creating meaningful representations from problem domains, whilst the modelling aspect is based on evaluating and refining modelling languages to map those representations onto suitable modelling constructs.

A number of researchers have proposed using ontology (the philosophical study of the nature of being, existence, or reality as such, as well as the basic categories of being and their relations) to be used as a theoretical basis for the framework in conceptual modelling. According to March and Allen (2009), CM is a presentation which is measured by the degree to which it facilitates a shared understanding among all stakeholders. The philosophical discipline of ontology provides a substantive basis for such a shared understanding (March and

Allen, 2009). Researchers have been proposing ontology as a basis to formulate and evaluate the CM grammar and methods. One of the most important work that formed the basis of theoretical foundations for conceptual modelling in IS comes from Evermann and Wand (2005), who propose the use of ontology for creating meaningful definitions of the proposed system. Since the model is an abstraction of the real world, they argue that the modelling constructs should be based on the concepts of ontology since it can capture the behaviour and existence of objects and their relationship in more natural way than any other abstract theory. Wand et al. (1995) have also presented theoretical framework based on human knowledge to support the conceptual modelling phase in IS development. They proposed the use of ontology, concept theory and speech act theory (linguistics) as the main models of human knowledge to be used for the purpose of conceptual modelling. They have discussed the main ideas; principles and application of the framework to CM. Classification theory and speech act theory have been proposed for application on conceptual modelling. Speech act theory has been discussed as it deals with the classification of communication acts according to the intention and possible effects. Authors believe that this will help to model the interactive behaviour within the components of a proposed system. In Wand et al. (1995), the authors have favoured adapting the ontology for the CM construct as they must represent the real world phenomenon. The paper is more about the structure of CM and the use of relationships rather than their view of CM itself. On the modelling side, the most significant work is reported by Wand and Weber (2002), who present a set of guidelines/rules for modelling language, grammar and context. A similar framework is proposed by Guizzardi and Wagner (2005), but unlike Wand and Weber do not provide explicit modelling rules. Their work captures different ontological categories relating to natural languages and maps into conceptual modelling languages like ER diagrams or UML class diagrams.

In summary, support for CM processes within IS comes for the most part from a theoretical basis and modelling languages.

2.1.1.4 IS: Testing/Evaluation of Conceptual Model

Quality evaluation and testing of a conceptual model has also gained attention in the IS field. Some research has focused on evaluating the theoretical approaches used for domain presentation, whilst other work has aimed to either evaluate the general quality criteria or provide explicit guidelines for a good quality conceptual model. Also, some frameworks have evaluated the process of CM while others take the view of evaluating the product (conceptual model) from the perspective of quality. Examples include a framework from Wand Weber 1990. Their work is based on Bunge's ontology (Bunge, 1977). Together, this framework is known as BWW (Bunge, Weber and Wand) framework. BWW targets the process oriented view of CM itself for quality evaluation. An alternative framework based on linguistics was proposed by Lindland et al, (1994) and this is known as LSS after Lindland, Sindre and Solvberg. LSS specifically focuses on evaluating the quality of a conceptual model from the presentation perspective (product view). Both frameworks are discussed in some more detail in section 1.1.3.4 where we discuss the quality framework for SE.

Gemino and Wand (2004) have discussed a framework that shows dependence between modelling task, model creation and model interpretation. The framework they propose aims to categorize the empirical work in the area of CM techniques. Their work studies dimensions (effecting and effected variables) and their measurement with respect to CM quality. It also provides some theoretical guidance in evaluating such empirical comparisons. This theoretical guidance is discussed with respect to five elements of model construction: content, presentation, method, individual characteristics and task. One of their claims is that the framework can be used to build a common source of terminology: to compare existing work, to identify where gaps exist, to identify where more work can be done and to support the formation of research hypotheses. Parsons and Cole (2005) have proposed four guidelines for experimental design in evaluating the CM techniques. The guidelines more specifically establish the criteria for the choice of independent variables (alternative forms for presenting the conceptual model), dependant measures (measurement specific to standards used in conceptual model), participants and procedures used in the process of CM. Details of this work can be found in Parsons and Cole (2005). Also, Gemino and Wand (2005) examine the role of mandatory and optional properties in an ER-diagram for high quality CM.

In summary, within IS, a quality framework for CM is mainly supported by a theoretical basis with some general quality guidelines also being available.

2.1.1.5 Summary of CM within IS

In IS, the definition of CM is expressed as a formal document and the scope covers the real world information. Objectives of CM are mostly reported for understanding and communication purpose. The process of CM mainly originates from a theoretical basis (ontology, linguistics) while good frameworks are based on modelling languages such as UML and ER-models. The quality evaluation frameworks mostly have a theoretical basis (ontology and linguistics), while some do define general quality criterion.

2.1.2 CM in Databases (DB)

This section covers the different components of CM within a DB system.

2.1.2.1 DB: Definition of CM

CM is considered as a fundamental phase in DB where knowledge about the universe of discourse is gathered and documented (Battista et al., 1989). In another view, CM is a process of identifying the information content of DB design in an abstract manner using the notion of entities, relationships, attributes and categories (Batra and Davis, 1992). In some studies, a CM is considered as the first formal specification of the DB design (Engels et al., 1992).

From the user's point-of view, a CM is a communication channel with the user which is platform-independent and this helps to achieve a focus towards the properties of the data instead of technical details like data structures and file properties (Batra and Davis, 1992). As identified by Crockett et al. (1991), CM is regarded as an abstraction of the user perception of the DB environment.

Abstraction, sharing and representation of DB information are common perceptions that exist in the DB community with respect to CM.

2.1.2.2 DB: Purpose of CM

According to Mineau et al. (2000), the purpose of CM language in DB is to capture the relevant knowledge in an easy and understandable way, so this information can easily be validated and integrated from higher-level (management) to the lower level (designers) in order to satisfy the business needs of an organization. A better understanding of CM in DB design can overcome the weaknesses in current processes of knowledge based systems, especially in the area of automated DB designs (Batra and Davis, 1992). According to this view, most of research in DB CM finds its basis mainly in creating meaningful representation of data and their comparison. However, it is also agreed that CM would help a modeller with problem-solving along with creative thinking processes within the design of DB (Batra and Davis, 1992).

CM in DB design is used to capture the descriptions of the phenomenon, properties and their interactions as perceived by the users related to the DB system development in UOD (Battista et al., 1989). The conceptual model of the DB produces a documented and structured form of knowledge describing some phenomenon in universe of discourse. Development of a conceptual model is considered an effective approach to manage an organization's information and maintain its integrity in a way that is shareable among all members (Crockett et al., 1991). In this way an effective representation of DB is produced based on the conceptual model which helps to maintain the corporate knowledge and information in a manner that is understandable to the members of organization.

Engels et al. (1992) have regarded the CM as the most important process in a DB design. According to their view, a conceptual model fulfils two aspects both from the customer and designer points of view. At one end, it helps to develop communication with the customer, and at the other end, it helps the designer to accurately proceed with low-level transformations of design using the conceptual model as a base. The role of a conceptual model is to make meaningful, high-level and expressive constructs which are able to capture both static and dynamic aspects in a DB design (covering perspectives of users, designers and application domain).

2.1.2.3 DB: Process of CM

The process of producing a conceptual model of the data is considered a very important activity since there are many aspects to consider. One of the main reasons identified in most studies is to take into account both static and dynamic elements of a DB design. It is good to have a brief overview about these two important terms used very frequently in the literatures for conceptual model in DB. The static component of the DB object deals with the structural part. Mainly it contains information about object identity, attribute, relationship, environment etc. The dynamic component focuses on the transition of a DB object from one state to another. This may include addition, removal, updating or evolution of any particular object in the DB to maintain and update it in a timely manner. Any DB system without a good dynamic information management system would merely become an information store room. According to Engels et al. (1992), a comprehensive representation of DB design consists of static (structural) and dynamic (behavioural) aspects for the DB entities. It is due to this reason; CM in DB intends to take into account both these aspects. However, very few studies have been reported to comprehensively cover both in a unified manner (Engels et al., 1992). We aim to discuss some important works in this section that are focused around the factors explained above.

One of the earliest attempts made for conceptual data modelling is the Entity-Relationship (ER) model introduced by Chen (1976). An ER model is a combination of some earlier proposed models, but it contains its own structure and presentation. Mainly, this model is taken as a standard in data modelling. The ER model divides the process of modelling into different levels including information concerning entities and relationships, information structure and access paths. An important point to note in this classification is that only relevant information is recorded for an entity to be entered in the DB design. The relationships and attributes together form the connection of an entity to be represented in a conceptual model. The information structure map is then used to present the entities and relationships in a model. Rules are then defined for insertion, deletion and updating of an entity in the information structure. Access paths are used to trace the information track of an object. ER model has been very useful for representation of a conceptual model in many disciplines, but it is very suitable for DB designs and management. This model although widely accepted, has been criticised for a number of reasons. The main reasons identified are the lack of expressiveness that is required to represent complex inter-relationship between the objects and to capture the dynamic (behavioural) aspects of a DB design (Engels et al., 1992; Mylopoulos et al., 1990). Since then, several extensions of ER-model and its application to DB design have been proposed. Engels et al., (1992) have proposed to model the DB structure and their dynamics during CM phase using an extended ER model. In their view, a conceptual DB design has to deal with all sorts of different complexities. They have proposed a unified framework based on the extension of an ER model by Chen (1976). In their approach, the static aspect (entities, attributes and relationships) are modelled using the extended version of ER-model while the dynamic aspect (transitions and evolutions) of a DB object are modelled using temporal logic (mathematical formalism used to describe logics and transitions). Their detailed extension of the ER-model and its application to conceptual constraints can be found in Engels et al. (1992).

Another proposal is given by Mineau et al. (2000) who have suggested the use of conceptual graphs (Sowa, 1984) for the purpose of CM in a DB design. They have criticised the use of famous ER formalism and UML in their paper. According to their proposal, ER and UML restrict the expressiveness of semantics (meanings) for sub-relationships between the objects. In their work, DB entities are modelled taking into account the static (description), dynamic (behaviour), context (environment) and evolution (life time) of a DB object. They advocate the complete description of a DB domain in the perspective of knowledge engineering and not just the storage of information. By the use of conceptual graphs, the authors aim to achieve certain qualities in a conceptual data model which we will discuss in section 2.1.2.4. They have compared their approach of conceptual graphs with another data modelling language called Telos (Mylopoulos et al., 1990) and have shown that conceptual graphs have certain advantages over semantically rich modelling language like Telos mainly because of features like graphics, simplicity and integration properties. Ceri et al. (2002) have suggested the use of conceptual web modelling language (WebML) based on the ER model to develop high level content for representation of web resources. By using the conceptual schema, a large amount of information from complex websites is abstracted using data templates and diagrams which are stored in the form of ER data models and hypertext models in WebML.

Sugumaran and Storey (2002) have proposed the use of ontology combined with ER-model for CM in a DB design. In their view, ontology is a very effective approach in DB designs for making a conceptual model richer in presentation, more meaningful and capturing all relevant aspects from the real world domain. According to their view, a conceptual model of a DB developed with this approach will be less time consuming and reusability is likely to increase. Also, constructing an ER-model of a DB design becomes refined and complete if ontological constructs are identified within the DB. Their proposed method includes mapping the ontological construct (entities, relationships, constraints) into CM formalism (in this case, they use the ER-model). An ontology management tool is also developed as a part of their work where a user can create an ER-model using ontological constructs. However in their view, not much work has been done to implement the use of ontology for CM within DB.

One important idea in developing a conceptual model for DB is the use of concept library for CM in DBs (Battista et al., 1989). In this approach, standard definitions are stored for an enterprise DB in graphical form. Battista et al. have differentiated their idea from the traditional concept dictionary where information is stored only in textual forms for a DB design. Their view is to create standard conceptual schemas in different levels of detail which can easily be re-used among different organizations and also across different departments. They have proposed the use of ER-diagrams along with a graphical editor for this purpose. This would also enable easy creation of a new DB from the users within an organization by integrating their existing conceptual schemas with standard libraries through the use of a graphical interface.

Batra and Davis (1992) have performed a comparative study of CM in DB between expert and novice designers with the use of process model based on ANSI (American National Standards Institute). Their study is focused on converting user requirements into a conceptual model of the data. This process model for CM in DB design is shown in figure 2.1. It contains three level of refinement for information namely, enterprise level, recognition level and representation level.

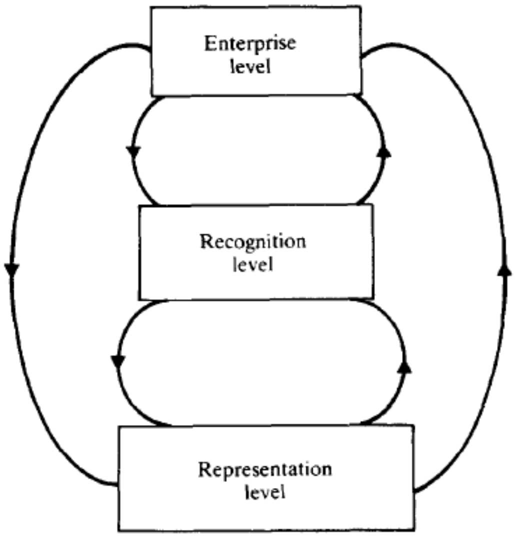


Figure 2.1: Process Model for CM, (Batra&Davis, 1992)

At the enterprise level, typical activities include requirements elicitation and clarification from users and developing an understanding of the problem. The recognition level focuses on some specific requirement and sub-problem so the relevant knowledge structure can be analysed. Lastly, at representation level, a conceptual model would be formed putting together all the relevant knowledge. Any necessary verification of the representation is also performed at this step.

In summary, within DB, modelling languages are mostly used for CM processes with some support from a theoretical basis.

2.1.2.4 DB: Testing/Evaluating of Conceptual Model

There are very few studies focusing explicitly on evaluating a conceptual model for the DB design. One of the main reasons is the properties of DB overlap with IS which is a much broader field and contains some well-established frameworks for evaluating the quality of a conceptual model. However, some works specifically focuses on CM quality when developed for a DB design.

In a work by Crockett et al., (1991), a set of 12 criteria and sub-criteria has been proposed to develop a conceptual model for DB. Their work is based upon the standards developed by ISO for bringing quality into the conceptual model. The four major characteristics identified by ISO for a conceptual model in DB include: ability to capture static and dynamic aspects, able to communicate with both users and designers, easy and understandable, and adaptive to changes. The detailed criteria and sub-criteria based on these standards can be found in the work by Crockett et al (1991). Simplicity, evaluation of design and having logical foundations are some of the important properties discussed in their paper with respect to a conceptual model. Mineau et al. (2000) have presented certain qualities that should be a part of conceptual model in a DB design. This includes completeness, formal meaningful structure and easy interpretation. With respect to these major qualities, they have further sub-divided them into different additional characteristics. Some of their noticeable classifications are structure, context, time and deduction capabilities of a conceptual model. As identified by Engels et al. (1992), conceptual model for DB should have the property to fully capture the relevant information from the domain through the effective use of modelling constructs; hence semantic (meaning) richness in a modelling language becomes an important factor in deciding the quality of the conceptual model produced. The ability to effectively model the dynamic aspects (operations and transitions) of a DB object is considered as decisive factor for quality in a conceptual model. It is only through capturing these behaviours that a conceptual model is able to manage the ever-changing environment of a DB design (Engels et al., 1992). Sugumaran and Storey (2002) have presented an ontology-based approach for the validation of a conceptual model in a DB design. Their proposal for evaluating quality in a conceptual model is to associate the mappings between quality criteria and ontological constructs. A computer-tool will then be used to check these mappings from modellers and from the users to validate the missing or incorrect elements in a conceptual model present in an ontological construct. The relationships and constraints for the DB objects are mainly validated in their work by comparing ontological constructs and elements of the conceptual model.

In summary, various approaches are used to test the quality of a DB conceptual model, including ISO standards, the establishment of general guidelines and ontological support for the evaluation of CM.

2.1.2.5 Summary of CM within DB

In DB, the definition of CM is mostly expressed as a formal document and the scope covers the real world information. The main purpose of CM in DB is for understanding the available information for design purposes. The processes usually use modelling languages with some studies reporting the use of formal methods also. Some quality evaluation frameworks are based on ISO standards, while some do have a theoretical basis from ontology and linguistics. A few quality evaluation frameworks also discuss general criteria and guidelines for CM.

2.1.3 CM in Software Engineering (SE)

In this section, we gather different perspectives of CM present in the SE community. First we define some terms that will be used during this discussion:

Software Development Life Cycle (SDLC): SDLC is defined as a process that begins with the decision to develop software and ends when the product is delivered. Major phases of SDLC include requirement phase, design phase, coding and test phase (Thayer and Dorfman, 1997).

Software Requirements Engineering (SRE): SRE is defined as the science and discipline concerned with analysing and documenting software requirements (Thayer and Dorfman, 1997).

2.1.3.1 SE: Definition of CM

Most opinion about CM in SE finds its basis in the process of abstraction from the real world environment. It is described to be an abstraction from the UOD in which the problem occurs (Andrade et al., 2004). A similar view exists in Rolland and Prakash (2000) who describes CM to be a process of abstracting the specification of the required system. Some other work places this process of conceptualisation as an activity for abstracting information gathered during the knowledge acquisition phase (Andrade et al., 2004). A conceptual model is also regarded as not only the abstraction of real systems, but also of the proposed systems (system to be built) (Kellner et al., 1999). A (conceptual) model is an abstraction of reality and existing system, but it also equally abstracts any system which is to be developed. Both views are discussed by

Stavely (1983). Some aspect of a conceptual model is taken from the user's perspective. In Kung (1989), a conceptual model is defined as a model of the application domain as seen from the user's community. Rolland and Prakash (2000) consider the CM process to be an analysis of information for the users. Conceptual model is also viewed from a solution-oriented approach other than abstraction. Rolland and Prakash (2000) places conceptual model as an intermediate solution between the process of domain knowledge acquisition and system design. In this approach, conceptual model helps to generate possible mappings of the real world entities and their inter-relationships which aids in the design process. According to Andrade et al. (2004), conceptual model is a model of the possible conceptual solution. Some views about conceptual model places it under a procedure of understanding a problem. For example, as Chesterton (1935) argues that the worst thing is not being unable to find a solution, but not being able to define a problem correctly. A conceptual model is regarded as a means to extract information from the environment to solve the problem at hand.

From the above discussion, we can conclude that slight variations exist among definitions of a conceptual model in the SE community. Mostly the studies declare CM to be a process of abstracting information from the problem-domain or from a system that already exists. While specifically some have argued a conceptual model to be an abstracted model of information, some advocates its importance in helping to understand a problem by creating a common platform for both the developers and users of the system. A conceptual model is seen by some as a model of users' perspective when solving the problem at hand, while some studies regard the conceptual model to be a conceptual solution of a problem. Despite these variations, in the definition of a conceptual model, it seems generally to be agreed that the conceptual model helps to understand a system (existing or proposed) in the early phase of software system development. For the most part, a conceptual model is considered to be an informal document that specifies the requirements of a real or proposed system.

2.1.3.2 SE: Purpose of CM

Within SE, a conceptual model is mainly believed to provide abstracted information from real world domains. Extending the view-point of the abstraction, a conceptual model is believed to be a simplified prototype of the system under consideration. It can include only those features of the system which are relevant to address some specific questions. It may not contain all aspects of the system being modelled; instead it provides useful insight about a problem being addressed (Kellner et al., 1999). The work of Rolland and Prakash (2000) has also considered a conceptual model to fulfil the purpose of abstraction from the problem domain. The resulting conceptual model contains only relevant information about the system. In view of Ares and Pazos (1998), modelling is a very powerful technique of abstraction. Since abstraction also helps to simplify the complexities of the system under consideration (by keeping only the relevant details), modelling is regarded as a significant process in software development to speed up the analysis of an overly complex system (Ares and Pazos, 1998; Stavely,1983). Also, CM is an iterative process, once an abstracted model is generated, more and more constructs can be added to develop a detailed model as and when required (Kung, 1989). A model also helps to promote the understanding of a system or problem since it provides a common reference platform for both the users and model developers to communicate (Ares and Pazos, 1998, Kung, 1989). A conceptual model in particular is recommended to include the users' perspective as discussed by Kung (1989). In the software production process, CM helps to keep track of the users' requirements in a traceable and predictable way (Insfran et al., 2002). This is important because a conceptual model aims to solve the problem of the users, and as identified by Kung (1989), the major issue in poor quality software development is to incorrectly understand and conceptualize the problem. In other words, the conceptual model helps to provide the basis of system validity, i.e. whether the product has fulfilled the users' need. This problem-sensitive approach in the context of software development is also discussed in Rolland and Prakash (2000) where the idea is to use the conceptual model in such a way that it helps to solve the "why" part of the system instead of only the "what" perspective. A similar view exists in Andrade et al. (2006), where emphasis has been made on creating a problem-sensitive CM language. The paper argues that most approaches in practice produces a conceptual model from a development perspective which is not the main objective of creating conceptual model, i.e. to understand a problem completely and not how the software solution will be implemented. Parnas and Darringer, (1967) and Stavely (1983) consider software design components (that are built before implementation) as a model themselves which can help in testing the design problem even before implementation. This view is also supported in Ares and Pazos (1998).

In summary, abstraction, understanding and communication are believed to be the most important objectives of CM in SE.

2.1.3.3 SE: Process of CM

Modelling is considered as a fundamental activity in software development (Stavely, 1983). Also it is interesting to note that with the rapid advancement of technology and strict requirements on software systems, (conceptual) modelling based software designs are becoming more important. A typical place for conceptual model in the software design process is shown in figure 2.2 which is based on the work of Kung (1989). As shown, CM is mostly performed at the early stage of the system development where knowledge about a particular domain is collected. This is the same stage where requirements (specifications) of the software design are gathered, documented and communicated using variety of forms (textual, graphical, tabular etc.). The importance of this pre-design phase resulted in a separate sub-discipline known as Software Requirements Engineering (SRE) in the early 1980's (Page Jones, 1980).

This overlap between SRE and CM is also supported in the literature which places CM in a broader context of requirements engineering (Rolland and Prakash, 2000). It is also argued that during CM, information about processes and events are also included along with the user's perspective of the systems' design, hence, CM is seen as a technique in practice even before the emergence of SRE (Insfran et al., 2002). This is one of the main reasons that CM and requirements-gathering are placed at the same stage of SDLC as identified in Kung (1989) and most of the SRE methods tends to produce an intermediate (conceptual) model of the specifications in a software design (March and Allen, 2009). We will come back to this assertion when we discuss the SRE methods later in this section. Other than the SRE methods, other techniques (based on linguistics, cognition and formalization) are discussed first which are also used in SE for producing a conceptual model.

This conceptual model is verified and validated against the requirements of the system before the design stages take place. However, with the rapid increase in the requirements of the software systems, more effort has been placed in developing the simplified models at each stage of the SDLC to aid the process of whole system design and maintenance. The process of conceptualising a problem in SE is a broad area where this process can range from a very basic conceptual constructs to a very advanced formal methods techniques. We aim to discuss some frameworks briefly that can help to get an overview of the nature of work being done.

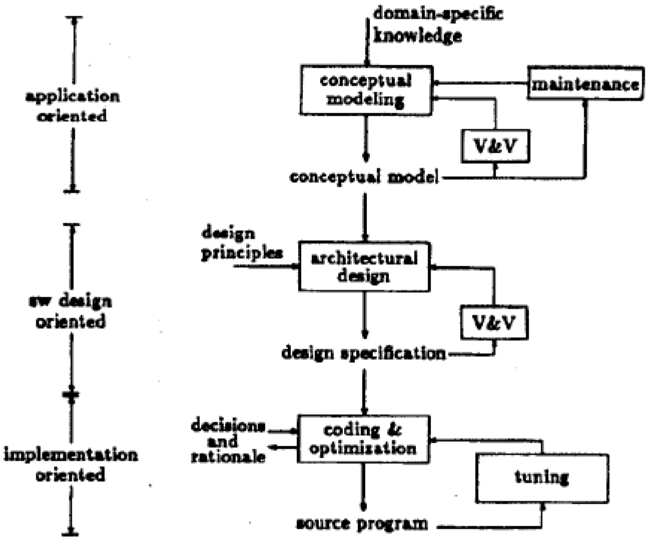


Figure 2.2: Place of CM within a software development life cycle (Kung, 1989)

One of the main studies that bring the idea of conceptualisation into SE originates from Buchanan et al. (1983) which states that it involves specifying key concepts and their relationships. Formally a conceptual model consists of three components (concepts, relationships and functions) (Ares and Pazos, 1998; Genesereth and Nilsson, 1986). This view of conceptualization is discussed in more detail in Goamez et al. (1997). According to this view, a *concept* is a description about any component of a system including tasks, functions and actions. In the first place, many concepts can be constructed about a problem, and then they are reduced to a workable numbers through the use of famous techniques of *abstraction* or *generalization. Relations* are used to represent the connections between the concepts. Finally, *functions* represent a special class of relations which can be used to represent any specific interrelated association. The framework of Ares and Pazos (1998) describes a three phase process for conceptualisation of a problem. These three stages are acquisition, conceptualization and formalisation. Knowledge about a problem domain is acquired in the first stage through information *elicitation*. Once this has been done, this knowledge is subjected to the conceptualization phase where it is analysed and synthesized to produce a conceptual model (*output*). Formalization is the stage where a formal model is produced after applying a simple logical process of deduction to a conceptual model. This model remains operational and used heuristically to infer the properties of the real system.

Their exists some work by Andrade et al. (2006) which suggests the use of natural language grammatical categories (nouns, verbs, adjectives) for creating the CM constructs. According to this paper, there exists great potential in this approach because of the parallelism between natural language and a CM language (Hoppenbrouwers et al., 1997). Their process of CM is more towards defining a CM language based from the view-point of the problem domain (rather than only from the developer's perspective).In this process, when all CM constructs have been defined based on natural language grammar, it is then classified according to the static and dynamic information level resulting in a conceptual model (*output*) containing all relevant information about the domain under study and hence can greatly help in understanding the problem at hand.

The framework of Kung (1989) discusses the approach of CM to construct an executable and visual conceptual model. Their argument in developing such features in a conceptual model is to make it understandable, testable and communicative both by the "humans" and "machines". This proposed framework models the static aspects of the system using Entity-Relationship figures and the dynamic aspects of a system are modelled based on traditional Data Flow Diagrams. The method also includes mathematical logic to develop a conceptual model which is then shown to undergo a consistency checking formally. The paper advocates the use of these combined approaches in developing a verifiable and complete conceptual model. Delugach (1992) have proposed the use of conceptual graphs (Sowa, 1984) to develop a common representation from four different modelling languages namely, Entity Relationship model, Data Flow Diagram, state-diagram and requirements network. Their work advocates the use of conceptual graphs in order to capture all relevant information about a particular system. A new software development methodology called eXtreme Programming (XP) has been introduced in recent years which involve greater use of testing before and during programming so it focuses on more responsiveness to changing customer requirements (Beck, 1999). Insfran et al. (2002) have proposed the use of CM in an XP style to find the desired solution. They suggest the use of higher level abstractions as programming constructs using XP based approach. These models are then automatically translated to their final software product using a code generation strategy. Their idea of extreme CM is a slight extension of their earlier work on CM (Insfran et al., 2002). Other related work that uses XP ideas to the modelling phase is discussed in Boger et al., (2000). Their idea is to use (conceptual) models in a way that available information are developed into UML and tested interactively. The validity of the model can be checked during modelling phase and more confidence is put in the model during the coding phase.

As per the discussion earlier in this section regarding the similarity between CM and SRE, we now consider some SRE methods that produce an intermediate conceptual model (or software specification document) during software development. The main approaches identified from SRE include:

- Goal-oriented methods (goals or objectives based SRE method where problem elicitation is based on considering different goals in a system)(van Lamsweerde, 2001)
- Scenario-oriented methods (requirement analysis is based on considering possible scenarios of a problem domain) (Sutcliffe et al., 1998)
- View-point oriented methods (a system is considered from different view-points to capture requirements from all possible stakeholders) (Kotonya, 1999)
- Hierarchal and graphical modelling methods (IDEF family/SADT). The Integrated Definition for Function Modelling (IDEF) is based on Structured Analysis and Design Technique (SADT) and provides a family of methods that supports structured methods for functional modelling during SRE (Dickover et al., 1977; Greenspan et al., 1982; Menzel and Mayer, 1998; Aguilar-Saven, 2004).

There are various frameworks in the category of goal-oriented and scenario-oriented methods which are discussed first. An important framework is proposed in Insfran et al. (2002). Their approach is to proceed in a systematic manner from a high-level user requirements model to the final computer model. In doing so, they suggest the use of a conceptual model which helps to make this transition as accurate and precise as possible. They discuss the use of scenario diagrams along with other techniques like use case, context diagrams and object-oriented methods. Their conceptual model (output) from the CM phase is a UML-based conceptual schema used for producing the final computer model. In Rolland and Prakash (2000), a complementary framework to the work of Insfran et al. (2002) has been discussed which suggests the use of SRE methods (particularly goal-oriented and scenario based methods) for producing the conceptual schema as a result of CM. This conceptual model (output) is then used for detailed system design. The paper advocates the use of SRE methods in order to formally process the CM phase.

The framework proposed by Andrade et al. (2004) presents a view-point oriented approach for CM. The idea presented is to look at the problem domain in the initial acquisition phase from different view-points. Each view-point helps to associate the individual or group of individuals to a particular view of the problem. The conceptual models for the different view-points are then generated iteratively. They are finally integrated into a complete conceptual model (output) of the problem under study. Another view-point oriented method called PREview (Sommerville and Sawyer, 1997) was proposed to handle multi-perspectives requirements in a software design. The conceptual schemas in this method document the information for every possible view-point in a system. This can include any person, department, policy, physical objects etc. The idea is to capture the system's information from all perspectives to generate a cohesive document about system's requirement. More details on this method are discussed in chapter 4.

IDEF family has been used extensively in the requirements specification for very long and dates back to 1970's (Dickover et al., 1977; Greenspan et al., 1982). Although there have been various versions of IDEF family packs since its introduction, the most important parts are: IDEF0, IDEF1, IDEF1X, IDEF2, IDEF3, IDEF4. Among all these versions IDEF0 and IDEF3 are mostly used for enterprise requirements modelling, functional specifications and for graphical representation of complex systems or processes (Hickey and Davis, 2003; Aguilar-Saven, 2004).

IDEF0 is based on Structured Analysis and Design Technique (SADT) (Ross, 1977) which has been the focus of software requirements specifications since early 80's (Greenspan et al., 1982). SADT's simple graphical and hierarchal structure was adopted by ICAM (Integrated Computer-Aided Manufacturing) in the mid-1970s as a result of initiative by US air force to model activities, data, and dynamic (behavioural) elements of manufacturing operations (Aguilar-Saven, 2004). More detail on SADT is discussed in chapter 5.

In summary, in SE, a conceptual model is mostly developed using SRE methods or modelling language approaches. Some studies also support the use of formal methods logic.

2.1.3.4 SE: Testing/Evaluation of Conceptual Model

Mostly the SE literature has not specifically focused on evaluating the conceptual model within the context of software development. The overall model quality has been the subject of most studies which is usually from an IS perspective. One reason for not making this work explicit may be that SE is seen as a subfield of CS and it is not seen as a separate discipline. However, with the emergence of the technologies based on a model-oriented approach in software development (for example, standards developed by the Object Management Group; (http://www.omg.org/, accessed 06th June, 2015), much effort has been put in determining the quality of the models developed during the different phases of SDLC. Around 40 studies have been reported in a literature review by Mohagheghi et al. (2009) from 2000 till 2009 which discuss this subject. Since our interest is in the CM perspective, those studies that are relevant to some or all aspects of quality in a conceptual model have been selected. Although there are frameworks which specify conceptual model quality within SE context, still some frameworks are taken as standard in SE, IS and databases community to evaluate the conceptual model quality. Some important frameworks are either cognition based (information processing view of an individual's psychological functions) or ontology based (philosophical study of the nature of being, existence, or reality). Moreover, they can also be classified on the basis of a *product* or *process* oriented approach in evaluating the quality of conceptual model.

One of the most important works on evaluating the quality of the CM process is presented by Moody (2005) who has strictly argued about the standards for developing a good conceptual model. In his view, a conceptual model quality is "*the totality of features and characteristics of a conceptual model that bears on its ability to satisfy stated or implied needs*" (the definition is based on ISO 9000 definition of quality). He places CM under a sub-discipline of SE as they are used to define users' requirements in the early phase of system design. It is due to this reason, he proposes the use of relevant international standards in software quality (ISO/IEC 9126) and quality management (ISO 9000) for evaluating a conceptual model. He has reported 12 issues related to the development of standards for CM quality evaluation within SE. By using the international standard of software quality for conceptual model quality, he aims to achieve a concise, measureable, well-defined and consistent quality framework for evaluating a conceptual model. Dromey (1995) has also proposed a similar framework for describing software quality models. Both frameworks differ in the level of quality criteria. While the first quality model based on ISO/IEC 9126 contains deep decomposition of quality characteristics into sub-characteristics (for example: reliability, usability, maintainability etc.), the latter one uses a simple single level of quality-carrying properties (for example, component properties and quality attributes). Moody (2005) has also presented several approaches in his paper for developing and validating the conceptual model quality frameworks. He has compared the use of Dromey's model and Goal-Question-Metric (a top-down approach to systematically build a measurement model) (Basili, 1992), to evaluate the quality of a conceptual model. Other approaches are also discussed in his work.

One important framework is proposed by (Lindland et al., 1994) which is product oriented and it aims at a deeper understanding of quality as it relates to CM. This work has been taken as a standard framework for CM quality and is used in many other derivations of conceptual model quality frameworks (for e.g. see Nelson et al., 2012). The essence of the framework lies in the distribution of quality from *syntax* (structure), *semantic* (meaning) and *pragmatics* (interpretation by audience) categories. These important linguistic concepts are being applied to four aspects of modelling in the framework, namely: language, domain, and model and audience participation. The proposed framework identifies several *goals* and *means* to achieve these three categories of qualities. For example, a well-structured model is able to achieve comprehension (goal) through inspection (means), hence this goal (comprehension) supported by some particular mean (inspection) in the framework is used to achieve *pragmatic* quality. The framework distinguishes from other frameworks in a way that it explicitly discusses the importance of goals and means in achieving a high quality conceptual model.

The complementary framework for quality in CM is proposed by based on Bunge's ontology (Bunge, 1977). This is also one of the most important frameworks along with the framework of Lindland et al.), but aims at the process view of a conceptual model. The difference being the first one focuses on final conceptual model quality, while the later one discusses a series of transformations for CM from the real world domain and defining quality along these transformations. Quality is described in a sense that every allowable state in a source should have a corresponding mapping in a transformation state. Nelson and Monarchi (2007) discuss a quality framework that combines the two approaches discussed in the above two works.

Qi et al. (2010) have discussed the influence of CM on software reliability which is defined as the extent to which a system performs the required task for a specific time period. This is divided into five elements; maturity, availability, fault tolerance, recoverability, and reliability compliance. This definition is based on the software reliability model of ISO/IEC 25010 which is used as a basis for their work. Their main idea is how to map conceptual model reliability characteristics to SR characteristics. For this purpose, they have proposed a layered framework that combines most of the experience-based quality frameworks. This work differs in a sense that it explicitly focuses on the contribution of conceptual model reliability towards final software product reliability.

There are some other proposals that also argue for an *error detection (product oriented)* approach in CM rather than *error prevention* (process-oriented) approach. The difference in the approaches is taken from the view point of software development. The former is focused on the final *product* inspection once it has been developed while the later approach has a *process* quality view of a conceptual model that put emphasis on the process deployed in generating a good quality conceptual model. One such framework is proposed by Marin et al. (2010). They argue that by using a defect detection approach, a high level of empirical validity is achieved through observation of different conceptual models created during software development. This work focuses on model-driven approaches for software products. The paper presents a very detailed modelling approach for defect detection for conceptual models. Their main criteria for the error detection approach are to find the defects from three different sources, namely data, processes and their interaction during the process of CM.

In summary, there exists strong evidence of standard processes for CM within SE community. The importance and purpose of a conceptual model has equally gained much attention by its application in the initial requirement analysis and problem structuring. One of the most important tasks performed through the use of a conceptual model is model abstraction. There are very important works, as identified, which support a conceptual model as a means of abstraction in requirement gatherings, communicating and development of the final design. ISO standards are mostly used for evaluating CM quality. Some useful support is available for the evaluation of general quality criteria along with a theoretical basis for testing a conceptual model.

2.1.3.5 Summary of CM within SE

The definition of CM within SE has mostly taken the perspective of informal diagrams at the start of the process and the scope covers the model of real or proposed system. The objective of CM within SE is mostly to understand and communicate initial requirements among stakeholders. Abstraction of the required details from initial description of software specification is also considered an important objective of a conceptual model. Processes of CM are mostly based on SRE methods along with the use of modelling languages. There are several frameworks that use SRE methods to produce the intermediate conceptual model of the software specification. Some frameworks are based on formal methods that are in practice for producing conceptual model during software development. Testing and evaluation of the CM is supported by ISO standards along with some general guidelines. Other frameworks (such as from Bunge, 1977) are based on theory from ontology for quality evaluation.

2.2 Conceptual Modelling in M&S

Until recently, little work has been done in the field of M&S to promote the importance of CM. There are only a few works in 1970s and 1980s that addresses some aspects of CM (Schruben, 1979; Innis and Rexstad, 1983; Landry et al., 1983; Ward, 1989). CM has been recognised as an integral part of M&S development as the complexity and size of M&S applications have continued to increase (Arthur and Nance, 2007; Balci and Ormsby, 2007; Robinson, 2002; Robinson, 2004a; Robinson, 2008a). In this section, CM within M&S is discussed using the same four components as for the discussion above.

2.2.1 M&S: Definition of CM

The definition of the conceptual model is not very well defined in the simulation literature. However there is some agreement in accepting that a conceptual model is a part of early stages in a simulation study (Robinson, 2008a).

According to Guizzardi and Wagner (2012), there exist at least three different views on conceptual model definition and broadly falls into the works from Balci et al (2008), Turnista et al (2010) and Robinson (2008a). From these definitions, Turnista et al (2010), takes a more IS approach where a conceptual model is a formal specification document based on some ontological foundations. Their idea is more towards a computer implementable presentation of the reality in a conceptual model. Such a conceptual model would differentiate between the details of model along with representing shared conceptualization. This will enhance composability and interoperability of models among different simulation projects. According to this view, current use of ontology within M&S restricts a modeller to capture a system according to a particular research question. This view appears to be more towards systems engineering approach where libraries of models are created that are reusable for different simulation context. Robinson (2008a) defines a conceptual model as follows:

“The conceptual model is a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model”

Robinson (2008a)

This definition of a conceptual model is probably the most comprehensive one to date. According to this view, a conceptual model is the simplified representation of a simulation/computer model that may or may not exist. This is to be noted that a conceptual model in this definition is actually *model of a model* that is to be or has been created. This view is in contrast with that of Balci et al (2008) which states: “A simulation conceptual model is a repository of high level conceptual constructs and knowledge specified in a variety of communicative forms intended to assist in the design of any type of large scale complex M&S application”. Balci’s view of conceptual model lies more within the problem domain where it is used for knowledge acquisition and system description. Robinson’s definition on the other hand is placed within the model domain where a conceptual model is actually a description of the computer model showing how the computer model is simplified from the understanding of the real world problem. Some other differences also need to be clarified here. The view of Balci about a conceptual model is not generic and applies to large scale and complex applications whereas Robinson’s view is more general and so applicable also to small-scale generic business operations models. This is because for a large-scale models (like in military and defence), there is a need to capture all details closely and document for communication purposes in the beginning of the whole simulation study. It also makes sense to have this conceptual model remain alive throughout the lifecycle of the project. On the other hand, for small business operation models (project duration of less than six months), temporarily (to be) created by a lone modeller, there is a need to rapidly identify simulation modelling objectives and scope in the very early phase of simulation study. This conceptual model may not be documented or saved in a communicative form and may remain only in the mind of the modeller. This is the reason that we may have more than one conceptual model of the same system answering different questions. So such a conceptual model has been or will be created to decide what to model in the final simulation model. Also Balci’s definition must involve some subject specialist and communicative media to result in multiple conceptual models that may be suitable for a reused-based application design. So this view of a conceptual model is supporting a library of conceptual models created in some specific domain. The same idea is supported in Balci and Ormsby (2007) who defines a conceptual model as the highest layer of abstraction closest to the mind of the modeller. However, their view is again placed within large-scale simulation design context. The major difference is that although this perspective places a conceptual model closest to the level of thinking of the modeller (same as Robinson's view), the objective for which the modeller has this conceptual model in his mind is different. Finally it is also to be noted that Balci's view seems to be driven more from software/simulation engineering where conceptual model aids the requirement gathering phase. Hence this conceptual model is more towards the system description defined in Robinson's view, where a system description refers to the structuring of the problem within the problem domain. This involves acquiring knowledge from the problem domain, connecting it with the other available information, making any necessary assumptions, and finally producing a document (or remains as a mental model) that describes what the problem is about (Robinson, 2011).

Other than these major definitions, researchers have regarded conceptual model to be an abstraction of a real or proposed system. According to Kotiadis and Robinson (2008), CM is the process of abstracting a model from the real world. This also agrees with Robinson (2011) which declares CM as the process of deciding what to model. Since this agreement about what to model cannot be reached without some simplification and assumptions, Robinson further explains that the conceptual model is the abstracted model from some part of the world it represents. This part of the world may or may not exist. This also implies that a conceptual model is a necessary component for all simulation modelling as every model is some simplification of the existing or proposed system. The view of abstraction is also supported by Tolk and Turnista (2012), who states modelling to be a useful process of abstracting theories from a system and capturing this extracted information in a conceptual model. According to Landry et al (1983), a conceptual model is the *mental image* of the problem situation and formed by the perceptions of the modeller and decision-makers. This mental image can be present in the minds of modeller or the users. Pidd (1999) also defines a model to be an explicit representation of part of reality as seen by the user. Although this is a more generic *model* definition, it seems consistent with the view of a conceptual model in M&S.

Form the above discussion, it can be concluded that there is some agreement about conceptual model definition (e.g. abstraction and the place of a conceptual in simulation project), but a more agreed approach is required (in terms of generic modelling domain, scale of the project and scope of a conceptual model) so as to develop standard presentations required for a consistent CM framework. It is hard to develop a standard working framework in the existence of ill-defined conceptual model definitions.

2.2.2 M&S: Purpose of CM

A conceptual model serves as a specification for the final computer/simulation model (to be/being/has been) developed (Robinson, 2011). According to Balci (1994) and Robinson (2004b), modelling is an iterative activity so a conceptual model serves as a continuous update for the simulation study. This also implies that a conceptual model should remain active throughout the entire life cycle of a simulation project. A conceptual model helps to decide the modelling objectives along with the general project objectives (Robinson, 2008b). This can include flexibility, time-scale, budget etc. In that sense, presenting organizational concerns is also one of the purposes of a conceptual model. In the editorial for *Journal of Simulation*-special issue on CM, Robinson (2007) presents potential areas for research in the area of conceptual model. One of the important findings is the objectives of CM. This includes identification of the right problem, understanding the system and achieving consensus over the nature of the problem. Among the model domain itself, major objectives identified include communication, verification and validation of the model. A conceptual model may be written in a form that is communicative to different stakeholders (managers, analyst and model developers) (Balci and Ormsby, 2007). Fifteen objectives of a conceptual model have been identified by Balci and Ormsby (2007) in their paper, mostly concerning the role of a conceptual model in a large-scale complex simulation models, but also applies to the general purpose of a conceptual model in a simulation project.

A conceptual model also serves as a representation of the user's perception about a particular problem (van der Zee and van der Vorst, 2007). This view confirms that a conceptual model not only acts as an abstracted model of the problem domain, but also serves to maintain a complete system description by generating different possible view-points of various stakeholders. According to Pidd (2007), a conceptual model as used in soft system methodology, serves as a basis for debate about the differences between the situation as it is now and the situation as it might become. The other type of model he mentions is the one that represent the features that should be present in a model. Both views are based on the works of Checkland (1995). According to Tolk and Turnitsa (2012), a conceptual model is a design guidance and inspiration for simulation. This is also supported by Robinson (2008a). Other than a guide to simulation some other purposes are commonly held including verification and validation tests, communication, analysis and optimization (Tolk and Turnitsa, 2012). This view is also supported in Balci (1994) and Sargent (2005). In the view of Guizzardi and Wagner (2012), the main purpose of a conceptual model is to capture, a part of the real-world domain under consideration. According to van der Zee and van der Vorst (2007), a conceptual model increases understanding of the analyst about the domain, provides a communication channel between developer and user, and serves as a basis of design and documentation. Finally, a conceptual model is found useful in developing a participative modelling approach which helps to engage all stakeholders in the early stage of a simulation study (Tako et al., 2010).

In summary, there is consensus on the conceptual model being used for understanding and communication purposes. However, there are differences when it comes to whether it represents the real system, proposed system or the abstracted model of the real system. Some studies also regard it as the basis for verification and validation of the final simulation model.

2.2.3 M&S: Process of CM

There are few well defined frameworks in the area of CM despite it being recognised as the most important phase in the simulation project process (Robinson, 2008a). Also, simulation practitioners, and especially novices, find it difficult to determine the level of detail (what to include, exclude or measure etc.) to be addressed in a conceptual model (Montevechi and Friend, 2012). There is no major categorization of the CM frameworks available in the literature. However, the frameworks can be broadly classified as descriptive, formal methods, diagrammatic and theoretical frameworks. Most of the work has been done in descriptive mode, while some frameworks are formal methods based. Others follow a diagrammatic technique. Very few proposals are based on traditional CS theories. It is often difficult to clearly identify these distinctions as for example, a descriptive framework and diagrammatic framework can overlap using a combination of both. We start by looking at one of the well-accepted descriptive frameworks from Robinson (2008b). Most of the other frameworks in this category are either extensions to this work or modification to suit a different modelling context.

The two major phases of CM as described by Robinson (2011) are knowledge acquisition and model abstraction. While the first phase helps to form the initial system description, the second phase will generate the possible conceptual model of the problem. These phases with additional artefacts of modelling are shown in figure 2.3

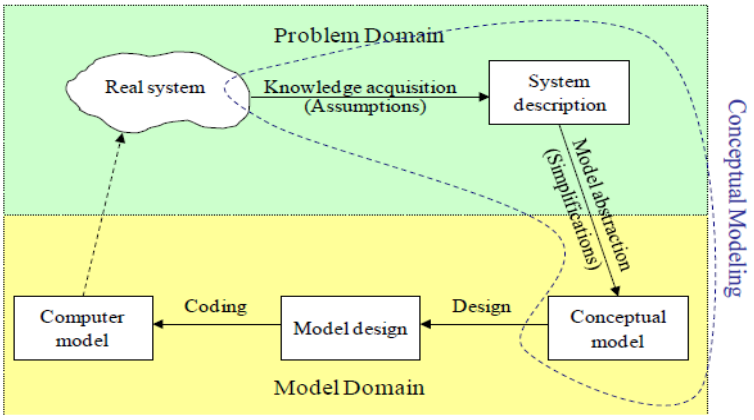


Figure 2.3: Artefacts of CM (Robinson, 2011)

According to (Robinson, 2004a) a conceptual model passes through these activities: 1) developing an understanding of the problem situation. 2) Determining the modelling objectives. 3) Design the conceptual model: inputs, outputs and model content. 4) Collect and analyse the data required to develop the conceptual model. The framework proposed by Robinson (2011) consists of these activities are shown in figure 2.4. The illustrations of these activities are discussed in more detail in (Robinson, 2008b).

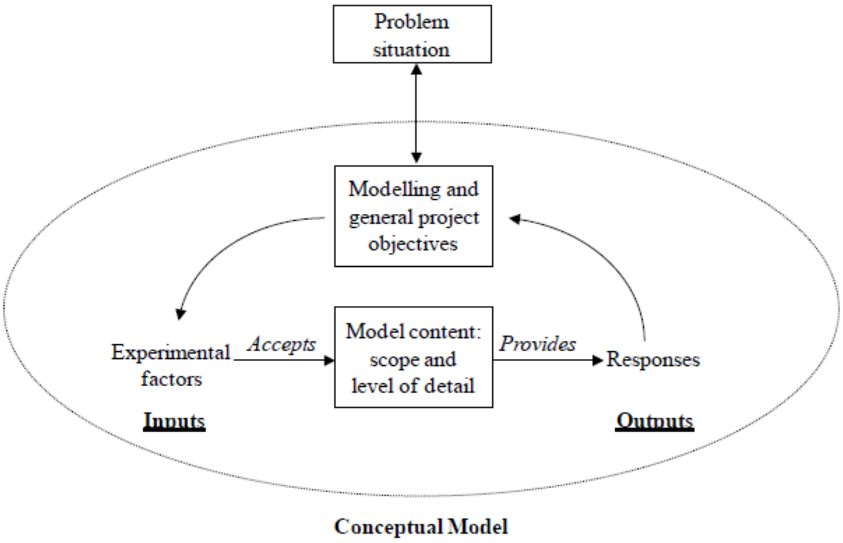


Figure 2.4: A Framework for CM (Robinson, 2008b)

As shown in the figure, these activities are linked in some order but may not necessarily follow this sequence as per the iterative nature of CM (Robinson, 2011). A CM phase usually starts with the understanding of the problem situation which is followed by determining the modelling objectives. The next step is to determine the scope and level of detail for the project. This will be accompanied by identifying possible inputs (experimental factors) and outputs (responses) related to the simulation study. This can be checked regularly by comparing the scope and level of details with the modelling objectives. Some other descriptive and diagrammatic frameworks have been discussed recently in Robinson et al (2011). This includes a framework for participative simulation models (van der zee, 2011) that uses the idea of decomposition principles for overall structure and the interaction of the components within the model. The framework is explicitly designed for the manufacturing domain. Another framework discussed in the book is an activity based framework (called ABC framework) proposed from Arbez and Birta (2011). This framework separates the structural and behavioural components of the system under investigation through uniquely identifying the static structures (entities, attributes, identifiers, variables) and dynamic constructs (activity, action). Input, output and data components are entered separately into the detailed conceptual model. Although the framework advocates covering a comprehensive and guided approach for DES modelling, it is non-trivial to identify such information for a quick decision support from within a simple conceptual model. Such an approach suits more a database design where it is necessary to model the static and dynamic information about the system. The other domain-specific framework presented is called the KAMA framework (Karagoz and Demirors, 2011) that has been used for developing conceptual models of the mission space for simulation systems. The three main components of the KAMA framework are; KAMA method, KAMA notation and KAMA tool. KAMA method consists of task flow diagrams detailing the main tasks at each phase of CM. KAMA notation consists of task flow diagrams and entity state diagrams that are used to represent behavioural aspects of the conceptual model while structural features are represented by entity-ontology, entity relationship, command hierarchy and organizational structural diagram. As reported, KAMA tool has also been developed to facilitate KAMA method and notation.

Kotiadis (2007) proposes the use of Soft System Methodology (SSM) to determine the simulation study objectives. This framework focuses on the initial part of the CM phase given by Robinson (2004b) where the objectives of the simulation study are determined. A recent work by Montevechi and Friend (2012) proposes a five-step framework based on SSM to achieve the phases of a conceptual model defined in Robinson (2008b).Mainly their work presents the combination of two different SSM proposals (Lehaney and Paul, 1996; Kotiadis, 2007) which tackles different phases of Robinson’s Framework. Pidd (2007) also discusses the use of SSM in relation to simulation projects. A more detailed work on the use of SSM has been acknowledged in the area of participative and facilitated CM (Kotiadis et al., 2013). The main contribution of these works is to involve stakeholders during the CM process so as to make better comprehension and increased transparency. In this area, another interesting work is the development of PartiSim (Participative Simulation Modelling) framework from Tako et al. (2010). It consists of six main stages as follows:

1. Initiate the study;
2. Structure the situation of interest;
3. Specify study objectives;
4. Develop Simulation model ;
5. Experimentation and
6. Implementation

These stages are based on processes from simulation modelling textbooks (Pidd, 2004; Robinson, 2004b; Law, 2007). (Recently an extension to the framework of Robinson (2008b) has been proposed by Chwif et al (2013). Mainly they have added complexity description, scenario description and input/output description within the same block. Other differences include more detailed forms of input (e.g. rough collected data) to be specified in a conceptual model and a full textual description along with diagrams to identify the model contents (assumptions and simplifications). Another proposal for integration of Robinson’s framework with the engineering processes is given by van der Zee (2012). His argument is that there exists a gap between simulation activities and engineering processes and by linking these two areas in terms of presentation, control, process and data integration, it could enhance the use of a conceptual model in other disciplines.

Heavey and Ryan (2006) have presented an overview of the methods/tools that could support (conceptual) modelling. They claim that none of the methods/tools have been specially designed to fully support CM. According to their classification, these methods and tools can be divided into *formal methods* (e.g. Petri nets, DEVS, state chart etc.) or *descriptive methods* (e.g. UML state charts and Activity Diagrams). While these tools support some characteristics required to develop a simulation model, they lack in capturing the requirements of a conceptual model. Their paper presents a modelling language called Simulation Activity Diagrams (SAD) to overcome this problem (Heavey and Ryan, 2006). Van der Zee and Van der Vorst (2007) have also presented diagrammatic techniques to support conceptual model creation in manufacturing simulation. Three techniques have been discussed in their paper: activity cycle diagrams, petri nets and flow charts. Their work is domain specific and does not provide a generic framework for CM. An interesting work in the area of CM representation is from Onggo (2009), who proposes use of various diagrams in the problem domain (objective diagram, purposeful activity model, influence diagram, business process diagram) and also in the modelling domain (activity cycle diagram, event graph, stock and flow diagram, causal loop diagram). The paper follows the definition of Robinson (2008a, 2008b) and suggests the use of these diagrams according to the contents of a conceptual model proposed by Robinson (scope, level of detail, inputs, outputs and contents).In the context of large scale simulation projects, Balci and Ormsby (2007) have presented a detailed simulation CM lifecycle which includes problem formulation, system definition, high-level conceptual model, detailed design and conceptual model specification. The activities are linked with each other in an iterative manner so as to perform any necessary verification and validation (V&V) in forward and backward directions. This V&V within (Balci and Ormsby) life-cycle is discussed briefly in section 2.2.4.

Recently some proposals of adapting CS theories and techniques have emerged in the series of Winter Simulation Conferences. Tunitsa et al (2010) have suggested the use of ontology in forming the research question that exists in the mind of the modeller. They have argued that ontologies in M&S serve to capture the reality according to the perception of the modeller whereas in other disciplines ontology is used to describe an observable system. They believe that conceptual model from ontology is formal enough to be machine readable and hence they treat the conceptual model as an ontological representation of the simulation that implements it. This perspective is more towards the IS domain. Guizzardi and Wagner (2012) have suggested the use of UML class diagrams and BPMN with some improved ontological foundation. They call the updated version as “onto-UML” and advocate its use for CM. Their view of producing conceptual model in this way can lead to a better representation of the real world producing a digital artefact implantable in computers. Another interesting work in this area is from Tolk and Tunitsa (2012) who propose a process-driven approach towards CM. They assert that for a valid conceptualization, it is necessary to consider views for the state (objects) and activity (process) together in a model to be able to answer questions about the system. The third element is the association of the objects and process which is called a relation in the literatures of ontology. As per this view, relations are a special class of functions that associates objects (inputs, outputs, identity, state etc.) with processes (transition functions, model components, abstraction etc.). This view is close to Robinson's (2008) view where a conceptual model acts as transformation process and associates inputs to the outputs thorough model content and details. From a CS view, the concept of relation is seen largely in the ER-model by Chen (1976) and can also be seen in other graphical languages like UML or SysML where boxes are interconnected to show different processes (Tolk and Turnitsa, 2012). The problem with these proposals is that they require expert knowledge of the subject matter to begin with and they are more focused on formalisation of the processes rather than on describing a structured method for novices to use.

In summary, a cluster of scattered frameworks for use in CM exist. From these, the descriptive frameworks have provided a good foundation for a CM framework; however, there is a need for guided methods which can provide step-by-step instructions for the development of a structured conceptual model. Also, from the above discussion, it is clear that an ideal CM technique would require a combination of approaches within M&S. Thus, for example descriptive frameworks lack in structure and precision for CM, while on the other hand, typical formal methods or modelling languages lack expressiveness or require subject matter expertise. Moreover, none of the above mentioned approaches discusses the integration of information about the physical components in a conceptual model (queues, entities, resources) along with the functions and processes (input, output, states etc.) in a simple and communicative form. Developing a CM framework that integrates and structure the physical components of a DES study into a conceptual model is important. A useful study that deals with this aspect is the approach proposed by Montevechi et al. (2010), where IDEF methods are used for CM. They suggest mapping the CM elements within computer model elements (entity, functions, controls etc.); however, the necessity to integrate different modelling techniques with IDEF0 nomenclature is not clear in the paper. Moreover there is no precise guidance available for the users to adapt the proposed framework. There are few other related works in this area which are discussed in section 5.3 where adaptation of SADT for M&S CM is discussed in detail.

2.2.4 M&S: Testing/Evaluation of Conceptual Model

Unlike in the field of CS, no theoretical frameworks exist in M&S to test the quality criteria of a conceptual model (for example, ontology based or cognition based). Work on deciding requirements, criteria and Verification and Validation (V&V) of simulation models in operational research is on-going (Kleijnen, 1995; Robinson, 1997; Sargent, 2011; Oral and Kettani 1993; Landry et al., 1983; Balci and Ormsby, 2007; Sargent, 2005). However, in a search for literature on conceptual model quality and testing, the work from Robinson (2008a) seems to cover reasonable requirements explicitly for achieving a good quality conceptual model. However, a standard framework to achieve these requirements and evaluate a conceptual model against them does not exist. The four key requirements of a conceptual model presented in Robinson (2008a) are validity, credibility, utility and feasibility. A very detailed discussion on these key requirements related to a conceptual model quality can be found in Robinson's paper. Here we briefly present an overview of these four quality criteria. Validity of a conceptual model is from the modeller view that a conceptual model can be produced in to a final computer model with sufficient accuracy. Credibility is from the clients' view that a conceptual model would build a computer model that is accurate to fulfil the purpose. Utility is from both the modeller and client perspective that a conceptual model would generate a computer model that is useful as an aid to decision-making within the specified context. Lastly, feasibility is again from both the modeller and client view that a conceptual model can develop a computer model within the time and data available (Robinson, 2008a). These criteria are an important foundation on the requirements of a conceptual model. Keeping in view these requirements, a well-directed conceptual model is possible within the time and resource available, equally fulfilling the purpose at hand. Robinson put further emphasis to keep the model as simple as possible due to several advantages reported in the paper including faster development, more flexibility and easier to run.

Sargent (2005) discusses *conceptual model validity* in a more scientific manner. This is because his view of a conceptual model is more towards a mathematical, verbal and logical representation of the system under consideration. According to this view of conceptual model validity, a theoretical validation is necessary to make sure that the conceptual model that follows from system theories is based on the right mathematical models, techniques and distributions. Only after that, a conceptual model can be validated for causal relationships and logical structure. This method of validity and formation, however, requires a subject matter expertise and questions the development of good quality model by beginners in business operations. A contrasting view to that of Sargent (2005) is presented by Landry et al (1983) who place conceptual model validity before the logical validation. According to this view, a conceptual model validation involves searching for correct objective (reasons to build a model) both from a modeller and decision-maker perspective. This may require iteration back and forth several times between a conceptual model and problem situation. It is only after this stage that a formal model validation can be performed which may involve detailed low-level checks to confirm if the formal model has been translated accurately from a conceptual model capturing all its richness in formal languages (computer code or mathematics) (Landry et al., 1983). An extension to the work of Landry is reported in Oral and Kettani (1993), who combine the experimental and logical validation (in the absence of a formal model) in what they call *legitimational validity.* According to this view, the suggestions proposed by the analyst and decision-makers, must be assessed on the scale of acceptability, level of commitment and their usefulness to formulate a 'decision'. Kleijnen (1995) discusses some techniques for simulation model verification and validation. He discusses various scientific approaches for verifying the final simulation models (like good programming practice, statistical methods, animation); however, he asserts that techniques like animation can be useful to detect errors at both conceptual (validation) and implementation (verification) level. In that sense, this is also a way of validating a conceptual model. Moreover, he places conceptual model validity in the category of *white box* testing where validation of the model can be performed through common sense and direct observation. In this respect, he believes that animation is a good mean to obtain face validity of white box simulation models (Kleijnen, 1995).

A simulation conceptual model life cycle presented by Balci and Orsmby (2007) contains various V&V activities that are integrated within the whole development phase of a simulation conceptual model. It is to be noted that this lifecycle was primarily used for a large-scale project and seems reasonable for a project on that scale. However, as we mentioned in section 2.2.1, Balci's view is more about generating several communicative models in a way that could be reused for large-scale projects, thus saving time and resources by using a repository of models. Naturally, this intention has also been reflected in V&V techniques for CM. Their proposal contains three sub-models of a conceptual model, namely high-level conceptual model, detailed-level conceptual model and integrated conceptual model before a final conceptual model specification can be achieved. There are two aspects of quality mentioned for each activity. One is the quality of work product produced, and second is the quality of the process used to generate the output from previous input. The decomposition and integration process itself has to go through a V&V process before a final conceptual model can be produced. It is due to the nature of such large-scale projects, that Balci's life cycle introduces the sub-models to handle the problem of complexity in a conceptual model. Finally, in addition to an overall

V&V, authors have mentioned *maintainability* and *reusability* as a final quality check which again is justified for a large-scale project.

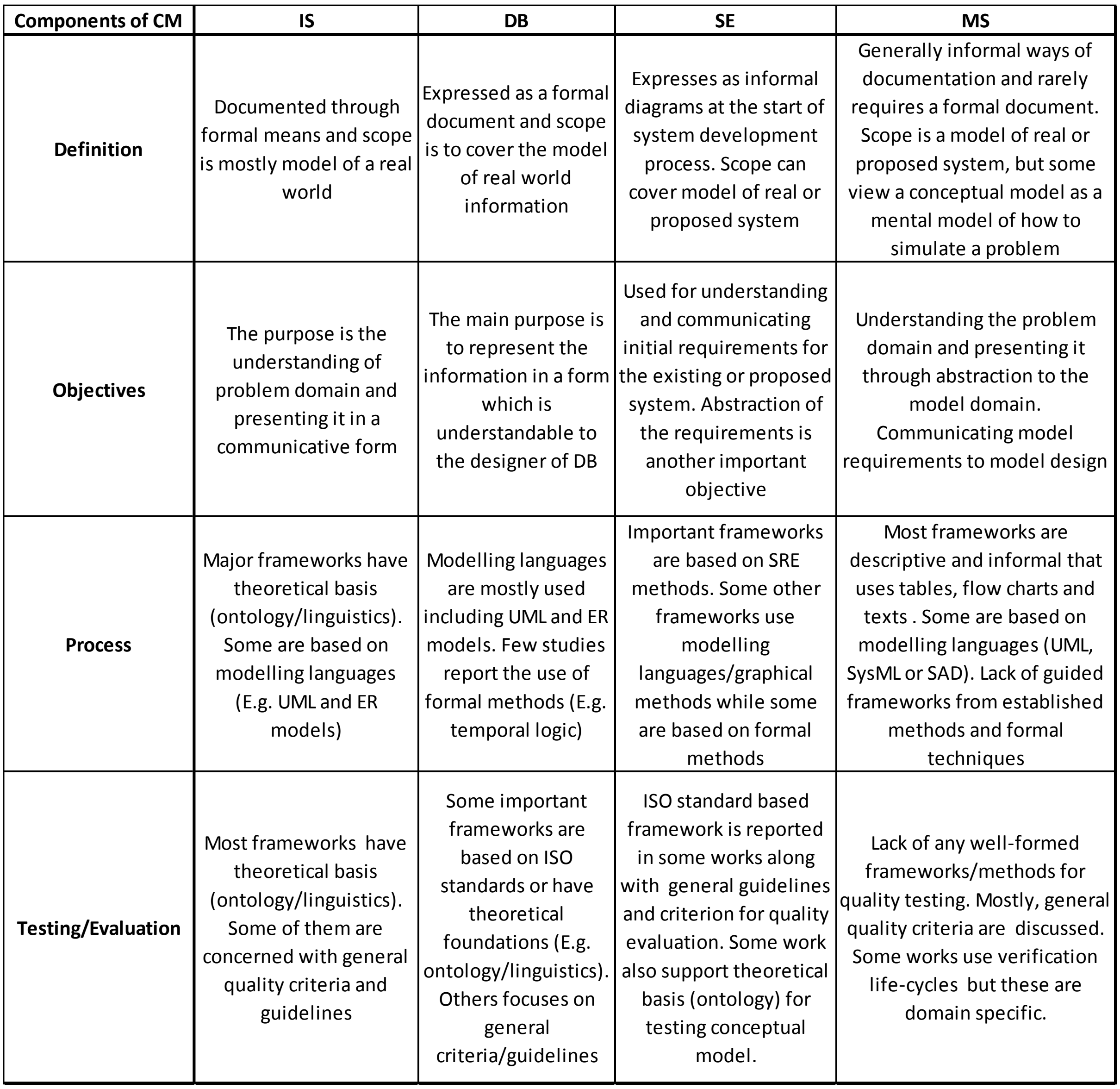
2.2.5 Summary of CM within M&S

The definition of CM is mostly documented informally with the scope of CM covering the real or proposed system. Some studies also define the conceptual model as a mental model of the problem being considered. Understanding the problem and abstracting the model from problem domain are reported to be the main objectives of CM. Conceptual models are also taken as serving the purpose of communication among different stakeholders. For the processes, mostly descriptive and informal methods are reported. Few approaches are based on modelling languages (such as UML, SysML or SAD). For the quality evaluation framework, mostly general guidelines and quality criteria are presented.

2.3 Conclusions

Our discussion on these components and characteristics is summarised in table 2.1, and the contents of the table can be used to draw conclusions. For example, we can conclude from information in the process row and within the SE column that the majority of the processes for CM are based on modelling languages, followed by some structured SRE methods, with very little support from formal methods or mathematical formalism. Similarly, other grids of the table can be interpreted and compared with each other for quick analysis of the literature on each aspect of CM between CS (IS,DB,SE) and M&S. An overall impression from table 2.1 suggests that CS holds more standard procedures and perspectives for CM in comparison to M&S across all four components.

Table 2.1 *Comparison of different components in CM across CS and M&S*



Components of CM	IS	DB	SE	MS
Definition	Documented through formal means and scope is mostly model of a real world	Expressed as a formal document and scope is to cover the model of real world information	Expresses as informal diagrams at the start of system development process. Scope can cover model of real or proposed system	Generally informal ways of documentation and rarely requires a formal document. Scope is a model of real or proposed system, but some view a conceptual model as a mental model of how to simulate a problem
Objectives	The purpose is the understanding of problem domain and presenting it in a communicative form	The main purpose is to represent the information in a form which is understandable to the designer of DB	Used for understanding and communicating initial requirements for the existing or proposed system. Abstraction of the requirements is another important objective	Understanding the problem domain and presenting it through abstraction to the model domain. Communicating model requirements to model design
Process	Major frameworks have theoretical basis (ontology/linguistics). Some are based on modelling languages (E.g. UML and ER models)	Modelling languages are mostly used including UML and ER models. Few studies report the use of formal methods (E.g. temporal logic)	Important frameworks are based on SRE methods. Some other frameworks use modelling languages/graphical methods while some are based on formal methods	Most frameworks are descriptive and informal that uses tables, flow charts and texts . Some are based on modelling languages (UML, SysML or SAD). Lack of guided frameworks from established methods and formal techniques
Testing/Evaluation	Most frameworks have theoretical basis (ontology/linguistics). Some of them are concerned with general quality criteria and guidelines	Some important frameworks are based on ISO standards or have theoretical foundations (E.g. ontology/linguistics). Others focuses on general criteria/guidelines	ISO standard based framework is reported in some works along with general guidelines and criterion for quality evaluation. Some work also support theoretical basis (ontology) for testing conceptual model.	Lack of any well-formed frameworks/methods for quality testing. Mostly, general quality criteria are discussed. Some works use verification life-cycles but these are domain specific.

A closer look at table 2.1 also reveals some interesting features with respect to our area of interest, M&S. M&S definitions of CM are mostly informal and can include real or proposed systems. This perspective is closest to the SE perspective of CM. The objectives of CM within M&S are mostly reported to be related to abstraction (hiding irrelevant details when transforming information for analysis purposes). This view is again close to the SE view of abstraction of CM. This similarity between SE and M&S provides the basis for adaptation of SE methods into M&S CM. This adaptation is required for the *process* and *evaluation* components of CM within M&S, since the other two parts (definition and objectives) hold similar perspectives and practices along the lines of SE. Processes for CM within M&S are mostly descriptive in nature, with little work based on modelling languages. This is in contrast to SE, where we see mostly SRE methods, modelling languages, or use of formal methods techniques. Similarly, CM testing within M&S is lacking, there being few well-formed methods for evaluating quality, unlike SE, where mostly quality standards are defined for CM and some quality evaluation frameworks have a strong theoretical basis.

This discussion leads us in a clear future direction. Some urgent work is required in the process and testing areas of CM within M&S along the lines of SE so as to gain the same benefits that are currently available in the latter field (mainly consistency, formality, communication and evaluation). We only address the process issue for CM in this thesis and not testing, although it is anticipated that with the development of a structured framework for the CM process, better testing procedures could be devised. The phase of the SE that is most relevant for this purpose is SRE. Moreover, as discussed in section 2.1.3.3, SRE activities and objectives mostly overlap with CM (Insfran et al., 2002). According to Rolland and Prakash (2000), CM was practiced even before SRE came into existence. Among all the SRE methods, discussed in section 2.1.3.3, view-point oriented methods and IDEF family methods have been extensively used in requirements engineering, CM, or mix of both (Sommerville and Sawyer, 1997; Dickover et al., 1977; Ross, 1977; Kim et al., 2003; Aguilar-Saven, 2004; Andrade et al., 2004; Lglesias et al., 1999). This forms the basis of our research methodology and more details on selection of these methods are discussed in section 3.2. In chapters 4 and 5 of the thesis, these methods are considered for developing a CM framework in DES. We first discuss the research methodology (chapter 3), followed by the actual implementation of the adapted frameworks in the subsequent chapters (chapters 4 and 5).

3 Research methodology

This chapter describes the research methodology for each phase of this thesis. An overview of the methodology is provided with respect to the research questions identified in chapter 1. The overall aim of thesis as presented in chapter 1 is as follows:

"To explore how CM approaches in CS could be adopted and adapted for M&S CM"

The details of each stage of the methodology with respect to above aim are given in the chapters that follow.

3.1 Stage 1: Identification of parts of CS for M&S CM

This stage of the research deals with the research question:

- ***RQ1: Which part of CS is most appropriate for adaptation to M&S CM?***

This research question is answered in chapter 2. It is identified in chapter 2 that Software Engineering (SE) and specifically Software Requirements Engineering (SRE) phase has the closest alignment for CM in M&S. This analysis is summarized in table 2.1 of chapter 2. Also, it has been identified that among the four components of conceptual model (definition, purpose, process, and testing), process and testing of CM needs most urgent attention within M&S; they are far more developed in CS. The focus of this thesis is on the process aspect of CM. Testing is identified as an area for future work and is discussed in section 7.4.

3.2 Stage 2: Selection of method from CS for M&S CM

The research question related to this stage of thesis is as follows:

- ***RQ2: Which methods from CS could best be adopted for M&S CM?***

Following the review in chapter 2, in this thesis, two methods from SRE are proposed for adaptation into M&S CM. The review chapter has highlighted the extensive use of View-point oriented and IDEF/SADT methods for CM in SE and also discusses the potential of these methods to be adapted in CM within M&S (Details are provided in section 2.1.1.3 and section

2.3). Here we briefly discuss a rationale for selecting SRE methods in general, and then discuss the two specific SRE methods in detail.

The reason to select SRE methods from a variety of CS methods available is as follows. Firstly, SRE is analogous with simulation CM in terms of the tasks performed in the early stages of the projects in SE and M&S, such as requirements gathering and problem structuring. Secondly, SRE methods are more flexible in terms of semi-informal and structured approaches with a wide range of choices available for adaptation into M&S CM. Other methods from CS are either rigid and lack expressiveness, or highly formal and mathematical that makes it difficult for adaptation into M&S CM. It is advocated in this thesis that M&S CM is best aligned with SRE in terms of the process itself and intended outcome of CM.

The two specific methods from SRE that are selected for M&S CM in this thesis are PREview (Sommerville and Sawyer, 1997) and Structured Analysis and Design Technique (SADT) (Ross, 1977). One of the main reasons to select PREview is because of its view point orientation approach which is close to the notion of stakeholders in M&S CM. Also it has good support for organising information in tabular form, which can be useful for M&S CM. SADT is selected because this method is very useful in SE to represent a large system as a part of an inter-connected hierarchy of functions. This breaking down of software system to structure the requirements and functionality has inspired its adaptation into M&S CM in this thesis. The idea is to benefit from the hierarchical and visual approach of the method into the system description and abstraction stage of M&S CM. In the following we give a brief introduction to these methods and some detail about how they might help in different stages of M&S CM.

PREView

PREView (Sommerville and Sawyer, 1997) is a SRE method that focuses on gathering initial requirements from all possible stakeholders for development of a software system. When a system is considered from a single perspective, all the necessary information about the system cannot be discovered. A view-point based approach to requirement analysis ensures equal emphasis on the requirements of all stakeholders' involved in the system. For example, developing a solution to a railway system from the perspective of only railway drivers would be unsatisfactory for pedestrians crossing the line or security personnel staff responsible at those crossovers (Sommerville et al., 1998). Similar to the process of requirements engineering from different view-points, a conceptual model needs to be devloped considering all possible stakeholders. This view-point can involve not only people, but it can be any physical component, legal standards, environmental concern, enterprise requirements or any other perspective that have an effect on the development of a conceptual model. Once all view-points and their requirements have been included in a system description, different levels of abstraction can be achieved by considering only relevant view-points for a particular modelling objective.

Although many different versions and methods have replaced PREview since the introduction of the method, the basic idea of view-point orientation is still alive (Andrade et al., 2004; Steen et al., 2004; Demraoui et al., 2014; Prakash et al., 2014). We aim to adapt the essence of this idea into simulation CM by adapting PREView for our research. The CM version of PREview is called PREmod in this thesis. PREmod uses basic templates and steps from PREview and additional details have been added to suit the simulation modelling context. Any unnecessary detail or component for PREmod has been left out in the adaptation from PREview and justifications for these changes are provided. Chapter 4 provides the detailed steps and methodology of PREmod.

Structured Analysis and Design Technique (SADT)

Another aspect of adaptation of SRE methods into M&S CM is the visual and hierarchical capability present in most of these methods. It is anticipated that such methods also have good potential for developing a conceptual model. SADT (Ross, 1977) is a semi-formal structured approach for SRE used in the early phases of software development. Since the introduction of the method several commercial packages have been built around this approach (IDEF0, IDEF1 etc.); however, we aim to focus on SADT as we find it flexible in terms of notations and steps and also it possesses all the basic qualities of the later methods. Moreover, since we are aiming at using it for simulation CM, the basic idea of SADT is more general and suitable for this case, in contrast to later versions of SADT which were built to handle specific software development issues. The M&S CM version of SADT is called SADT CM in this thesis and the detailed framework of SADT CM is presented in chapter 5. The aim is to achieve the benefits of the structured and visual approach of SADT along with the flexibility of the method to structure problems. SADT CM is modified from SADT in terms of a hierarchal approach and abstraction perspective, the details of which are provided with justification and examples in chapter 5.

3.3 Stage 3: Evaluation of the frameworks for M&S CM

This phase of the research is related with the evaluation of the developed frameworks and about how these methods would be adapted for M&S CM . The research question related to this stage is as follows:

- ***RQ3: How should these methods be adapted for M&S CM***

This question is answered in two parts. The first part deals with feasibility testing. Semi-real test cases have been used for preliminary testing of both frameworks. The test cases have been selected to evaluate the following aspects of a CM framework:

- Modelling of System Description (SD)
- Abstraction of conceptual model from this SD
- Modelling of physical components in DES problem
- Interaction between different components of a DES conceptual model
- Focus on the modelling objective of the DES problem.

The above mentioned criteria are used in this thesis to assess the suitability of the framework. The selected test cases had the potential for assessing these important aspects of a CM framework. The first two points (system description and abstraction of the conceptual model) are considered as the main issues in M&S for CM and these are discussed in detail in section 2.1.2.3. Also, as already stated in section 2.2.3, the modelling of the physical components in a DES problem (entities, resources, queues etc.) and their interaction has not been the subject of most existing frameworks, hence the test cases have been selected to evaluate this property of the CM framework. In either of the two approaches selected in this thesis (view-point orientation and structured analysis), the main objective is to focus on the modelling objective throughout the entire CM process. More details on the significance of driving the framework with a modelling objective is provided in subsequent chapters where PREmod and SADT CM frameworks have been discussed in detail.

Also the test cases have been selected to test the feasibility of the framework in both the product and service operations. For instance, the aspects of waiting time, resource availability etc. are primarily tested in the service operation cases and the product aspects such as throughput from the system, analysis of buffers etc. are tested from the product test case. Lastly, in the case of SADT CM, the test cases needed to evaluate the hierarchal methodology are adapted in the thesis in the visual display of the inter-connected processes.

The problem descriptions for test cases used for PREmod are provided in chapter 3 (section 4.3.1 and section 4.3.2), while the test cases used for SADT CM are included in appendix 2. The approach is for the author to apply all steps of PREmod and SADT CM on the test cases. These results are analysed to identify the elements of the conceptual model both with respect to the system description and the abstracted conceptual model.

The process of developing the framework was iterative. Every step of the framework was checked and discussed with the thesis supervisors before moving onto the next step. In the event of an unsatisfactory result or inadequate justification of the step, that part of the framework was redeveloped and cross-checked. It was only after there was a consensus over the final solution that all steps of the framework were finalized for the second part of the evaluation. Practically, adaptation of the framework was carried out by defining each step and applying it on the test cases for validation. Hence, generation of the reference solution for the test cases and adaptation of the framework were interleaved processes and the reference solutions were generated alongside the agreed steps of framework.

The second part of the evaluation deals with the usability of the framework. First, pilot testing was carried out with research students. They were tutored with the frameworks with examples and then asked to apply the framework on one of the above test cases. The responses to the pilot testing were recorded and any necessary modification was carried out in the framework before conducting the main testing. In the case of PREmod, the method was taught using PowerPoint slides, while for SADT CM, a comprehensive user guide was developed to aid this tutoring phase. In the case of PREmod, it was sufficient to tutor the method with the Power point presentation as the method was not composed of a large number of detailed steps. However, in the case of SADT CM, a user-guide was necessary because the method was much longer and more detailed, hence using only power-point slides was not sufficient for the participants to keep track of every detail. Hence, it can be assumed that the people that were tutored with power-point slides for PREMod were not in a less more advantageous position than those that were provided with user-guide for SADT CM since the latter framework was more detailed and it required a more detailed guide.

The pilot testing was successful with SADT CM but showed significant issues with PREmod and so the final testing with PREmod was not carried out. Following a successful pilot test for SADT CM, the framework is subjected to final testing. In the final test, three different research students were tutored with all the steps of the framework before asking them to apply it on the test case. Complete results were recorded from the participants and the results were thoroughly analysed by comparing them with the reference solution generated in the feasibility test. The participants were then asked to provide feedback on the framework and the responses are documented in section 6.5. It is important to mention that due to time limitations, only three researchers were used to carry out the final evaluation of the framework. Moreover, no usability testing has been reported in the literatures for all CM frameworks discussed in section 1.4 and section 2.2.3. One of the main reasons is that due to the nature of the CM framework it is difficult to teach in an experimental setting within time limitations. This study is the first in the existing CM literature to have tested the CM framework proposed.

3.4 Summary

This chapter has briefly outlines the methodology for the different phases of this research and related to the research questions for the thesis (chapter 1). Mainly, this research can be divided into three phases as explained in this chapter. Chapter 2 is related with the first question, where the literature review has informed the direction of the research. The chapters that follow address the second and third research questions where development and testing of the CM framework are discussed in detail based on specific CS methods. Chapter 7 summarises the findings from this work and possible future research directions.

4 PREmod (framework for conceptual modelling)

This chapter describes a framework for CM given the name “PREmod” (meaning a PRE-modelling phase). The method is inspired from the Software Requirements Engineering (SRE) method “PREview” proposed by Sommerville and Sawyer (1997). The method is based on the View-Point (VP) oriented approach for developing software systems. Originally, the method was called VORDS (View point Oriented Requirements Development) from Kotonya and Sommerville (1996) but with some modification, it was developed into “PREview”. In this chapter, an adaptation of this method for developing a conceptual model within M&S projects is proposed. The original method has been modified to suit the M&S domain. A justification for making these changes is provided in this chapter. It is argued that the PREmod framework helps in both phases of CM, i.e. generation of a system description and abstraction of a conceptual model from this system description. This chapter aims reports the research into PREmod from its original idea and modification, through framework development, to application on test cases and pilot tests.

The rest of the chapter is structured as follows:

- Introduction to PREview
- Adaptation of PREview into PREmod
- Application of PREmod on test cases
- Pilot testing of PREmod
- Conclusion

4.1 Introduction to PREview

PREview (Sommerville and Sawyer, 1997) is a multi-perspective SRE approach where requirements of a software system (to be built) are gathered from different perspectives. It is not the first view point oriented approach in software engineering and several other approaches have been discussed in the literature, (e.g. See Sommerville and Sawyer, 1997). However, PREview claims to be more flexible and generalized in its approach when it comes to driving the requirements of the software systems in practical setting. Also, it is shown to be compatible with other SRE methods. PREview was developed as a result of enhancing requirements elicitations and negotiations rather than system specification itself (Sommerville et al., 1998). PREview focuses on gathering initial requirements from all possible stakeholders for development of a software system. When a system is considered from a single perspective, all the necessary information about the system cannot be discovered. A VP based approach to requirement analysis ensures equal emphasis of all stakeholders' requirements involved in the system. For example, developing a solution to a railway system from the perspective of only "railway drivers" would be unsatisfactory for "pedestrians" crossings or security personnel staff responsible for those crossovers (Sommerville et al., 1998).

The key characteristics of PREview are:

- Requirements (for a single VP) may be expressed in any preferred notation, including natural language, structured or formal approaches, tables and diagrams.
- Concerns in a VP are the main driving force for requirement analysis
- All VPs must have their scope and their perspective explicitly defined

(Sommerville and Sawyer, 1997)

A VP defined in PREview has the following information as explained in Sommerville and Sawyer (1997) and (Sommerville et al., 1998).

- The VP name: This is used to refer any VP in the system.
- The VP focus: defined the scope of the VP and its distinct perspective in the system.
- The VP concerns: reflects high-level requirements and business goals
- The VP sources: provide information about specific sources for this VP generation.
- The VP requirements: These are the requirements generating from the VP's focus.
- The VP history: This records changes of the VP history for traceability purposes.

Apart from the "history" field, all other definitions are retained for adaptation to PREmod. The reason being the history is less relevant in the case of simple systems which we intend to test PREmod with. However, this information could be very useful and can easily be added in the later stages of research if required. It is also important to understand the difference between the focus and concern fields. While *focus* defines the original purpose and scope of the VP in the system, the *concerns* are related to the high-level business goals of the organisation. They are used to define explicitly the strategic goals of the organisations which are further sub-divided into sub-concerns and questions that can be cross-checked with the *focus* of the VPs to ensure consistency (Sommerville and Sawyer, 1997). In this way, we can say that *concerns* are used to drive the requirement analysis procedure for all VPs. An example of this approach is shown in figure 4.1 where the three major *concerns* are shown to drive the other VPs in a train control system. This being the case, it is important to identify the *concerns* first before going on to find all the other information in a VP template.

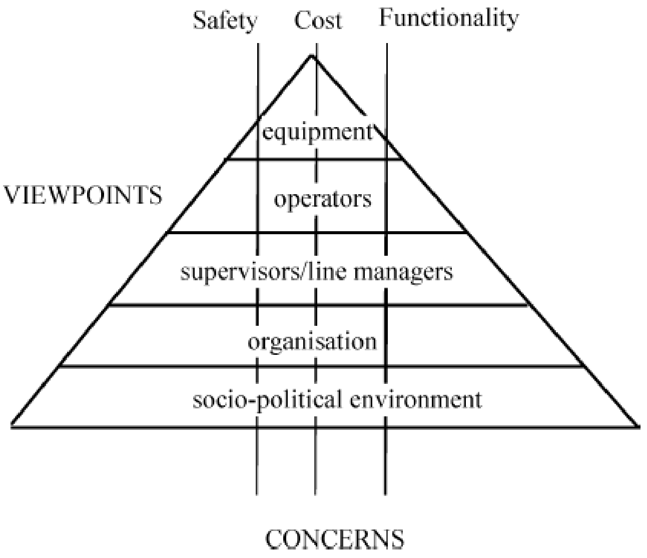


Fig 4.1: The orthogonally of Viewpoints and concerns in a train control system (Sommerville and Sawyer, 1997)

Next, a brief overview of the process in PREview is presented. It is not intended to give a detailed explanation as we are mainly interested in the basics of the method so it can be linked with the framework of CM in the next section. The PREview process can be explained with the help of figure 4.2. PREview starts with the phase of requirements discovery where the concerns are first identified. These concerns are broken down into further sub-concerns and any particular questions related to the concerns are documented. After this step, VPs are identified in the system and all the information about name, requirements, focus etc. are gathered in the VP template. The next phase is the analysis of the VP interactions and removal of any inconsistencies found within the requirements of the VPs. This can be performed by visual comparison of the requirements within each VP template or a suitable tabular method can be used where the checklist of all requirements' compliance is recorded (Sommerville and Sawyer, 1997). This must be accomplished by comparing individual foci of the VPs with the

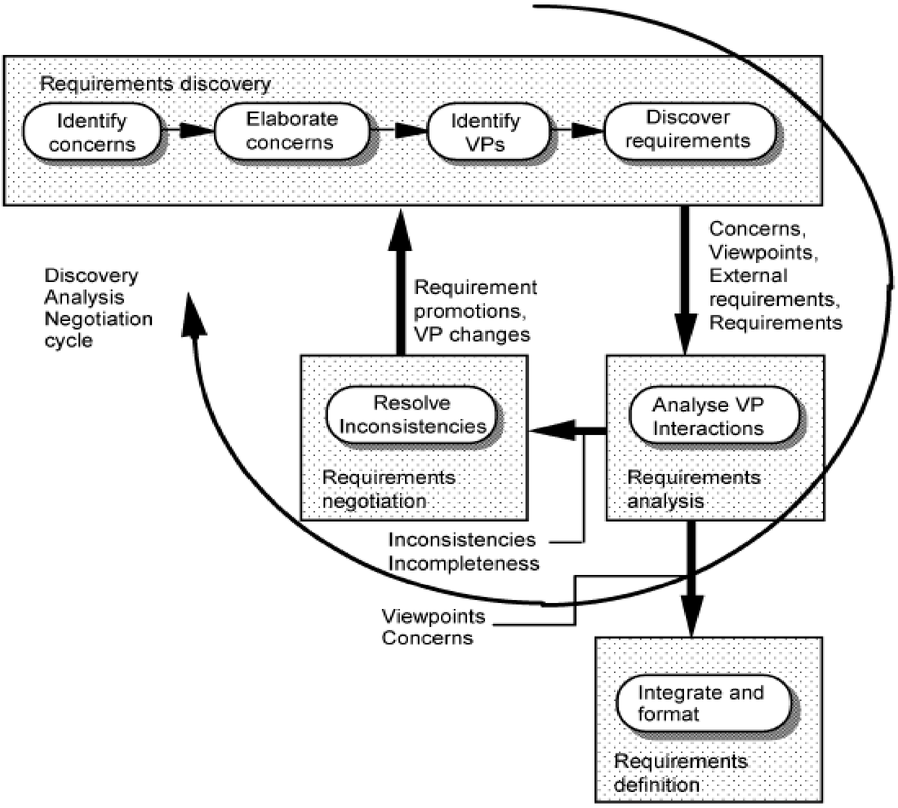


Figure 4.2: The PREview process (Sommerville and Sawyer, 1997)

global *concerns* as explained in Sommerville et al., (1998). Even in the case of no inconsistency among the VPs themselves, the most important question is "Does a VP requirement affect in any way the *concern* of the system?" For this purpose, an additional step called "concern decomposition" is usually performed when elaborating the concerns, so that each VP requirement can easily be compared to all refined high-level business goals. After negotiations on requirements conflict and re-iterating within the cycle (if necessary), the final version of the requirements document is produced in accordance with the concerns of the system (Sommerville and Sawyer, 1997).

4.2 Adaptation of PREview to PREmod

We explain the adaptation of the PREview method to PREmod with the illustration of a small call centre example whose problem description is given first. We also intend to explain the key differences and any modification in the method as we go along explaining the framework.

4.2.1 Call Centre Problem

Two types of calls are made to the call centre during a busy hour of the day. One from customer type X (standard calls) and the second from customer type Y (emergency calls) enters the system. X calls arrive at a rate of every 3 minutes while Y calls arrive at a rate of one every 10 minutes. Both Calls are queued before an automatic router until there is space in the router to process each call. The layout of the process is shown in figure 4.3.

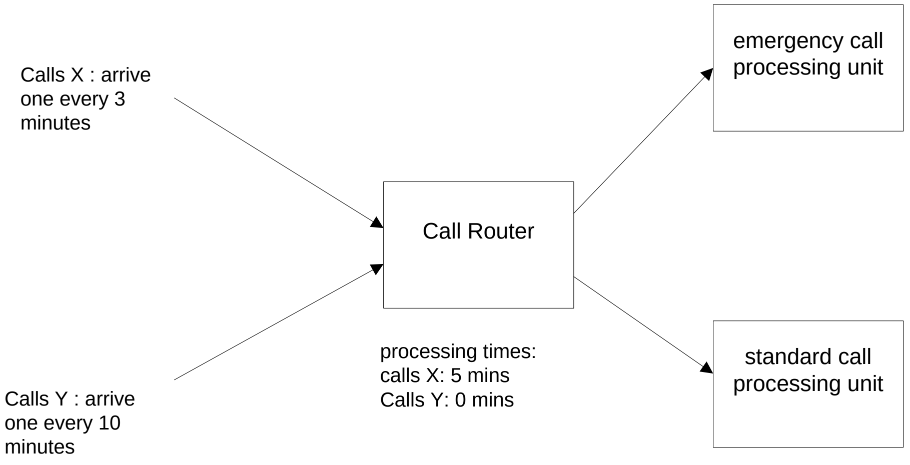

Calls X : arrive one every 3 minutes
Calls Y : arrive one every 10 minutes
Call Router
processing times: calls X: 5 mins Calls Y: 0 mins
emergency call processing unit
standard call processing unit

Figure 4.3: Layout of the processes in call centre problem

Calls Y, being emergency calls, will be taken out of the queue to process immediately as space become available in the router. The router then processes this emergency call without any delay (ignoring very small processing time) and sends it to the respective emergency call unit. Calls X, being standard in nature, has to wait in the queue until there is no Y call waiting for processing. The router then takes 5 minutes to process the standard calls and sends it to respective standard call unit. Management of the call centre are concerned about the average waiting time of the X calls and also the maximum waiting time for Y calls. Besides this they are also interested in investigating if the service quality of the call is affected by the number of calls answered (total calls lost during any busy hour). A call can be lost if there is no room for the call to enter the router, or if the waiting time of the call is large and so the customer decides to drop the call. Management is considering investing in an extra room for the call centre with more employees if the service quality and waiting times are not to the required standard. The required standard for average waiting time for call X is 2 mins and maximum waiting time for call Y is 1 minute. Management wants to have 85% of calls answered from the total calls received during any busy period.

4.2.2 PREmod Process

There are six main steps in PREmod:

- Step 1: Identification of concerns
- Step 2: Concern decomposition
- Step 3: View point identification
- Step 4: View point analysis stage 1
- Step 5: View point analysis stage 2
- Step 6: Integration of VPs

Next, each of these steps will be explained with reference to the call centre problem.

4.2.2.1 Step 1: Identification of the Concerns

As in PREview, concerns are the main high-level objectives of modelling in PREmod. They are decomposed into particular modelling questions in the next step. The difference is, in PREview, concerns represent the high-level requirements of the system from the management VP, whereas in PREmod, the concerns reflect the modelling objectives of the modelling project. However, they also represent the interests of the managers and decision makers in the modelling process. In some cases, concerns can also arise from customers and clients but it depends upon the nature of the problem being modelled. In our case of the call centre problem, concerns are at the managerial levels which are decomposed into modelling questions in the next step.

The main concerns for the call centre problem are identified as follows:

- Waiting time of calls
- Service quality
- Resource requirements

4.2.2.2 Step 2: Concern Decomposition

In this step, concerns are decomposed into further modelling questions using a decomposition chart as shown as shown in figure 4.4.

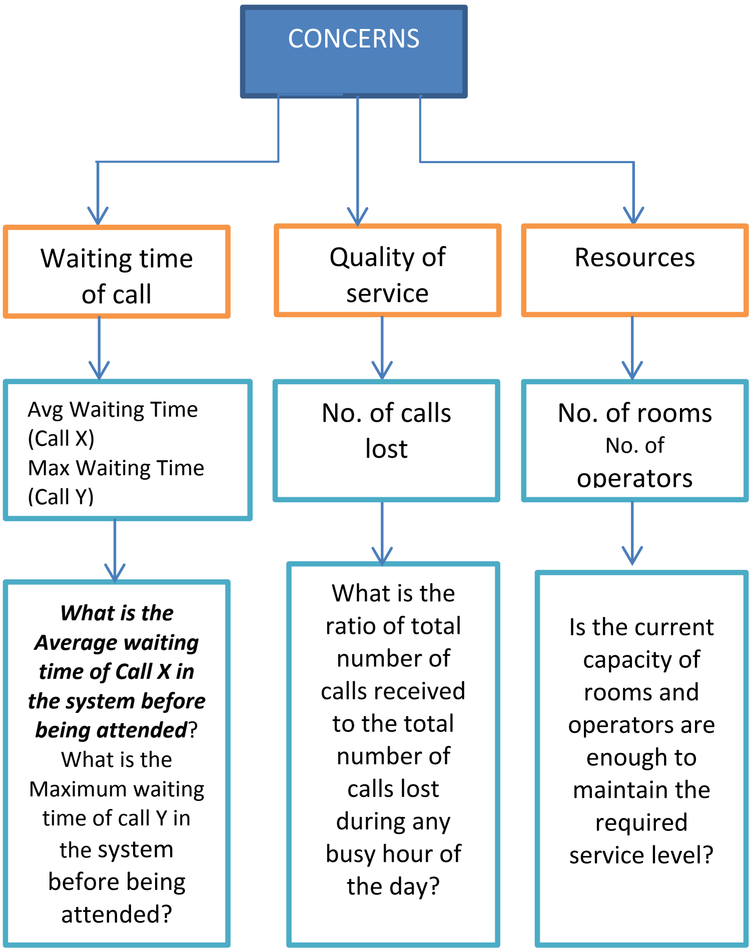

CONCERNS
Waiting time of call
Quality of service
Resources
Avg Waiting Time (Call X)
Max Waiting Time (Call Y)
No. of calls lost
No. of rooms
No. of operators
What is the Average waiting time of Call X in the system before being attended? What is the Maximum waiting time of call Y in the system before being attended?
What is the ratio of total number of calls received to the total number of calls lost during any busy hour of the day?
Is the current capacity of rooms and operators are enough to maintain the required service level?

Figure 4.4: Concern Decomposition for the call centre problem

In figure 4.4, concerns identified in step 1 for the call centre problem are decomposed into particular modelling questions using the information given in problem description. There can be multiple concerns in a system and similarly multiple objectives. For example, In PREview, several VP requirements can be derived under the concern of safety or compatibility when designing a software system for a train control system (Sommerville et al., 1998). Similarly within M&S, many VPs can be derived for any concern. For example, to model an ATM machine for checking the service level over a certain period of time, different VPs would be derived than for the concern of waiting times of customers. However, there can be certain overlaps between the VP lists derived under different concerns. In PREmod, we work with a single concern at a time. This is to focus completely on one concern at a time and to show how the abstraction process works with relevance to a single concern. Thereafter, with little alteration, other concerns can also be modelled by repeating steps 3 to 6 in the PREmod process. Most information can be reused easily when dealing with multiple objective problems in PREmod. In general, within M&S, we see that the model created for a single purpose can be altered to address other issues also. This can be achieved with little or no modification in the model. In some cases, however, major model modification is required. The advantage of working precisely with the main objective of the modelling is to reduce some modelling burden. This is demonstrated in PREmod approach.

In the case of call centre problem, we decide to work with the decomposed concern of *Time* stating:

“***What is the Average waiting time of Call X in the system before being attended***?”

4.2.2.3 Step 3: View Point identification

In this step, all possible VPs are identified from the problem description. It is important to note that any VP that even seems irrelevant with the concern at this stage should also be identified. This will ensure that the whole system description has been taken into account. Every component in the problem description is treated as having a VP. Taking advantage of the flexibility of the method, it is useful to list as many VPs as possible at this stage even if they do not appear to have a stake in the modelling project. The initial VP list can be refined at later stages. The following components can be treated as having a VP and serve as a useful guideline in identifying different VPs in a system:

- Entities (static, dynamic, fix, moving etc.) e.g. cars, people, houses, calls etc.
- Activities (processing units, machines, consultation units, service counter etc.) e.g. welding machines, patient consultation rooms etc.
- Queues (storage areas, waiting areas etc.) e.g. waiting queue of patients in hospitals, customers waiting in queue for ATM, cars waiting on a road.
- Standards (policies, limitations, governmental regulations etc.) e.g. health and safety, national speed limits, security policies, constraint from city council etc.

For the call centre problem, the following VPs have been identified:

- Call Centre
- Calls_X
- Calls_Y
- Q_router
- Router
- Emergency call unit
- Standard call unit

"Q" is added to indicate a queue preceding the activity it represents. Any form of naming convention can be used; however, it is useful to give meaningful names to the VPs for easy identification in the later stages.

4.2.2.4 Step 4: View point analysis stage 1

In this step, initial screening of VP list is carried out. For this purpose table 4.1 is used in the call centre problem. Each VP is compared with the concern and decision is made to include or exclude the VP. The justification of the VP being included or excluded is also recorded. The decision to retain any VP depends on its relevance to the concern. Either a VP directly affects the concern, e.g., a queue whose entities' waiting time is to be measured, or indirectly effects it, e.g., an activity that is affecting the flow of entities into the concerned activity or queue. Similarly, a VP can be considered irrelevant to the concern if it appears after the activity or queue whose response is to be measured.

Table 4.1: View point analysis – stage 1 in the call centre problem

<table>
<tr><td colspan="3">CONCERN: What is the average waiting time of call X in the system before being attended?</td></tr>
<tr><td><b>VP</b></td><td><b>Decision</b></td><td><b>Justification</b></td></tr>
<tr><td>Call X</td><td>include</td><td>Directly effects the concern</td></tr>
<tr><td>Calls Y</td><td>include</td><td>May effect the concern indirectly as it can have an affect on waiting time of calls X</td></tr>
<tr><td>Q_router</td><td>include</td><td>Directly effects the concern as this is the queue where calls X are waiting</td></tr>
<tr><td>Router</td><td>inlcude</td><td>Directly effects the concern as waiting time of calls X is dependant on how quick a call is processed inside the router</td></tr>
<tr><td>Emergency call unit</td><td>Remove</td><td>Activity after the call is being attended and hence won't affect waiting time of calls X in router queue</td></tr>
<tr><td>Standard call unit</td><td>Remove</td><td>Activity after the call is being attended and hence won't affect waiting time of calls X in router queue</td></tr>
</table>

The VPs selected to be removed seem not to be relevant to the concern at this stage. It is important to note that, as this is the initial stage of screening, only those VPs that seem not to be relevant to the concern are removed. If there is a doubt about the decision to exclude a VP, then that VP is left in. The final list of the VPs to be included in further steps for the call centre problem is as follows:

- Calls_X
- Calls_Y
- Q_Router
- Router

4.2.2.5 Step 5: View point analysis stage 2

In this step we create templates for each VP. The structure of this template is shown in table 4.2. These templates make sure that details are put in the right structure. As we go through making these templates, we make further decision about excluding or including a certain VP from the model, based on the information recorded in the VP template.

The exclusion of a VP based on this template can either be a complete removal of the whole VP (for example based on some routing detail) or any particular information from the template. This information (focus, requirements, source or details) are primarily compared with the original concerns identified in step 2. This agreed concern is repeated in the header of every VP template, so any information is easily compared with this concern and any justification is suitably provided. In summary, these VP templates can be taken as second-level abstractions for VPs in the model. Comments are made within any field of the template to justify the decisions.

Table 4.2: Template for VP-analysis stage 2 in PREmod

Name	This contains the name of the VP
Concern	The main concern (objective) is repeated here on each VP template
Source	The source of this VP
Requirement	Requirements of this VP as seen in the modelling context
Focus	Focus of this VP in the modelling environment
Comments	Comments about this VP after comparing focus with concern
Decision	Decision to include or exclude this VP based on comments
Details	This field contains further details of a VP (if included) such as arrival pattern, quantity, routing details, breakdown, resources, cycle-time etc. If a VP is excluded in the decision field, then the details row is omitted

The template in table 4.2 contains the same fields as in the original PREView, but with additional fields to include the modelling information; the comments, decision and details fields have been added. The first two are added to check if any of the VPs still need to remain in the model after stage 1 of the analysis or if there is any information (for example focus or requirement fields) that can help us exclude the VP at this stage. Similarly, the details field contains specific low-level information about a VP; based on which any particular VP can be excluded from the modelling even if it was selected at stage 1. This is reasonable as stage 1 is a more high-level analysis where all we know is the name of the VP and no further details. However, as we create these templates, certain other useful information can arise leading to the exclusion of a VP or any of its low-level details. This flexibility in PREmod also allows a VP to be included again if any of the information acquired in the VP template demands for that. For example, a VP may have been excluded in stage 1 by comparison with the concern, however, later in the analysis, it becomes apparent that a particular VP routing detail requires entities from the excluded VP, and therefore the excluded VP could be added back along with its template. This inclusion and exclusion of a VP is illustrated with the call centre problem and also in the test case examples of section 4.3.

The requirement and focus fields in the templates also need some clarification. The requirements and focus details record the details from the modelling context and may not necessarily reflect the same perspective as in the PREview. This is because in PREView, any VP can have two, three or even more requirements and this forms the very basis of requirements gathering. On the other hand, the requirements of a VP in PREmod may not be very relevant but merely provide some useful background information. For example, in PREview, requirements of the VP *emergency braking* can be detection of excess speed, detection of overshooting and frequency of invocation (Sommerville et al., 1998), while in the PREmod , the requirement of a queue in the router template is hard to find (or at least hard to elicit). At best the detail that can be put up here is “accommodate as many calls as possible”. It is not conclusive at this stage that requirements and focus can be easily distinguishable for the users of PREmod. However, it still seems useful to have this in the initial design of the method.

Next, the VP templates of the call centre problem are presented and analysed. We repeat the list of the VPs selected for stage 2 analysis as follows:

- Calls_X
- Calls_Y
- Q_Router
- Router

What follows are the VP templates in tables 4.3 (a-e) for the above VPs in the call centre problem.

Table 4.3 (a): VP analysis in stage 2 for Calls_X

Name	**Calls_X**
Concern	Average waiting time of Call X in the system before being attended
Source	Customers (Standard callers)
Requirement	To be attended as much as possible .
	No calls lost (not required by concern)
Focus	Arrivals of calls type X
Comment	Relevant with concern
Decision	Include
Details	*****************************
quantity arrived	No. of calls arrived as given in the data during busy hour on Monday morning (for simplicity)
Arrival Pattern	Arrival distribution as per given data
Attribute	Calls type X
Routing	In: Entry point out:Q_router
Exp.Factors	Varying arrival patterns (Not required)
Outputs	No. of calls arrived, remained in system or lost (Not required)

Table 4.3 (b): VP analysis in stage 2 for Calls_Y

Name	**Calls_Y**
Concern	Average waiting time of Call X in the system before being attended
Source	Customers (emergency callers)
Requirements	To be attended as much as possible
	No calls lost (not required by concern)
Focus	Arrivals of call Y
Comment	irrelevant with concern (at this point it seems having no effect on waiting times of calls X as they are processed without delay)
Decision	***Exclude***

Table 4.3 (c): VP analysis in stage 2 for Router queue

Name	**Q_router**
Concern	Average waiting time of Call X in the system before being attended.
Source	Storage point for calls before going to router
Requirement	Accommodate as many calls as possible
Focus	Number of calls queued for processing
Comment	Related with concern
Decision	Include
Details	******************************
quantity	One
Capacity	Unlimited, assumption that no calls are balked
Dwell Time	Remain until space available in the router
Discipline	Priority based (Calls_Y have priority)
Break down/repair	N/A
Routing	In from : Calls_X, Calls_Y out to : Router (These details suggest routing calls Y also even if we require only waiting time of X calls)
Exp Factors	Queue capacity , dwell time (Not required)
Outputs	Waiting times (required by concern)

The routing details of Q_router suggest including VP of Calls_Y also in the modelling, so its template is recorded again and the details section is added. The name is updated from Calls_Y to calls_Y_2 just to differentiate between the two versions.

Table 4.3 (d): VP analysis (iterated) in stage 2 for calls Y

Name	**Calls_Y_2**
Concern	Average waiting time of Call X in the system before being attended
Source	Customers (emergency callers)
Requirement	To be attended as much as possible
	No calls lost (not required by concern)
Focus	Arrivals of call Y
Comment	Relevant with concern
Decision	Include
Details	******************************
quantity arrived	No. of calls arrived as given in the data during busy hour on Monday morning (for simplicity)
Arrival Pattern	Arrival distribution as per given data
Attribute	Calls type Y
Routing	In: Entry point out:Q_router
Exp.Factors	Varying arrival patterns (Not required)
Outputs	No. of calls arrived, remained in system or lost (Not required)

Table 4.3 (e): VP analysis in stage 2 for Router

Name	**Router**
Concern	Average waiting time of Call X in the system before being attended
Source	Processing point for calls . Main component of the system
Requirements	serve as many calls as possible
	No calls lost (not required by concern)
Focus	Processing of calls coming from Q_Router
Comment	related with concern
Decision	Include
Details	*****************************
quantity	One
cycle time	as given in data 5 min for X calls, 0 min for Y calls
Resource	N/A
Breakdown/Repair	Assuming no breakdown
Routing	IN: Q_router OUT: Exit (as Call Units are not modelled so this is a safe simplification)
Exp factors	Varying service time (Not required)
Output	no of calls processed (Not required)

It should be noted that requirement “No call lost” from all VP’s templates have not been included as this belongs to concern of “service quality”. However, it is anticipated that if we have to work with the concern of “service quality” at some point, these templates could be added with this information and the current templates adapted as required for the new concern.

4.2.2.6 Step 6: Integration of the VPs

The final step is to integrate the selected VPs into the process flow map. One way to do this is to link all VPs by checking their routing details. In our call centre example , if we look closely at the routing in and routing out detail of every VP (Calls_X, Calls_Y, Q_Router, Router), we can easily come up with the process flow diagram of figure 4.5. In this map, triangles represent the entities, oval shapes represent queues in the system and rectangles represent processes. A hexagon is used to indicate an exit point in the model.

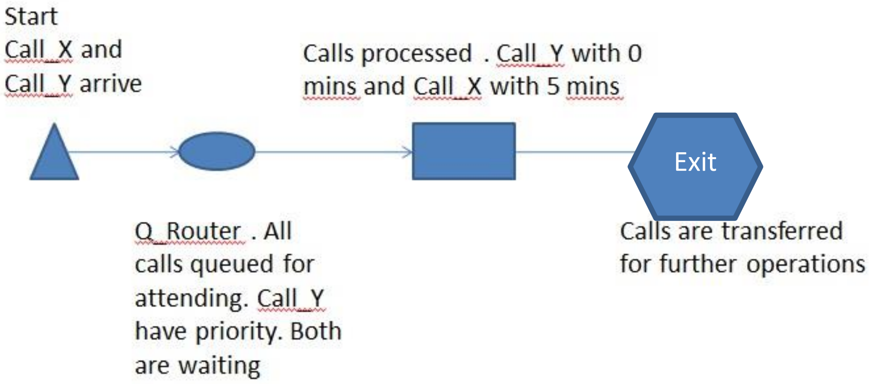


Figure 4.5: Process flow map of the call centre problem

4.3 Application of PREmod on Test Cases

In this section, we apply the framework of PREmod, on two selected test cases. This is done by the author to evaluate the framework. In section 4.4, a pilot test of PREmod is presented which was performed by three different participants. The first test case (Wardeon Cinema) is taken from Robinson (2014) and represents the modelling of a telephone system in a cinema. The second test case (Happy faces Daycare) is about modelling a children's nursery in a residential area which is based on a real modelling task. This was set up as a conceptual modelling challenge in M&S magazine (2010).

4.3.1 Wardeon Cinema problem

The problem situation is presented first, followed by the application of PREmod to develop conceptual model for the problem. The test case is slightly modified to include an additional concern for the purpose of demonstrating the approach for multiple concerns in PREmod. The problem situation is directly copied from Robinson (2014), and the addition has been highlighted in italic fonts without the quotes.

4.3.1.1 Problem Situation

"Wardeon Cinemas own and manage multiplex cinemas in the United Kingdom. Their Nottingham complex has been open for five years and is very successful in attracting clientele, despite fierce competition in the region from other cinema groups. Recently, however, there have been a number of complaints about the telephone enquiries and booking service.

"The telephone system was installed at the time the complex was opened. It provides a message service giving information on film times and an option to seek further information or book tickets by speaking to a booking clerk. Initially this service was adequate, but with rising demand the telephone queues are now often full, particularly on the busiest days (Saturdays). As a result, callers either balk (give up having received an engaged tone) or wait for some time and then hang-up (renege) having obtained neither the information nor the booking they require. Meanwhile, the booking clerks are regularly lambasted by irate customers who have waited up to 15 minutes or more for a response. The cinema's managers are obviously concerned about the loss of good will and custom.

"Wardeon Nottingham has decided to purchase and install a more modern telephone system. They have obtained various quotes for the system, the favoured choice being from a local company, Dodgey Phones Ltd. The proposed system uses digital technology, allowing customers to use their telephone keypads to choose from an automated menu system. As a result, it is believed that customers will be able to obtain the information they require from the recorded message service more rapidly. Further to this, the system will also provide facilities for booking tickets via the telephone keypad, without the need to speak to a booking clerk. Because some members of the public still wish to speak to a member of staff, particularly if they require further information or would rather book tickets in person, then there will be an option to speak with a customer service representative (CSR). Dodgey have also suggested increasing the capacity of the system from its present level and playing soothing music to customers should they have to wait for a response.

"Figure 4.6 shows a schematic of the system proposed by Dodgey. Calls enter the system via a call router (or automatic call distributor, ACD), where the callers enter a number 1, 2 or 3 on their keypad for the service they require (information, ticket sales or CSR respectively). Calls are then routed to the appropriate service. Having completed a call with one of the automated services, callers will be given the option to return to the call router to select another service. The plan is to install four lines for each of the services including the call router. It is unlikely, however, that Wardeon will employ four CSR staff; the current intention is to employ three staff during the busiest periods. Wardeon have the option to purchase extra lines for each service, in blocks of two, albeit at quite some additional expense.

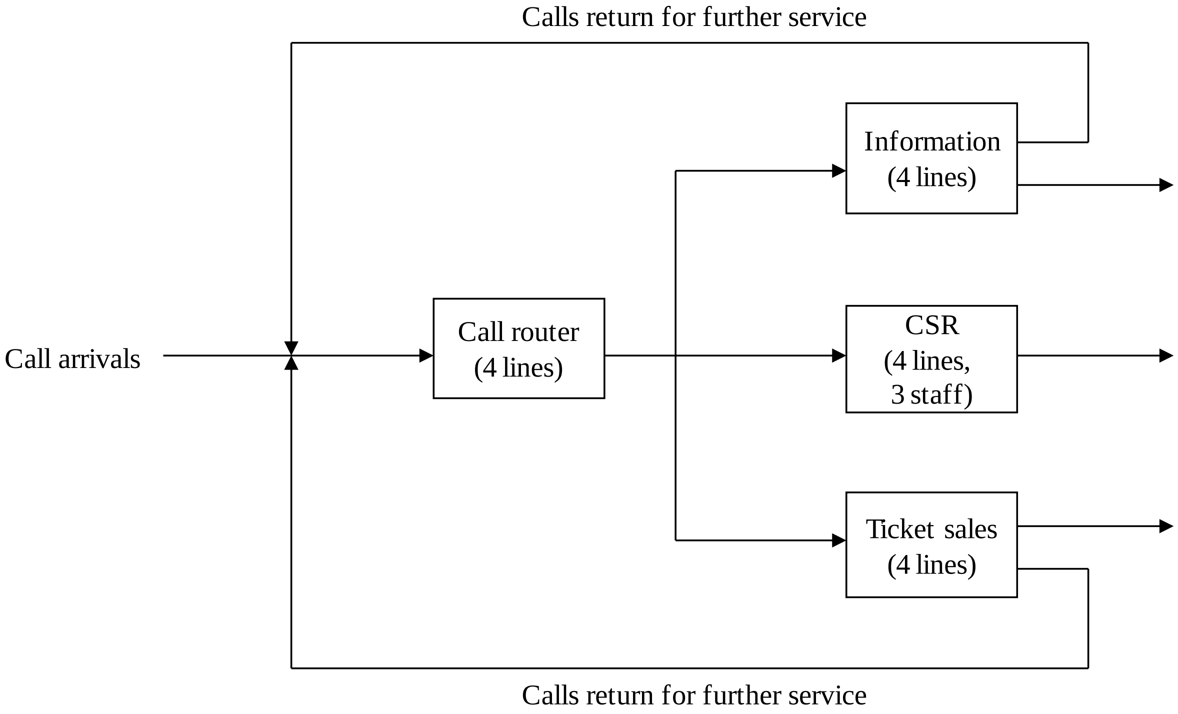


Figure 4.6 *Wardeon Cinema New Telephone System*

"Wardeon's aim is to resource the system sufficiently so that less than five percent of calls are lost (balk or renege), even on busy days. They also want the total waiting time; that is the sum of waiting times in all queues, to be less than two minutes for at least 80 percent of calls and the mean to be less than one minute. There are a number of concerns, however, with the system that is being proposed (apart from the name of the supplier!), in particular:

- Whether there are sufficient information lines.
- Whether there are sufficient ticket sales lines.

"Wardeon are quite happy with the current plan to have four call router lines because of the short time it takes calls to be handled by this part of the system. They would also like to know how many CSR staff to have available at different times of the day"

Robinson (2014, pp. 291-293)

As all the staff members are trained on the old system, they have a concern about the software interface on the system also. The system needs to be installed in the basement so they are also interested to know its feasibility and whether this will have any effect on the quality of the services (e.g. calls audibility, noise in the lines etc.)

"In the future Wardeon are planning to expand the Nottingham cinema complex. As a result, it is expected that the demand on the telephone system will continue to grow. They would, therefore, like to know the maximum capacity of the telephone system so they are able to identify the demand level at which further expansion will be needed.

The management board are meeting in three weeks and need to make a decision as to whether to go ahead with the Dodgey proposal, or not. The intention is to have the results of the simulation ready for that meeting. It should be noted that a number of the managers at Wardeon are sceptical concerning the need for a simulation. They have never used it before and will need convincing that the tool is useful"

Robinson (2014, pp. 370-371)

4.3.1.2 Application of PREmod

In the following, we present the six steps of PREmod for the Wardeon Cinema problem.

Step 1: identification of concerns

The concerns are identified by going through the problem description in detail and focusing on the required response from the model. The following concerns have been identified for the Wardeon Cinema problem. A brief explanation on each concern is also provided.

- Service Level (related to resources required, waiting times of call and no. of calls lost)
- Capacity (related to call capacity)
- Compatibility (related to compatibility of the new system with the current staff)
- Location (related to the service level of system as location can have an effect on it)
- Standard (related with the quality of the calls, audibility etc.)

Step 2: Concern Decomposition

Concern decomposition chart for Wardeon cinema is shown in figure 4.7.

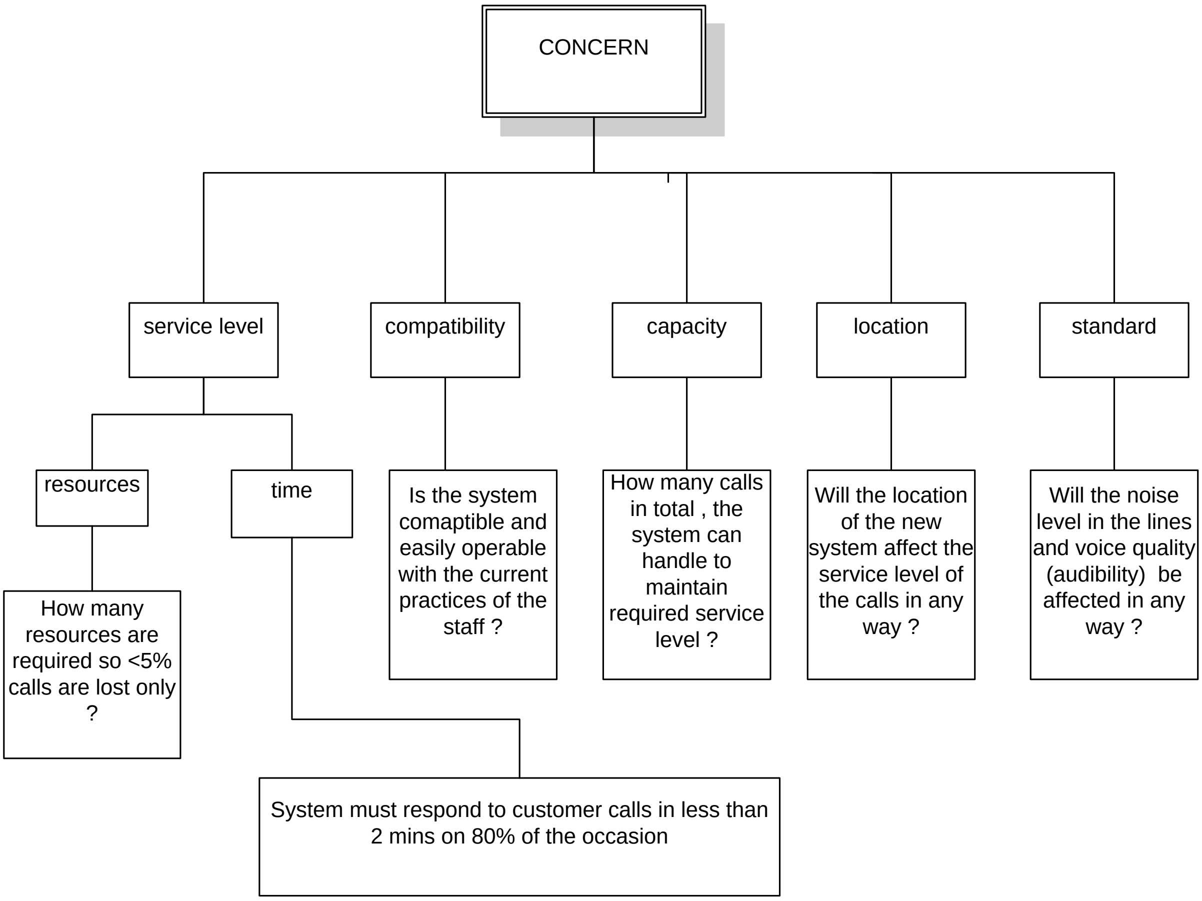

CONCERN
service level
compatibility
capacity
location
standard
resources
time
Is the system comaptible and easily operable with the current practices of the staff ?
How many calls in total , the system can handle to maintain required service level ?
Will the location of the new system affect the service level of the calls in any way ?
Will the noise level in the lines and voice quality (audibility) be affected in any way ?
How many resources are required so <5% calls are lost only ?
System must respond to customer calls in less than 2 mins on 80% of the occasion

Fig 4.7: Concern Decomposition chart for Wardeon Cinema problem

We aim to work with the concern of *service level* and leave out others for demonstration purpose as it is explained in section 4.2.2.2.

Step 3: View Point identification

12 view-points have been identified for Wardeon Cinema:

- Customer
- Call_router
- Information_lines
- Ticket_lines
- CSR
- Q_Call_router
- Q_Information_lines
- Q_Ticket_lines
- Q_CSR
- CSR(staff)
- GUI(software/hardware)
- Control room (basement)

Step 4: View point analysis stage 1

Table 4.4 shows VP analysis stage-1 for Wardeon cinema. All the VPs identified in step 3 are analyzed and the results along with any justification are also provided.

Table 4.4: VP analysis stage 1 for Warden Cinema

CONCERN: Service Level (resources, time)		
VP	**Decision**	**Justification**
Customer	Include	Main starting point of the process and arrival point of entities
Call_router	Include	Necessary activity to route calls correctly (affects the time concern also)
Information_lines	Include	Experimental factor (related to the resource concern)
Ticket_lines	Include	Experimental factor (related to the resource concern)
CSR	Include	Experimental factor (related to the resource concern)
Q_Call_router	Include	Relevant to time concern (contributes to the total waiting time)
Q_Information_lines	Include	Relevant to time concern (contributes to the total waiting time)
Q_Ticket_lines	Include	Relevant to time concern (contributes to the total waiting time)
Q_CSR	Include	Relevant to time concern (contributes to the total waiting time)
CSR(staff)	Include	Experimental factor (related to the resource concern)
GUI(software/hardware)	Exclude	Does not affect concern of resource or time
Control room (basement)	Exclude	Does not affect concern of resource or time

Step 5: View point analysis stage 2

In this step all the VP templates for the Wardeon cinema are provided in tables 4.5 (a-j). First, the updated list of the VPs selected for analysis in stage 2 is given:

- Customer
- Call_router
- Information_lines
- Ticket_lines
- CSR
- Q_Call_router
- Q_Information_lines
- Q_Ticket_lines
- Q_CSR
- CSR(staff)

Table 4.5 (a): VP analysis in stage 2 for customer

Name	**Customer**
Concern	Service Level (Resources, Time)
source	Customer
Requirements	Calls no longer than 2 mins
	Calls do not balk (only achievable at proportion consistent with concern)
	Clear voice quality (Not directly relevant with concern)
	CSR available if required
Focus	Initiator of the call process
Comments	Relevant with concern
Decision	Include
Details	************************************
quantity arrived	As determined by arrival pattern
Arrival Pattern	Inter-arrival times as set by suitable distribution varying at different hours of day
Attribute	Determines route of call to CSR, inf_lines etc.
Routing	In: Entry point Out: Q_Router
Exp Factors	Varying call rate arrivals (Not required)
Output	No. of calls completed, No. of calls balked or dropped, No. of calls directed to CSR (Not monitored any of these as not required by concern)

Table 4.5 (b): VP analysis in stage 2 for call Router

Name	**Call router**
Concern	Service Level (Resources, Time)
Source	Telephone system design, Management Wardeon cinema
Requirements	provide good sound quality and presentation to customer (Not directly effecting the concern of service level)
	Accurately present different choices to the customer
	Accurately records the customer choice
	Provides accurate routing to selected services 1-3
	Must be compatible with all digital keypad telephone system (Not directly related with concern)
	Must be able to handle calls at maximum capacity during busy hours (Not directly related with concern)
Focus	Accepts calls and routes to the required service
Comments	Relevant with concern
Decision	Include
Details	*************************
Quantity	Single router
Cycle_time	As determined by processing time distribution
Resource	None
Breakdown/Repair	n/a
Routing	In: Q_Router out: different activities as required
Exp Factors	Varying processing time (Not required)
Output	Calls lost during processing (Not required)

Table 4.5 (c): VP analysis in stage 2 for Information lines

Name	**Information lines**
Concern	Service Level (Resources, Time)
Source	Telephone system design and managers of Wardeon Cinema, customers
Requirements	Provide accurate information to customers
	Able to meet capacity requirement of the calls (Not directly related with concern)
	Provide the option to return for further service or end the call.
Focus	Handles incoming calls from router and provide specific information
Comments	Relevant with concern
Decision	Include
Details	**
Quantity	Four information lines
Cycle_time	As determined by processing time distribution
Resource	None
Breakdown/Repair	N/A
Routing	In: Router out: router/Exit model
Exp Factors	Varying processing time for different nature of information (Not required)
Output	No. of calls lost during this process (Not required)

Table 4.5 (d): VP analysis in stage 2 for Ticket sales lines

Name	**Ticket Sale lines**
Concern	Service Level (Resources, Time)
Source	Management of Wardeon Cinema
Requirements	Provide update pricing to the customer
	Process purchase orders
Focus	Manages ticket purchase for the customer over the phone
Comments	Relevant with concern
Decision	Include
Details	******************************
Quantity	Four Ticket lines
Cycle_time	As determined by processing time distribution
Resource	None
Breakdown/Repair	N/A
Routing	In: Router out: router/Exit model
Exp Factors	Varying processing time for different nature of orders (Not required)
Output	No. of calls lost during this process (Not required)

Table 4.5 (e): VP analysis in stage 2 for CSR

Name	**CSR**
Concern	Service Level (Resources, Time)
Source	CSR , management of Wardeon cinema
Requirements	Provide all general information not available through automated service
	Provide manual ticket booking service
	Effective communication skills (Not directly related with concern)
Focus	Manual ticket purchase, price updates and any general information
Comments	Related with concern
Decision	include
Details	*********************************
Quantity	Four CSR lines
Cycle_time	As determined by processing time distribution
Resource	Three CSR staff (may not be needed to model explicitly if only three lines are to be utilized , as they serve as dedicated resource then)
Breakdown/Repair	N/A
Routing	Exit from the system after completing the purchase orders
Exp Factors	Varying time to model different processing times for different purchase orders (Not required)
Output	Average and maximum time to process an order (Not required)

Table 4.5 (f): VP analysis in stage 2 for Router Queue

Name	**Q_Router**
Concern	Service Level (Resources, Time)
Source	Router Activity
Requirements	Accommodate as many calls as possible during any busy hour
Focus	Receive and queue calls from customers
Comment	Relevant with concern (contributes to total waiting time)
Decision	Include
Details	****************************
quantity	One
Capacity	Need for modelling balked calls due to space constraint
Dwell Time	Need to model as appropriate distribution for reneging calls
Discipline	FIFO
Break down/repair	N/A
Routing	IN: customer OUT: Q_inf_lines; Q_Ticket_Lines; Q_CSR
Exp factors	Queue capacity (not required)
Outputs	Waiting time (need to monitor as this adds to total waiting time)

Table 4.5 (g): VP analysis in stage 2 for Information Lines queue

Name	**Q_Inf_lines**
Concern	Service Level (Resources, Time)
Source	Inf_Lines Activity
Requirements	Accommodate as many calls as possible during any busy hour
Focus	Receive and queue calls from Q_Router
Comment	Relevant with concern (contributes to total waiting time)
Decision	Include
Details	**
Quantity	One
Capacity	Need for modelling balked calls due to space constraint
Dwell Time	Need to model as appropriate distribution for reneging calls
Discipline	FIFO
Break down/repair	N/A
Routing	IN: Q_Router OUT: Inf_Lines
Exp factors	Queue capacity (not required)
Outputs	Waiting time (need to monitor as this adds to total waiting time)

Table 4.5 (h): VP analysis in stage 2 for Ticket lines Queue

Name	**Q_Ticket_lines**
Concern	Service Level (Resources, Time)
Source	Ticket_Lines Activity
Requirements	Accommodate as many calls as possible during any busy hour
Focus	Receive and queue calls from Q_Router
Comment	Relevant with concern (contributes to total waiting time)
Decision	include
Details	***********************************
quantity	one
Capacity	need for modelling balked calls due to space constraint
Dwell Time	Need to model as appropriate distribution for reneging calls
Discipline	FIFO
Break down/repair	N/A
Routing	IN: Q_Router OUT: Ticket_Lines
Exp factors	queue capacity (not required)
Outputs	waiting time (need to monitor as this adds to total waiting time)

Table 4.5 (i): VP analysis in stage 2 for CSR queue

Name	**Q_CSR**
Concern	Service Level (Resources, Time)
Source	CSR Activity
Requirements	Accommodate as many calls as possible during any busy hour
Focus	receive and queue calls from Q_Router
Comment	Relevant with concern (contributes to total waiting time)
Decision	Include
Details	************************************
quantity	One
Capacity	Need for modelling balked calls due to space constraint
Dwell Time	Need to model as appropriate distribution for reneging calls
Discipline	FIFO
Break down/repair	N/A
Routing	IN: Q_Router OUT: CSR
Exp. factors	Queue capacity (not required)
Outputs	Waiting time (need to monitor as this adds to total waiting time)

Table 4.5 (j): VP analysis in stage 2 for CSR staff

Name	**CSR(Staff)**
Concern	Service Level (Resources, Time)
Source	CSR Activity
Requirements	Serve CSR lines efficiently
Focus	CSR Activity
Comment	Relevant with concern (contribute to overall resource efficiency)
Decision	Include
Details	***
quantity	Three
Activity	CSR
Nature	When activity free
Type	As we have only 3 CSR staff at the moment, So we can model them as floating resource on 4 CSR lines. However, they do not need separate modelling if only three lines are to be modelled as they will be taken as dedicated resource.

Step 6: Integration of View Points

The integration of all the VPs is shown in figure 4.8. As stated before, all VPs can be connected by linking information from the routing details fields. This results in a process flow map that shows all possible VPs in the model along with routing paths for the entities. The process flow map provides a summary of the conceptual model developed through PREmod.

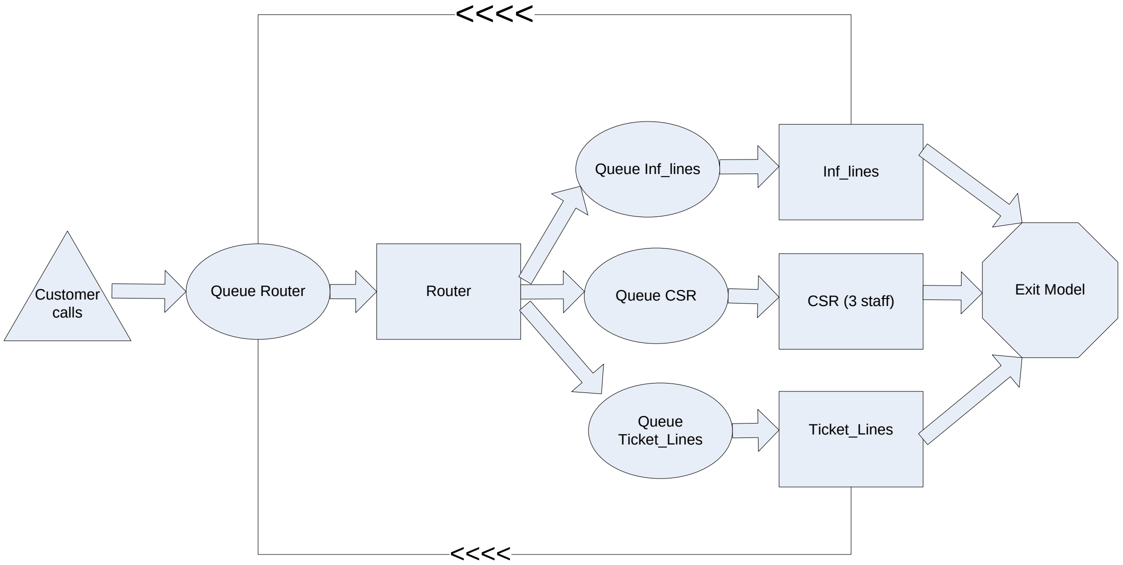


Fig 4.8: process flow map for Wardeon cinema problem

4.3.2 Happyfaces Day Care (HDC) Problem

The second test is about modelling a nursery problem in a residential area which has raised concerns among residents of the street. First, we present the problem description and then application of PREmod is explained next. This problem description has been copied from M&S magazine (2010) and no change has been made.

4.3.2.1 Problem description

"Cedar Avenue is a pleasant residential street in the commuter town of Warentry. Recently the Happyfaces Daycare Center (HDC) applied for permission to build a daycare center on Cedar Avenue to accommodate the needs of this growing community. The residents of Cedar Avenue are concerned about the impact of the center and have challenged HDC's application. There are 63 houses on Cedar Avenue and the adjoining Cedar Court (see Figure 4.9). Residents leaving by car generally encounter difficulty in turning into the adjacent main streets (Elm St and Oak St) because of the relatively heavy traffic flow on these streets. This situation is especially aggregated during the morning and evening rush hour periods because many vehicles use Cedar Avenue as a “pass-thru” between the Elm St and Oak St. Congestion is certainly going to be further aggravated by the traffic resulting from parents arriving to drop-off and collect children from the proposed HDC. The impact on residents will be most severe in the morning rush between 7.30 am and 9.00 am because of the interference with departing residents on their way to work. A simulation study has been commissioned by the municipality and it will focus on this morning period. All of the houses on Cedar Avenue and Cedar Court have off-street parking for one car. There are, furthermore, 26 parking places on these streets but these are reserved for the exclusive use of residents who own more than one vehicle (permits must be acquired). Each workday morning, approximately 60 vehicles belonging to residents leave during the 7:30 am to 9:00 am period. These vehicles exit in equal numbers to Elm St and Oak St.

“A single traffic lane exits onto to both Elm St. and Oak St. and because both of these streets are major thoroughfares, entry onto them is particularly difficult during the rush hours (there are no traffic lights). The pass-thru component of the Cedar Ave traffic is heaviest during the morning rush period and averages about one car every two minutes in the Oak St to Elm St direction and one every three minutes in the Elm St to Oak St direction. It has been observed that in the morning rush period the turning maneuver from the lead position in the exiting lane may require as much as one minute and queues of considerable length often form.

“HDC plans to provide four parking spaces at the rear of the building for use by parents dropping off and picking up children. These will be accessed via a driveway that passes around the building. It is expected that parents will occupy one of these spaces for about 2 minutes and possibly up to 5 minutes in cases where conversations with staff or other parents take place. The planning application indicates that the daycare center will be designed to handle up to 50 children which translates to possibly 40 cars that deliver children in the 7:30 am to 9:00 am period. It is anticipated that the majority of these cars will arrive in the 8:00 am to 8:30 am period. Departing parents are expected to exit in equal numbers to Elm St and Oak St.

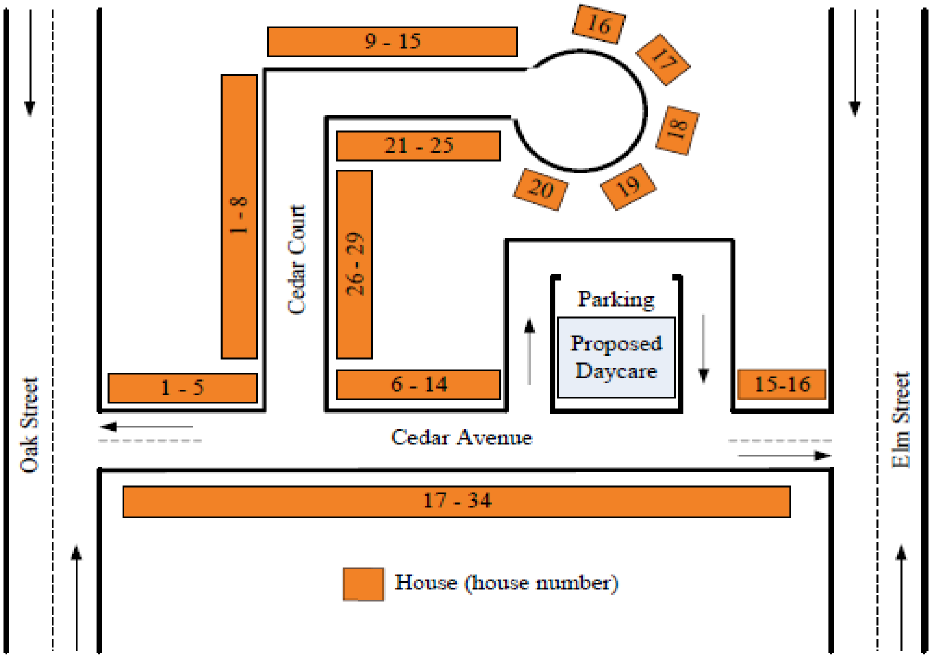


Fig 4.9: Cedar Avenue and proposed Daycare centre

“The primary goal of the study is to obtain insight into the exit queues that develop at each of Elm St and Oak St. during the morning rush period (7:30 am to 9:00 am). With respect to this goal, the following output data has been identified as having particular relevance:

a) The average time to exit to Elm St by parents and as well, the maximum waiting time (measured from the completion of the drop-off task)

b) The average time to exit to Elm St. by residents and as well, the maximum waiting time (measured from the time of exiting their respective homes)

c) The average length and the maximum length of the queue of cars waiting to exit to Elm St

d) The average time to exit to Oak St by parents and as well, the maximum waiting time (measured from the completion of the drop-off task)

e) The average time to exit to Oak St by residents and as well, the maximum waiting time (measured from the time of exiting their respective homes)

f) The average length and the maximum length of the queue of cars waiting to exit to Oak St

“A secondary goal is to determine the time spent waiting for one of the four parking spaces by parents and the average and maximum lengths of the queue of vehicles waiting for one of these parking spaces. Because this queue may spill out onto Cedar Avenue, the municipality is also interested in the effect of having an additional parking space (5 rather than 4).

"Extension An Emergency Services Center is located nearby on Oak St. When there is a call for an emergency service vehicle, the traffic flow on Oak St is significantly impacted. As an extension of the main study, there is an interest in the impact on the primary output data of an emergency vehicle call that occurs at 8:00 am. A satisfactory approach for handing the effect of the emergency call is as follows: (a) the traffic flow onto Oak St is halted for 5 minutes and (b) for the 3 minutes following the 5 minute stationary period the time to turn onto Oak St is doubled for all cars exiting from Cedar Ave."

M&S magazine (2010)

4.3.2.2 Application of PREmod

Step 1: Identification of concerns

Three major concerns have been identified for HDC problem:-

- Time/length of queues (identified through the response required from the exiting queues)
- Parking space (identified through the secondary goal of model i.e. to experiment with an extra parking space at nursery)
- Emergency service centre (to study effect of this centre on the traffic of Oak street)

Step 2: Concern Decomposition

Concern decomposition for HDC is shown in figure 4.10.

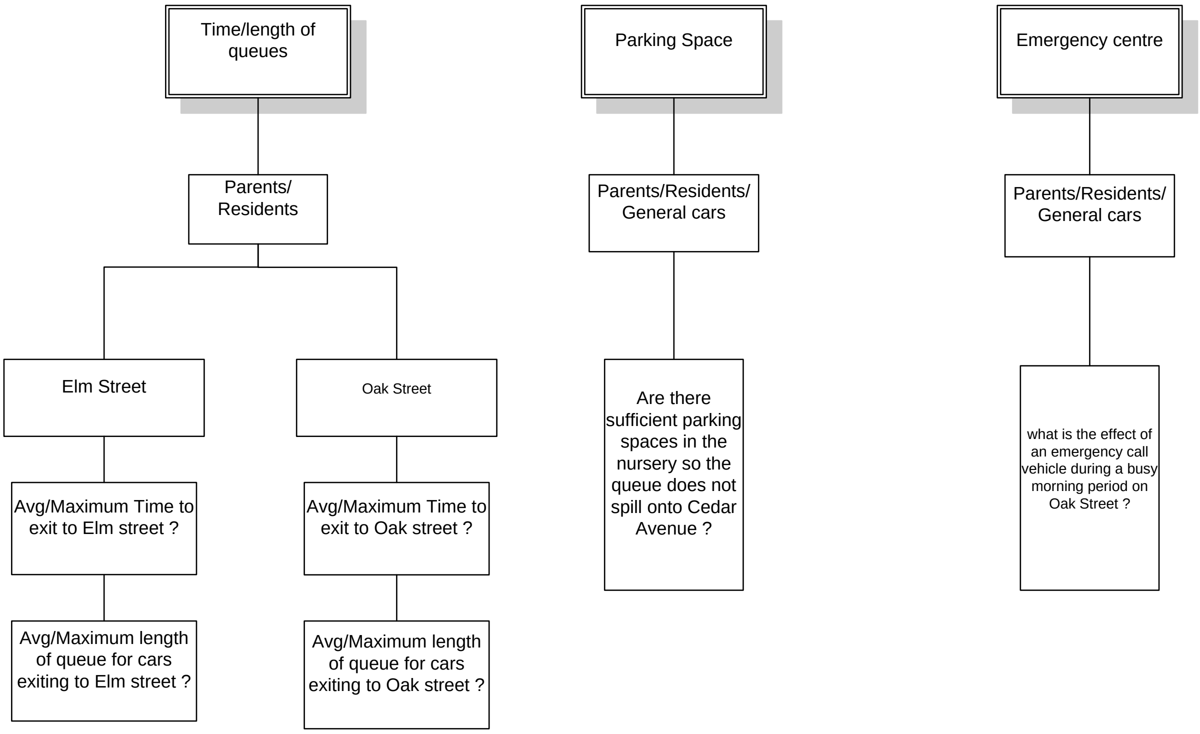

Time/length of queues
Parking Space
Emergency centre
Parents/ Residents
Parents/Residents/ General cars
Parents/Residents/ General cars
Elm Street
Oak Street
Are there sufficient parking spaces in the nursery so the queue does not spill onto Cedar Avenue ?
what is the effect of an emergency call vehicle during a busy morning period on Oak Street ?
Avg/Maximum Time to exit to Elm street ?
Avg/Maximum Time to exit to Oak street ?
Avg/Maximum length of queue for cars exiting to Elm street ?
Avg/Maximum length of queue for cars exiting to Oak street ?

Fig 4.10: Concern Decomposition of HDC problem

As required by the methodology of PREmod, we focus on one decomposed concern at a time. Here we select to work with the concern of *time* derived for Elm Street for the rest of solution, i.e.

What is the average/maximum waiting time for parents and residents to exit to Elm Street?

Step 3: View point identification

VPs in table 4.6 have been identified for the HFDC problem:

Table 4.6: List of the VPs identified for HDC problem

Daycare	Q_Daycare
Drive_Cedar Avenue	Q_Drive_Cedar Avenue
Drive_Cedar court	Q_Drive_Cedar court
Drive_Elm_Street	Q_Emergency Service Centre
Drive_Oak_Street	Q_Parking Spaces_Houses
Emergency Service Centre	Q_Parking Spaces_Road
Gen_Elm_Street	Q_Turn_Elm street
Gen_Oak_Street	Q_Turn_Oak Street
Parents_Elm_Street	Residents_Elm_Street
Parents_Oak_Street	Residents_Oak_Street
Parking Spaces_Houses	Turn_Elm street
Parking Spaces_Road	Turn_Oak Street
parking_daycare	

Step 4: View point analysis stage 1

The VP analysis stage 1 for HDC problem is recorded in table 4.7

Table 4.7: view point analysis stage 1 for HFDC problem

CONCERN	***What is the average/maximum waiting time for parents and residents to exit to Elm street ?***	
VPs	**Decision**	**Justification**
Daycare	include	Driving out from daycare leads to traffic build up on cedar avenue
Drive_Cedar Avenue	include	Traffic on cedar avenue affects queues in exiting lanes
Drive_Cedar court	include	Driving out from cedar court builds traffic on cedar avenue
Drive_Elm_Street	include	Traffic on elm street can affect cars exiting on this street
Drive_Oak_Street	exclude	Traffic on oak street does not affect cars exiting on elm street
Emergency Service Centre	exclude	Does not affect traffic on elm street
Gen_Elm_Street	include	Contributes to the flow of car towards elm street
Gen_Oak_Street	exclude	Does not contribute to the flow of car towards elm street
Parents_Elm_Street	include	Contributes to the flow of car towards elm street
Parents_Oak_Street	exclude	Does not contribute to the flow of car towards elm street
Parking Spaces_Houses	exclude	Does not affect traffic on elm street
Parking Spaces_Road	exclude	Assuming does not effect traffic flow and no resident is coming back in the morning to park on road
parking_daycare	exclude	assuming does not affect waiting time of exiting lanes
Q_Daycare	exclude	assuming does not affect waiting time of exiting lanes
Q_Drive_Cedar Avenue	exclude	assuming no build up of traffic before reaching exiting queues
Q_Drive_Cedar court	exclude	assuming no build up of traffic before reaching exiting queues
Q_Emergency Service Centre	exclude	Related activity is excluded
Q_Parking Spaces_Houses	exclude	Related activity is excluded
Q_Parking Spaces_Road	exclude	Related activity is excluded
Q_Turn_Elm street	include	Response required from this queue
Q_Turn_Oak Street	exclude	Related activity is excluded
Residents_Elm_Street	include	Contributes to the flow of car towards elm street
Residents_Oak_Street	exclude	Does not contribute to the flow of car towards elm street
Turn_Elm street	include	Affects time for cars turning on elm street
Turn_Oak Street	exclude	Does not contribute to the flow of car towards elm street

Step 5: View point analysis stage 2

Next, all templates and the analysis on the VPs for stage 2 are provided in tables 4.8 (a-i).

Table 4.8 (a): VP analysis in stage 2 for Parents on Elm street

Name	**Parents_Elm_Street**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Parents
Requirement	Exit to Elm street as quickly as possible after dropping their children
Focus	Flow of parents cars in the system
Comment	Relevant with concern
Decision	Include
Details	***
quantity arrived	Cars arrived in Elm street during morning period
Arrival Pattern	Arrival distribution
Attribute	Parents Cars
Routing	In: entry point out: Drive_Cedar_Avenue
Exp.Factors	Arrival distribution (Not required)
Outputs	No. of cars engaged in Daycare discussion (Not required)

Table 4.8 (b): VP analysis in stage 2 for Residents on Elm street

Name	**Residents_Elm_Street**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Residents
Requirement	Exit to Elm street as quickly as possible after leaving their home
Focus	Flow of Resident cars in the system
Comment	Relevant with concern
Decision	Include
Details	**
quantity arrived	Cars arrived in Elm street direction during morning period
Arrival Pattern	Arrival distribution
Attribute	Resident Cars
Routing	In: entry point out: Drive_Cedar_Avenue
Exp.Factors	Arrival distribution (Not required)
Outputs	No. of cars left home towards Elm street in the morning (Not required)

Table 4.8 (c): VP analysis in stage 2 for general cars on Elm street

Name	**Gen_Elm_Street**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	General cut through cars
Requirement	Pass through to Elm street as quickly as possible
Focus	Flow of general cut through cars towards Elm street
Comment	Relevant with concern (as it contributes to the flow of cars towards Elm street)
Decision	Include
Details	**
quantity arrived	Cars arrived in Elm street direction during morning period
Arrival Pattern	Arrival distribution
Attribute	Cut-through cars
Routing	In: entry point out: Drive_Cedar_Avenue
Exp.Factors	Varying arrival distribution (Not required)
Outputs	No. of pass through cars towards Elm street in the morning (Not required)

Table 4.8 (d): VP analysis in stage 2 for turning activity on Elm Street

Name	**Turn_Elm street**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Parents/Residents/General Cars
Requirements	Complete the turning maneuver as quickly as possible
Focus	Cars turning onto elm street
Comment	Relevant with concern (as this activity effect waiting times in the queue to exit)
Decision	Include
Details	**
quantity	One
cycle time	Sample time to turn onto elm street as given in the data
Resource	None
Breakdown/Repair	n/a
Routing	In: Drive_Cedar_Avenue out: Drive_Elm Street
Exp factors	Varying Cycle Time (Not required)
Output	No. of cars completed this maneuver (Not required)

Table 4.8 (e): VP analysis in stage 2 for Driving activity on Cedar Court

Name	**Drive_Cedar court**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	cars on cedar court
Requirements	Process the driving of cars as efficiently as possible on cedar court
Focus	All driving on the cedar court
Comment	May have an effect on concern as driving out from cedar court build up traffic on cedar avenue
Decision	Include
Details	********************************
quantity	one (for processing of Resident cars driving out from cedar court)
cycle time	Drive time distribution on cedar court
Resource	none
Breakdown/Repair	n/a
Routing	In: parking_space_house/ parking_ space_road out: Drive cedar avenue (These details suggest to model two VPs as routing in which have been already excluded in stage 1, also this activity only routes out to Drive_cedar_avenue which can directly be modelled taking accounting for the data of this VP also. So following these details , it seems useful to exclude this VP from modelling)
Exp factors	n/a
Output	n/a

Table 4.8 (f): VP analysis in stage 2 for Driving activity on Cedar Avenue

Name	**Drive_Cedar_Avenue**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	cars on cedar avenue
Requirements	Process the driving of cars as efficiently as possible on cedar avenue
Focus	All driving on the cedar avenue
Comment	Has an effect on concern as traffic flow on cedar avenue effects building up of queues in exiting lanes.
Decision	Include
Details	*********************************
quantity	three (each for representing drive for Resident cars, parent cars and general cut-through cars
cycle time	Different Drive time distribution on cedar avenue
Resource	none
Breakdown/Repair	n/a
Routing	In: Parking Spaces_Houses/ Parking Spaces_Road/Drive_Cedar court/Daycare out: Turn_Elm street/ Turn_Oak Street. It can be seen that only Daycare VP and Turn_Elm street VP seems relevant for the concern as others have either been removed or won't affect the concern. Also since we have the data for all the entities flowing into this activity , we can start modelling by taking cedar avenue as Entry point simplifying routing in details
Exp factors	Different drive activities (Not required)
Output	No. of cars that drive in total on cedar avenue (Not required)

Table 4.8 (g): VP analysis in stage 2 for Daycare activity

Name	Daycare
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Parents/nursery children
Requirements	Handle issues with parents as efficiently and timely as possible
Focus	Parents stopping by for dropping their children and to undergo any discussion with staff
Comment	May have an effect on concern as driving out from Daycare builds up traffic on cedar avenue
Decision	Include
Details	****************************
quantity	1
cycle time	Discussion time distribution for parents
Resource	none
Breakdown/Repair	n/a
Routing	In: Parking_daycare OUT: Drive_Cedar_avenue (here again we see that this VP is only routing out to Drive_cedar_Avenue which can directly be modelled as an entry point in the model taking account of the flow of cars coming from this VP source , so it seems useful to exclude this VP in the model also)
Exp factors	n/a
Output	n/a

Table 4.8 (h): VP analysis in stage 2 for queue for Turning Elm Street

Name	**Q_Turn_Elm street**
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Parents/residents/general cut through cars
Requirement	Accommodate as many cars as possible turning onto elm street
Focus	Cars turning onto elm street
Comment	Relevant with concern
Decision	Include
Details	***************************
quantity	1
Capacity	Infinite (to measure the extent of the queuing problem)
Dwell Time	n/a
Discipline	FIFO
Break down/repair	Assuming no breakdown in queues
Routing	In: Drive_Cedar_avenue out:Drive_Elm_Street
Exp Factors	Capacity of queue (Not required)
Outputs	Waiting times of cars exiting to elm street (include as required by concern)

Table 4.8 (i): VP analysis in stage 2 for Driving activity on Elm Street

Name	Drive_Elm_street
Concern	What is the average/Maximum waiting time to exit to Elm street? (Parents/Residents)
Source	Parents/residents/general cut through cars
Requirements	Process the driving of cars as efficiently as possible on cedar court
Focus	Cars on Elm street
Comment	May have an effect on concern
Decision	Include
Details	****************************
quantity	Three (each for representing drive for Resident cars, parent cars and general cut-through cars
cycle time	Driving time distributions
Resource	None
Breakdown/Repair	n/a
Routing	In: Turn_Elm street out: Exit model (the routing details suggest that cars enter the activity after the concern data has been collected from previous activity, so it seems useful to exclude this VP from modelling. Any previous routing details containing this VP can also be simplified to Exit Model in the model.
Exp factors	n/a
Output	n/a

Step 6: Integration of View Points

All the selected VPs are combined into a process flow map as shown in figure 4.11.

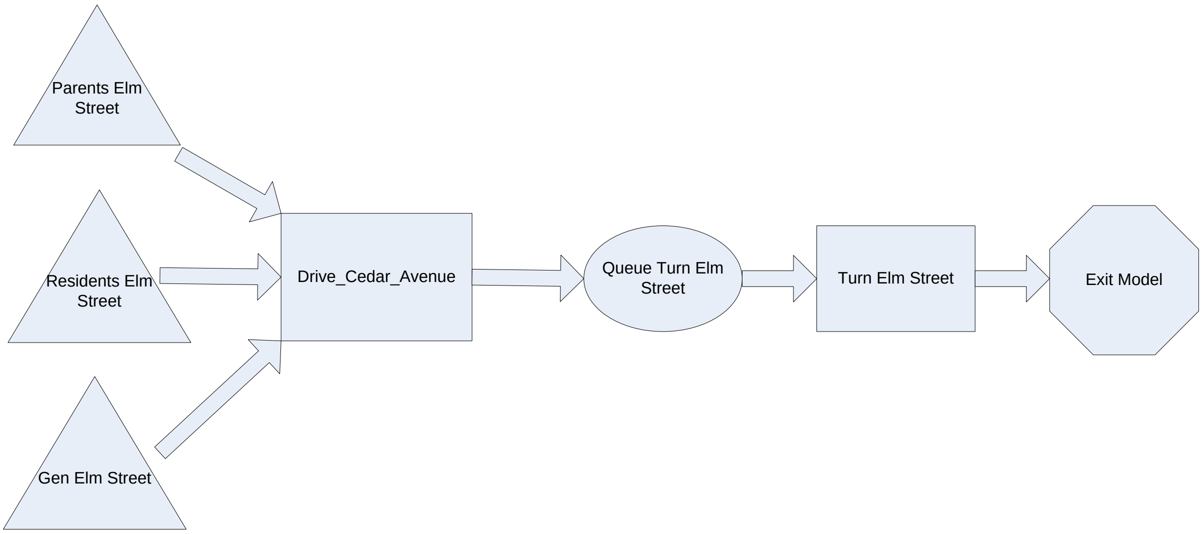


Fig 4.11: Process flow map of HDC problem

4.4 Pilot Testing of PREmod

This section describes three pilot tests that were conducted to evaluate the PREmod framework. In each case, there was one participant at a time but had a different background and experience with simulation. Participants were tutored comprehensively about the method and then provided with the test case for application of PREmod. The total time plan for each pilot experiment was 2 hours, allocating 1 hour to the tutoring process and the remaining hour for the application of the approach on a test case. The call centre problem in section 4.2.1 was used to demonstrate the approach and Happy Faces Day Care test case of section 4.3.2 was used by the participants for evaluating PREmod. In each case, their results and comments were documented. We list these comments and main results for all three pilot tests.

4.4.1 Pilot test 1

Background

Participant "A" was from an operational research and analytics background with a strong interest in e-commerce, Business Analytics, Prescriptive Analytics and big data. The participant did have a background of OR modelling using simulation and statistical methods and had good familiarity with discrete event simulation modelling.

Results

The key results of pilot 1 are summarised as follows:

- The solution of the test cases was produced correctly for most of the steps in the framework. Concern and VPs identification was close to the expected but with few VPs less than the solution in section 4.3.2. However, no important VP was dropped in the test.
- Most VP templates were correctly produced

Participant's comments

These were the main comments from the experiment:

- It was suggested to do the VP analysis stage 2 for all the queues first. This way , it could reduce the chances of making additional VP templates that may not be needed later as they can appear at the time of most queue routing details
- A more clear explanation of the concept of a concern is required. May be some simple examples from OR simulation modelling problems can be used to explain this.
- A clear distinction needs to be made between requirement and focus.
- Some detail fields could be omitted or further explained if required.
- Test cases need further simplification for pilot or experiment purpose within prescribed time-limits.
- The method seems systematic and logical when it comes to developing the system description; however, the abstraction part (removing certain VPs based on the focus) needs to be better structured.

- The method is not suitable for small problems as the decision to include or exclude a certain VP is obvious in most cases.

4.4.2 Pilot test 2

Some improvements were made after the feedback from pilot test 1 and the second pilot test was conducted two weeks later. Major improvements included simplification of some technical details in the VP template. Instead of having requirement and focus both in the template, only focus was retained in this pilot for better understanding of abstraction phase. Test cases were simplified (with respect to the details) so pilot experiment could be conducted more efficiently and timely.

Background

Participant "B" was from OR modelling background and had a very good familiarity and experience of discrete event simulation modelling. This pilot had been working for industrial projects and firms for a few years before starting her research in OR simulation modelling.

Results

The following were the main results from the test:

- Activities in the system, queue, resources and entities were correctly identified for the problem test case.
- Stage 1 and stage 2 analysis was completed successfully with the exception of some extra VPs identified and retained in the final process flow diagram (including VP template)
- Simplified templates were recorded (as opposed to the detailed one in first pilot), allowing the flexibility of the method to be tested.

Participant's comments

These were the main comments from the experiment:

- The process of VP reduction is fairly quick in simple problems; it just requires a comparison of the name of the VP against the concern.
- The definition of each component in the VP template needs more explanation and should be identified at the beginning of every step so users can come up with consistent solutions.
- A clear distinction between different types of resources and its association with different activities should also be made during stage 1.
- Stage 1 of VP analysis may require some more structure about how to identify different VPs; however, it is very useful for a beginner in the area struggling to differentiate between different components of a discrete event model (queue, entity, resource etc.)
- PREmod may require a modeller to have some familiarity with discrete event simulation modelling.
- It seems to be a time consuming process for experts in the field although they may be interested to adopt the first few steps for problem structuring.
- PREmod is a bit of a complicated method for medium sized problem.

4.4.3 Pilot test 3

Pilot "C" was a non-expert in the simulation modelling area and the only idea was to test whether the method is useful (in some way) for users of any other domain. Mainly through this pilot, the communicative part of PREmod needed to be validated, that up to what extent the method is communicative to the non-experts clients.

Background

Participant "C" was from a marketing background and has been doing research in the field of marketing strategy during his PhD thesis. This pilot was not familiar with the area of OR simulation modelling and so the method was tutored with some simplifications to get some general feedback.

Results

Some key results were as follows:

- A different list of VPs was generated during the problem solving phase (mainly due to the participant’s different background) and reflected more perspectives in the problem. For example, congestion, pass-through, traffic light etc. were added in addition to cars, parents and roads.
- Not all steps were followed for VP analysis in stage 2 as templates were difficult to follow; however, interesting measures to solve the problem was put forward from the participant’s own perspective as given in the comments below.

Participant’s comments

The following were the main comments from the third pilot experiment:

- The method in general possesses a way of analysing different behaviour (in any domain). For example, in this case, when analysing consumer behaviour for instance, modeller would understand the situation from the consumer VP (which is usually low price and high quality) along with the seller VP (which is mostly high revenue and acceptable quality). An additional requirement from seller VP can be loyalty to customer, quality, time etc.
- The notion of view point is an interesting idea to structure the problem and also the VP template makes it easy to describe and analyse the problem in detail.
- It may be very good for a very complex model (next stage of research), but it looks too hard for a simple problem, as abstraction is very straight forward in most cases.
- Comparison of focus with the concern for elimination purposes sometimes gets confusing, as there is a lack of structural support. Unless, every requirement or detail has been investigated, the user can make the model over-simplified by removing some important VPs.

4.4.4 Findings

The key findings from the PREmod pilot tests can be summarised as follows. PREmod is more suited for complex problems because a large problem has more potential for breaking down into VPs lists and also allow more room for abstraction during VP analysis. On the contrary, for simpler problems, it seems an unnecessary effort to do the comparison of focus with the concern as this process does not include any visual checks and merely the “words” are compared in most cases. Knowledge of discrete event simulation is necessary to use PREmod although the general idea of VP orientation still seems useful in other domains.

PREmod clearly lacks some structure in stage 2 of VP analysis where decisions are being made on the basis of information in the template only. This can increase the variability in choices due to the subjective and flexible nature of the process.

4.5 Conclusion

In this chapter, PREmod has been proposed for CM within M&S projects. It has been shown that being adapted from one of the software requirements engineering methods (PREview), PREmod has potential to gain similar benefits as that of PREview in the initial phase of model development. Similarities between the two methods have been identified and any modification within PREmod from the PREview method has also been clearly explained. It has also been shown that different sets of requirements and concerns can be structured from different viewpoints using PREmod for simulation modelling problems. Some information gained during the application of PREmod was not as effective as in PREview but it was still found to be useful in structuring different aspects at both high and low levels of modelling.

There are however certain limitations which were identified clearly during the pilot testing of PREmod. The main problem identified was during the abstraction (VP stage analysis stage 2) of the method. There seems to be a lack of structural approach for removal of VPs and abstraction of details within them. As there is no visual link between the activities and components of the whole problem, the abstraction part turned out to be a more subjective process where the users need to refer back to concerns for every decision. Also, though the notion of VP looks very appealing, extra care is required by beginners who may unintentionally miss modelling components during VP identification. This is because there is a great deal of variability under VP categories. A lack of a diagrammatic layout during the structuring phase of VPs was felt strongly during pilot tests as this would have provided a clear connection between the components which would ease the abstraction phase. This would also increase the structural support during the abstraction phase.

Due to the reasons mentioned above, PREmod is not pursued further and an alternative approach based on the Structured Analysis and Design Technique (SADT) from software requirements engineering is explored. It was anticipated that the method would overcome the limitations of PREmod due to its visual and structural support for problem structuring in the initial phases of software development. In the next chapter we describe the framework for CM based on SADT.

5 SADT CM (framework for Conceptual Modelling)

5.1 Introduction

This chapter describes the framework for CM adapted from the software requirements engineering method called SADT (Structured Analysis and Design Technique) (Ross, 1977). The methodology is called SADT CM. It is expected that by using SADT CM, a more structured approach for CM is learned by the modellers. Since at present, most CM approaches are subjective in nature, SADT CM aims to provide a better structuring of the problem and provides a mechanism to perform the abstraction process in CM which is largely absent in other approaches.

This chapter is structured as follows. Section 2 presents the rationale of using SADT in CM, section 3 provides some related work in the area, section 4 introduces original SADT while section 5 discusses the adaption of SADT into SADT CM. Section 7 illustrates the complete SADT CM framework with a detailed step-by step explanation. This is illustrated with the help of a bank problem provided in section 6. Section 8 discusses the application of SADT CM on the Panorama TV case. The chapter concludes with a brief summary of SADT CM framework. A comprehensive user-guide is also developed as a part of this research which is included in appendix 1. This is only a "how to" guide based on this chapter, so it does not include theoretical details and a justification of the approach as it is intended for the quick application of the approach by modellers.

5.2 Structured analysis and design technique (SADT)

SADT has a long history within software engineering. It was developed by Ross (1977) as a result of on-going work (1969-1973) in problem solving dating back in 1950s at Softech. SADT is a graphical language and was used extensively for describing complex systems in communicative designs, military planning and computer-aided manufacturing (Dickover et al., 1977). SADT was successfully applied in problem analysis and functional specifications; however, it has been used most effectively in the requirements definition phase for software design (Ross and Schoman, 1977; Ross, 1985).

SADT was adopted as Icam DEFinition for Function Modelling (IDEF0) by the US Air Force Integrated Computer Aided Manufacturing (ICAM) in 1980s. It was part of ICAM's program to fulfil the need of powerful modelling methods for system analysis and design. Several other IDEF packages were introduced during that period; mainly, IDEF1, IDEF2, IDEF3, and IDEF1x. These are not discussed in this chapter, but details can be found in Menzel and Mayer (1998). Since our focus is mainly on the root definitions and labels developed in SADT, we prefer to use SADT instead of IDEF0. IDEF0 is mentioned, however, when any related work is addressed.

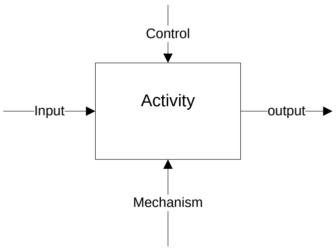


Figure 5.1: General notation of SADT.

SADT notations consist of box-arrow diagrams (blocks), with four arrows on each side defined as: input, output, control and mechanism and one activity in the middle as shown in figure 5.1. Their definitions consist of the following:

Activity: an activity is any function or process that serves to transform inputs into outputs
Input: the data/information required by an activity to start the transformation process
Output: the data/information produced by the activity as a result of this transformation
Control: any constraint that affects the behaviour of activity in some way
Mechanism: persons, resources, or any means that are required to run the activity

SADT uses the traditional "top-down" approach for the hierarchical modelling of functions. The process starts with specifying the highest simplified level of detail in the top-level diagram which is then decomposed into further details at each step until the required level of detail has been reached. This approach is shown in figure 5.2. In the first stage, a top-level diagram A is created (stage 1 of figure 5.2). This diagram simply states that, the input I_0 is transformed to output O_0 by means of an activity A_0 being constrained by control C_0, and using some resource R_0. Usually C_0 and R_0 are group notations which indicate that this particular type of constraint or resource would be needed to complete this transformation. In the next stage, A_0 is decomposed into further sub-activities, showing a more detailed lower-level view of the process (stage 2 of figure 5.2). Each stage now shows individual inputs, outputs, controls and resources; however, they are working discretely to produce the same transformation (I_0 into O_0) as in the original high-level diagram.

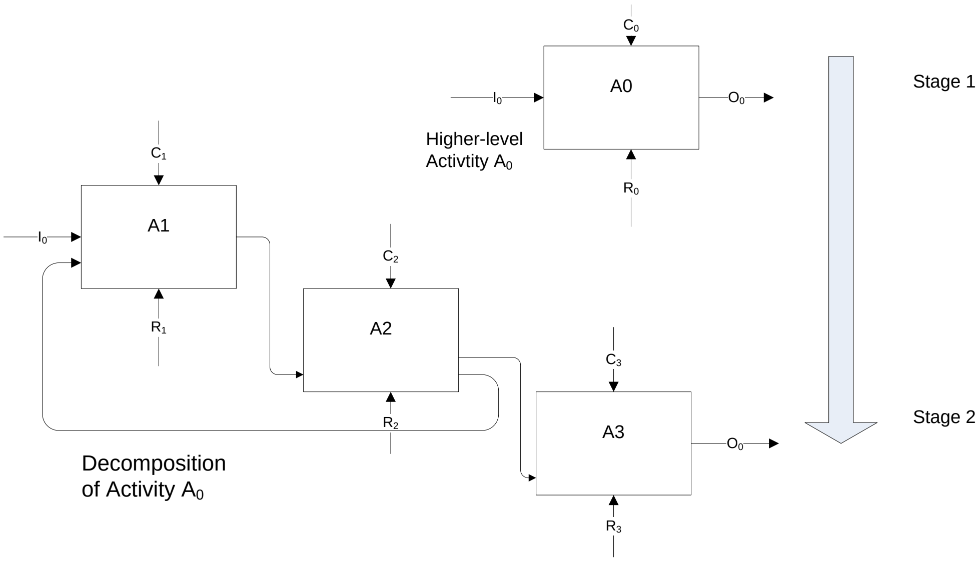


Figure 5.2: Hierarchical decomposition of SADT in “Top-down” style.

5.3 Related Work

There is not much evidence on the use of SADT within M&S, and more specifically in the CM phase. Some studies, however, indicate that SADT is a potential candidate for use in modelling service operations and queuing systems. For instance, Santarek and Buseif (1998) propose the use of SADT in creating high level system design specification for flexible manufacturing systems. Van Rensburg and Zwemstra (1995) demonstrate the process of developing simulation models of air traffic controller positions using IDEF0 and IDEF3 hierarchal modelling. Congram and Epelman (1995) explains a procedure to describe a service operation (individual tax-return) using SADT diagrams at each stage. Kim et al. (2003) discusses the integration of IDEF models with UML models in order to bring richer semantics in enterprise modelling. Whitman et al. (1997) describes a procedure for converting static models (built in

IDEF0/IDEF3) to simulation models using some rules and model annotation. Jeong (2000) uses IDEF0 and IDEF3 for simulating optimization of scheduling systems. Some efforts have been made towards the use of IDEF0 in creating a standard documentation for modelling methodologies (Nathan and Wood 1991). However, all these studies do not focus explicitly on CM.

Some recent work (Montevechi et al., 2008; Montevechi et al., 2010) has proposed the use of IDEF0 combined with other techniques (SIPOC/flowchart/IDEF3) for CM. Montevechi et al. assert that IDEF0 alone can not capture enough details for DES CM study. Meanwhile, in the Eurpoean research project "PROTOCURE II" (Lucas et al., 2005) an extensive review of 70 modelling langauges is presented for the purpose of selecting a languge for the medical guideline development process. The selection is carried out on the basis of requirements and features desirable for a process modelling langauge, and takes into account whether the selected langauge is *living* in terms of references, documentation and known history of origin. Lucas et al. selected IDEF0 as the main process modelling language in preference to some other well-known modeling languages, for instance, E3, GRAI and UML2.

5.4 Rationale for using SADT in CM

CM is mainly divided into two steps:

1 - Generation of a System Description

2 - Abstraction of the Conceptual Model

These two steps are explained by Robinson (2011) and shown in figure 5.3.

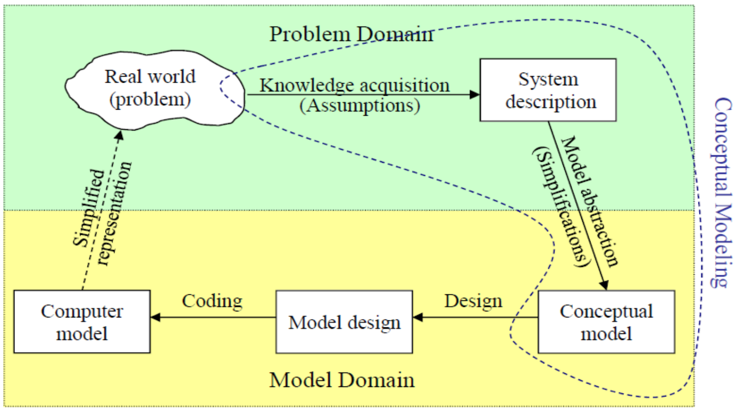


Figure 5.3: Artefacts of Conceptual Modelling (Robinson 2011)

While the System Description (SD) helps in knowledge acquisition from the real world problem domain, the abstraction phase helps to simplify this SD into a conceptual model by removing any irrelevant details required for modelling. SADT CM helps to achieve these two steps in a structured framework. In the first phase it collects information about a problem situation and the main components involved with respect to that problem (people, products, processes, constraints etc.). In the second phase, it provides a means to analyse this information by removing elements of the SD through abstraction. Both steps and their relevance to SADT CM are explained below.

SD refers to the structuring of the problem within the problem domain. This involves acquiring knowledge from the problem domain, connecting it with the other available information, making any necessary assumptions, and finally producing a document that describes what the problem is about. The SD can contain as much information as can be acquired in the initial stage of problem structuring and therefore is separated from the abstraction stage (which mainly focuses on keeping the relevant details for modelling). Often these two stages are mixed up and modellers jump into the design of the computer models directly from the SD (Robinson, 2012). In other words, the SD lies within problem domain and real world while the conceptual model (from abstraction stage) lies within the modelling domain and explains what part of the SD will be included in the final computer model.

Although there are number of works that provide guidelines for generation of SD (van der Zee and van der Vorst, 2007; Balci and Ormsby, 2007; Robinson, 2008; Montevechi and Friend, 2012; Kotiadis, et al., 2013), most of this work is either domain-specific (healthcare, military, engineering) or descriptive in nature, and so lack a precise structure for creating conceptual models. Also, in all of these studies, there are discussions for activities, actions, objectives, communications, contents, inputs and outputs, but there is little attention on explicitly structuring different components of the SD (entities, resources, controls and activities) in a DES study. Such an approach requires a thorough investigation of these key individual components and interactions among them from the beginning of the CM process. If these details are put together in a structured and communicative form, then it can minimize the efforts required in the abstraction and design phase. This purpose is achieved by using the SADT CM framework. Its generic style and notations makes it suitable for explicitly structuring various components of the conceptual model in a well-organized form. This helps to better abstract a conceptual model in the next stage. SADT CM is shown to efficiently capture the SD along with all the necessary information allowing easy development of the conceptual model in the abstraction phase.

The abstraction phase of CM involves removing any unnecessary detail from the SD. SADT CM facilitates this step by explicitly providing a step-by-step procedure to achieve the desired conceptual model. SADT CM involves critically identifying the modelling objectives at the beginning of the abstraction phase. All abstraction is then carried out with respect to the modelling objectives. This is essential in getting the right conceptual model. Another idea that is related to the modelling objectives is the “model outputs (responses)”. In this context, the model outputs are the response that we want to measure from the model after the model simulation has been performed. These responses are collected by changing some model inputs (experimental factors) to see the effects on the model outputs. SADT CM gives room to accommodate all the necessary inputs as required in the abstraction stage by primarily focusing on the responses derived from modelling objectives.

5.5 SADT CM (adaption from SADT)

Since we believe there is a tendency to over-elaborate the SD, we propose an approach to CM using SADT with little modification. The proposed method creates SADT CM branches (activity based models) within the style and notation explained later on in this section. Each individual activity is screened and details of various components (entities, resources and controls) are added. The reason for selecting “Activities” as the starting point of the process is that they are central to DES. All other components are dependent on the individual activities in the system. Any input/output activities are also identified for the “Activity”. This allows for modelling any one “Activity” in a system comprehensively along with all dynamics at a lower level. Queues are not identified at this stage since it is normal for activities to have a queue feeding them in a DES unless the activity is an entry or exit point in the system. In some cases queues can be identified as single or multiple depending upon how the activity is fed by other activities in the system. This being the reason, provision to identify the queues is added as a last step in the abstraction stage where modeller has chance to comment on inclusion of a queue and its nature. It would not absolutely necessary to define the nature of the queue, but the inclusion or exclusion decision is necessary and should suffice at the CM stage.

Due to the nature of DES, the original SADT notations and style have been modified to suit a DES modelling context. These are shown in figure 5.4. The box represents any one activity and the input/output arrows represent the preceding/following activity. The resources and control arrows represent the same components as in SADT. The output activity arrow includes an entity label indicating the flow of the entity on this route. This is an additional label to the original SADT activity block.

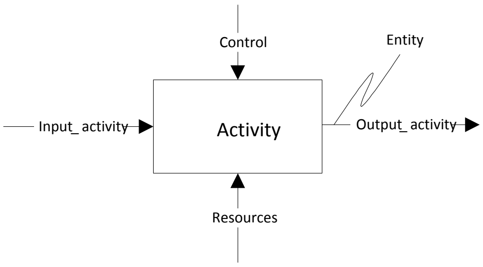


Figure 5.4: Proposed notation of SADT CM.

The definitions of the labels used are as follows (figure 5.4):

- ***Activity***:

This corresponds to the activity under consideration (transformation activity). This is the central part of the analysis. Therefore, it is important to identify all activities carefully in the system. There is only *one* transformation activity in a single SADT CM block. The label uses a notation of alpha-numeric style indicating the operation or activity (e.g. OP1, and A1 to indicate operation 1 and activity 1 in the system).

- ***Input activity***:

This determines all activities preceding the activity. Arrows can be replicated in case of more than one input activity.

- ***Output activity*:**

This determines all activities following the activity. Arrows can be replicated in case of more than one output activity.

- ***Resources*:**

Any resource (or group of resources) required by the activity is indicated by the bottom arrow of the box. The notation used is "R_x" where "x" represents the ID of a resource (or group of resources). In case of more than one ungrouped resource, "x" can indicate this value as a range for e.g. R_{1-3} means three individual resources (R_1, R_2, and R_3) are required on a particular activity. The values of "x" are unique unless used for shared resources (or a group of resources) on different activities. Arrows can be replicated for two completely different types of resources on a single activity, for example, the operators and transportation resources can be both represented by R_x and T_x on two different arrows respectively.

- ***Controls*:**

This represents any constraint imposed on the activity by the environment, management, production plans, customer demand etc. A simple notation "C_X" is used, where "x" denotes the ID number of the constraint. This label will also be the same for activities sharing any similar constraints. Arrows can be replicated for multiple constraints on the same activity.

- ***Entities*:**

This label represents the entities flowing out from the activity and following this route. These are represented by E_x where "x" represents the ID of the entity. In the case of more than one entity type on a single branch, entities are denoted by E_xE_y where "X" and "Y" represents the IDs of two different entities. Other acceptable notations for different type of entities on one branch are Ex\Ey and Ex+Ey; where "\" or "+" simply denotes "AND", meaning first and second entity on any single branch.

The labels are used in alpha-numeric style to indicate different components of SADT CM block. For example, E_1, E_2 for entities, OP_1 and OP_2, for operations, and R_1 and R_2 for resources are used. Where more than one resource or control is required for a single activity, the same style of notation is used as for multiple entities. For shared resources, a range can also be specified, for example, R_{1-5} (to indicate 5 resources required for an activity). It is also acceptable to use regular fonts in alpha-numeric fonts without using subscript, e.g. R1 can be used instead of R_1 and E1E2 in place of E_1E_2. This allows easy labelling of the SADT CM branches and could be preferred for quick application of the framework.

The next modification proposed is changing the modelling style from the original "top-down" style of SADT to a "bottom-up" approach for SADT CM. This is because we believe that the DES domain is different from that of software systems in this respect. The details in the DES model are present at the lowest individual decomposition level rather than the highest level of information as in enterprise wide modelling (Whitman et al., 1997). These atomic models can then be connected in a "bottom-up" style to represent the complete system. This would also maintain the consistency in representing the dynamics of the components at all levels of modelling. This approach has also been used in component based or hierarchal modelling in DES (Sargent et al., 1993; Nance, 1979; Cetinkaya et al., 2009). This style of modelling is the basis for creating local *agents* in Agent Based Modelling (ABM). *The agents' behaviour is* specified at an individual scale and then the global behaviour of the system emerges at a higher level as a result of these local interactions (Batty et al., 2003; Borshchev and Filippov, 2004). Another benefit of this approach is that the individual modelling units can be studied at various locations (distributed modelling) which can then be combined at a central server to get the complete system information. This individual study can be carried out at lower levels even when knowledge about the complete system is unknown. Some interesting work in this area is from Pratt, Mize, and Kamath (1993).

The approach is shown in figure 5.5. First, the low-level diagrams are created, where all individual activities are investigated individually after listing all entities, activities and resources (explained in more detail in the next section).

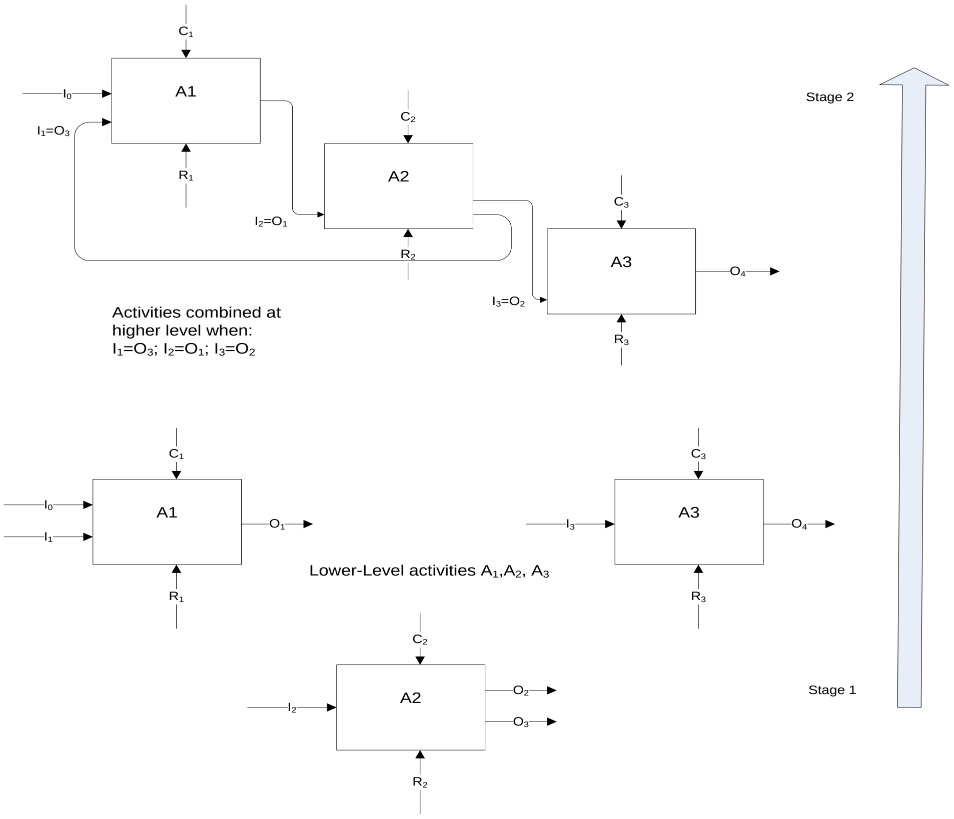


Figure 5.5: Proposed "Bottom-up" approach for SADT CM

During this investigation, all SADT CM *branches* are created with the proposed notation (figure 5.4) and following the label conventions explained earlier in this section. Once these individual lower-level diagrams (OP1, OP2, OP3) have been created, they are combined together to generate a higher level (but detailed) description of the entire system. This combination follows from the inter-connection of individual input/output activities to represent entity paths and feedback loops. For example, OP1 is fed by two activity sources: a source or an external activity and an output activity from OP2. It should be noted that in figure 5.5, input/output labels are for illustration purposes only in order to give an analogy to SADT labels (figure 5.2). In the proposed framework, these labels represent input/output activities and are labelled similarly to the main activity. The complete framework for SADT CM is explained in subsequent sections with reference to a simple bank problem whose description is explained next.

5.6 Bank problem

Customers enter the bank where they need to go to the required service counter. They first meet a Customer Representative (CR) who directs them to either the cash counter or bill payment counter. Both counters are served by dedicated bank tellers. The bank is facing a queuing problem at the cashier counter. The manager of the bank is interested to know about the number of customers served by the cashier during any busy hour of the day (their waiting times also) and whether it is beneficial to invest in another cashier for this service. The manager can take the decision to hire a new cashier if the average waiting time of the customers is more than five minutes.

5.7 SADT CM Framework

Using the adapted SADT approach explained in section 5.5, and with reference to the bank problem explained in section 5.6, we now discuss all the steps and procedures involved in SADT CM.

The SADT CM framework consists of the following steps:

Part A: Generation of System Description (SD)

- Step SD1: Identify all the entities in the problem
- Step SD2: Identify all activities in the problem
- Step SD3: List any control associated with each activity
- Step SD4: List any resources associated with each activity
- Step SD5: Draw the SADT CM branches for each activity
- Step SD6: Combine all SADT branches to generate a complete SD
- Step SD7: Verification and validation of SD

Part B: Abstraction of Conceptual Model from SD

- Step CM1: Identify the modelling objective
- Step CM2: Identify the experimental factor
- Step CM3: Identify the Objective Activity(OA)

- Step CM4: Identify the Connected Activity (CA)
- Step CM5: Conduct CA impact test
- Step CM6: Check if all CAs have been analyzed
- Step CM7: Analysis of Controls, Resources and Entities
- Step CM8: Analysis of the queues
- Step CM9: Make abstracted diagram of the conceptual model
- Step CM10: Verification and validation of conceptual model

5.7.1 Generation of System Description (SD)

The generation of the SD is achieved by following the set of steps as listed in Part A. The process starts with identifying the major components of a conceptual model (activities, resources, entities, controls) followed by a construction of the CM SD in diagrammatic format using the SADT CM notations. The whole process of generating a SD can also be understood with the help of the flowchart in figure 5.6. The flowchart explains that after step 4, a thorough re-check is required to ensure that all activities, resources, controls and entities have been identified correctly. A final verification of SD is also performed at the end of part A. The two tables used to record information in all these steps are shown in figure 5.7. They are called entity and activity tables. These tables will be filled-in as we go through each step of part A. Entity Table(s) will be used to record information in step 1 while Activity Table(s) will be used to record information from steps 2 to 5. The first instance of these tables are blank and then subsequently numbered as the information is added at each step.

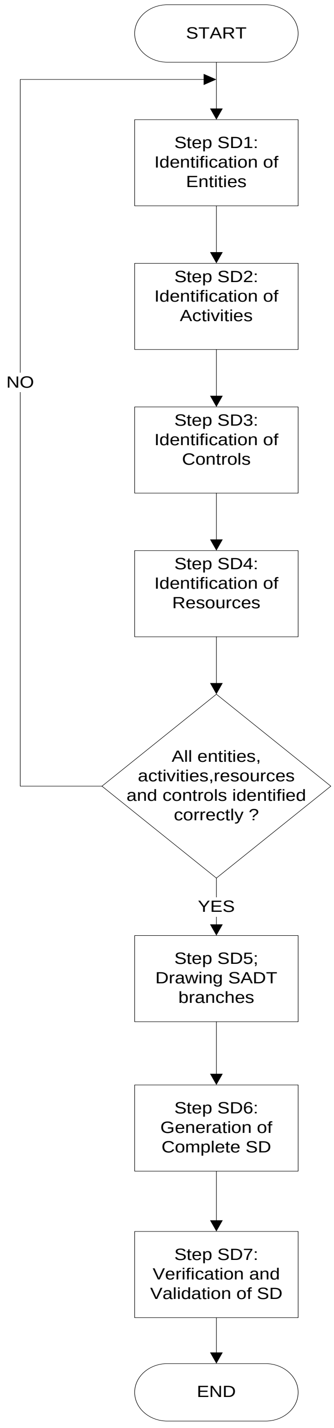

START
Step SD1: Identification of Entities
Step SD2: Identification of Activities
Step SD3: Identification of Controls
Step SD4: Identification of Resources
All entities, activities,resources and controls identified correctly ?
NO
YES
Step SD5; Drawing SADT branches
Step SD6: Generation of Complete SD
Step SD7: Verification and Validation of SD
END

Figure 5.6: Process of generating SD in part A of SADT CM

Entity Table:

Entity	Label

Activity Table:

Activity	Control	Resource	Output Entity

Figure 5.7: Tables for recording information through Steps 1 to 6 in part A

5.7.1.1 Step SD1: Identification of entities

The first step is to identify all entities in the system. These include any type of raw material, product, moving objects, people, cars, unfinished products, pallets, fixtures, telephone calls, etc. All entities are listed in a table allocating a unique label name to each entity (Ei). The list of the entities can be generated by looking into the problem description and locating any possible entities in the system, through discussion with managers or the project leader and checking any other information available (layout diagram, customer notes, operations details etc.). It is not necessary to identify any combined or split entities explicitly at this stage as this will be determined in the later stages when activities and output entities are determined. The Entity Table (figure 5.7) is used for recording the information at this step.

Bank Problem: Entities

The entities identified for the bank problem is recorded in Entity Table (1).

Entity Table (1)

Entity	Label
Customers (Cashier)	E1
Customers (Billing)	E2

5.7.1.2 Step SD2: Identification of Activities

In this step, all activities are defined and labels are assigned to them. The activities are carefully identified by brainstorming, aiming to identify all possible activities in the problem domain and concerning any other related data given about the problem (for example, lay out diagram, tables or flow charts etc.). Some of the possible general activities can include (but not limited to) the following:

- Arrival (can be initiated by entities like customers, telephone calls, data signals, raw materials etc.)

- Services (for example, machine operations, calls servicing, customers’ service, drivers parking or driving etc.)
- Exit (executed usually when any finished product or service leaves the system)

Any Activities identified are placed in the “Activity” column of the Activity Table (figure 5.6). Only one activity must be placed at each field of the column and an appropriate label should be assigned to this within parenthesis. For the entities that form the starting point of the flow of entities in the system, a prefix of “Sc” is attached with their label. This is to indicate that this activity acts as “Source” in the system. The same is repeated for the activities which exits the entities out from the system with the pre-fix of “Sk” added to their label to indicate the “Sink” in the system. The “Activity” column in the Activity table is used to record the information at this step.

Bank Problem: Activities

The activities identified for the Bank problem are listed in Activity Table (1) along with the label information as shown below.

Activity Table (1): Activities in the Bank Problem (Step SD2)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)			
Meeting CR (A2)			
Cashier Service(A3)			
Billing Service(A4)			
Customers Leaving Sk(A5)			

5.7.1.3 Step SD3: Identification of control related to activities

In this step, any control is identified related to the activities identified in step SD2. Controls can be seen as "constraints" on any single activity that limits the process at this point and hence can affect the output in the complete operations chain. Constraints can take many forms as explained below:

- Physical (tangible or non-tangible: examples include equipment maintenance, set-up times, repair times, material shortages, lack of people, lack of space, product types, arrival patterns, technology, layout, buffer specification etc.)
- Policy (company procedures, union contracts, government regulations: examples include overtime policies, mandatory breaks, maximum working hours, new employee training etc.)
- Paradigm (some beliefs or habits: for example, the belief that "we must always keep our equipment running to lower the manufacturing cost per piece")
- External (environmental, safety or geological concerns: for example, a particular machine may be sensitive to changing weather and hence require a specific time for running)

A modeler is advised to consider all the above points to determine any possible constraints in the problem. A careful discussion around each activity is required. Some constraints might not be visible at first sight, but can come to light through detailed discussions in the initial meetings of the simulation study. Constraints are identified as "Controls" in SADT CM in a way that it determines whether an activity can proceed or not under certain conditions. In some cases, it controls the pace of an activity in terms of a restriction for a condition to be met before processing. Also controls are not to be confused with "Resources" which are explained in the next step. The information obtained in this step is recorded in the "Control" column of Activity Table along with the assignment of a label (C_x).

Bank Problem: Controls

The only control that can be identified through the problem description in the case of the bank problem is the customer arrival pattern (based on their choice of service), since this will largely affect the queuing of people at both the cashier and billing counters. This is recorded along with the label in Activity Table (2) as shown below:

Activity Table (2): Control in the Bank Problem (Step SD3)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)		
Meeting CR (A2)	N/A		
Cashier Service(A3)	N/A		
Billing Service(A4)	N/A		
Customers Leaving Sk(A5)	N/A		

5.7.1.4 Step SD4: Identification of resources related to activities

In this step, any resources are identified that relate to the activities. The resources fall under the following categories:

- Humans (examples: workers, machine operators, maintenance operators, shift operators, security staff, line managers, customer representatives, service operators, hygiene persons, etc.)
- Equipment (examples: tools, forklifts, computers, barometer, thermometer, pressure gauge, heat sensors, fire detectors etc.)
- Information (examples: databases, schedule sheet, time sheets, capacity and material plan etc.)
- Natural (examples: air, water, land, minerals, etc.)

It is advised to identify the resource in a problem by considering all potential resources described above and relating them to every activity. This can be done by reading the problem description, meeting with the managers, observing the factory personnel, taking data about the organizations resources from the operations managers and discussing any external or natural resources to be used within the processes. It is also important to distinguish between the nature of the resource (shared or dedicated). A shared resource is shared among different activities while a dedicated resource is only required by a single activity and is available at all times for that activity. The resources are documented in the Activity Table in the "Resource" column along with a unique label. If a resource appears more than once in a table (shared resource) then the same label will be used on each occasion. A dedicated resource is represented by a unique label. Before moving on to the next step, it is advised to verify all identified components in the Activity Table obtained in this Step (SD4) as this can affect the drawing of the SADT CM branches and any missing component would be hard to locate in the later stages.

Bank Problem: Resources

In the case of the Bank Problem, the resources are identified are the CR, cashier and billing accountant. These are documented in Activity Table (3) as follows:

Activity Table (3): Resources in Bank Problem (Step SD4)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	
Meeting CR (A2)	N/A	CR (R1)	
Cashier Service(A3)	N/A	Cashier (R2)	
Billing Service(A4)	N/A	Bill Accountant (R3)	
Customers Leaving Sk(A5)	N/A	N/A	

To verify that Activity Table (3) contains all components of the bank problem it can be carefully checked against the problem description to ensure that no activity, control or resource is missing at this stage.

5.7.1.5 Step SD5: Drawing SADT branches

In this step, all the information we attained from steps SD1 to SD4 is used to create SADT CM branches. Before drawing the SADT CM branches, the last column of the Activity Table is filled-in. The “output entity” column is filled-in by considering any possible entity on its output path along with its label. This can be thought of as an output from an activity. These entities are labeled as explained in section 5.5. These entities will be placed on the “Entity” label of the SADT CM block. These entities must have been identified in step SD1 and no new entity should be identified at this step. The only difference is to jointly or separately list entities from the Entity Table as per the output path of the activity under consideration. Once this is done, SADT branches can be created using the symbols and notations of the SADT block as explained in section 5.5. This will contain input activity, output activity, output entity, control, and resource with respect to the main activity.

Bank Problem: Drawing SADT CM branches

For the Bank Problem, the output entity column is filled up as shown in Activity Table (5) below and corresponding SADT CM branches are shown in figure 5.8.

Activity Table (5): Output Entities for the Bank Problem

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2
Meeting CR (A2)	N/A	CR (R1)	E1 E2
Cashier Service(A3)	N/A	Cashier (R2)	E1
Billing Service(A4)	N/A	Bill Accountant (R3)	E2
Customers Leaving Sk(A5)	N/A	N/A	N/A

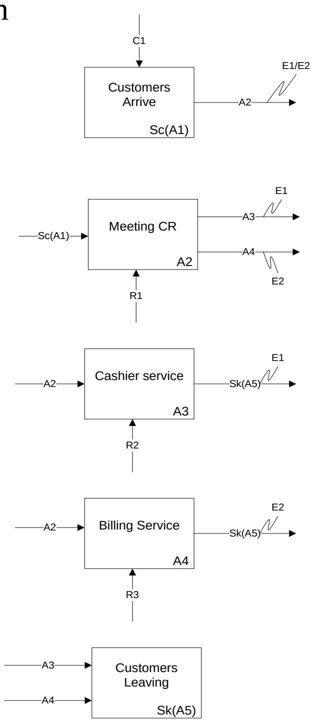


Figure 5.8: SADT CM branches for the bank problem (STEP SD5)

5.7.1.6 *Step SD6: Generation of complete SD*

After drawing the branches in Step SD5 individually, they are combined in this step to produce the compete SD. The following steps ensure the correct generation of the complete SD from the individual SADT CM branches.

- Start with a “source” (preceding) activity and follow the output activity.
- Locate the branch with the main activity similar to the “output activity” of the preceding activity. Consider this as the following activity.
- Connect the output activity arrow of the source (preceding) activity to the input activity position on the following activity. Remove any input activity arrow on the following activity.
- Remove any activity label from this new connected arrow and label this arrow with the output entity of the preceding activity.
- Consider this following activity as a new preceding activity.
- Repeat all steps as prescribed above until only the “sink” activity is left.
- Connect the final sink activity in the same manner but without any output activity or entity labels.
- In case of multiple sources or multiple sinks in a system, repeat all the above steps until no source or sink is left unconnected.

Bank Problem: Compete SD from SADT CM branches

Figure 5.9 presents the complete SD for the bank problem obtained as a result of observing the steps explained in step SD6.

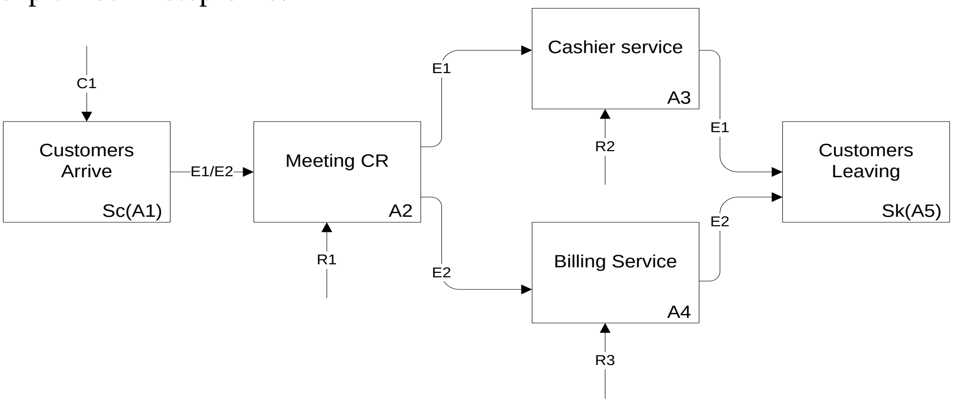


Figure 5.9: complete SD for the bank problem (STEP 6)

5.7.1.7 Step SD7: Verification and Validation of SD

A verification of the SD can be performed after completing all the steps of generating a SD. This primarily requires a thorough check in the last two steps where the branches of the SADT CM have been created. According to the flowchart in figure 5.4, an intermediate verification should already have been performed before the drawing of the SADT CM branches to ensure that no component from the problem description (activities, entities, resource and control) is missing. The verification of the complete SD will involve consistency checking of the entities from one activity to another and that all activities are linked together. Also, verification should ensure that the correct source and sinks have been put in place and linked to the following and preceding activity. The labels also need to be verified and cross-checked with the information in the completed Entity Table (step SD1) and Activity Table (Step SD5). Similarly all other information from the Activity Table (in step SD5) is used to verify the complete SD diagram. The validation of the SD can be achieved by comparing the problem situation with the SD layout and getting a sense about how closely it reflects the actual problem at this stage. Some discussion is probably required also at this stage to validate the SD. The complete SD diagram is a useful basis for this discussion.

5.7.2 Abstraction of Conceptual Model

The abstraction of the conceptual model from the SD is the next part of the CM process (Part B). We need to refer to completed Activity Table (Step SD5) and completed SD layout (Step SD6) to abstract the conceptual model. The flowchart of figure 5.10 shows the abstraction process. The whole procedure of abstraction goes through the following steps:

- Step CM1: Identify the modelling objective
- Step CM2: Identify the experimental factor
- Step CM3: Identify the Objective Activity(OA)
- Step CM4: Identify the Connected Activity (CA)
- Step CM5: Conduct CA impact test
- Step CM6: Check if all CAs have been analyzed

- Step CM7: Analysis of Controls, Resources and Entities
- Step CM8: Analysis of the queues
- Step CM9: Make abstracted diagram of the conceptual model
- Step CM10: Verification and Validation of conceptual model

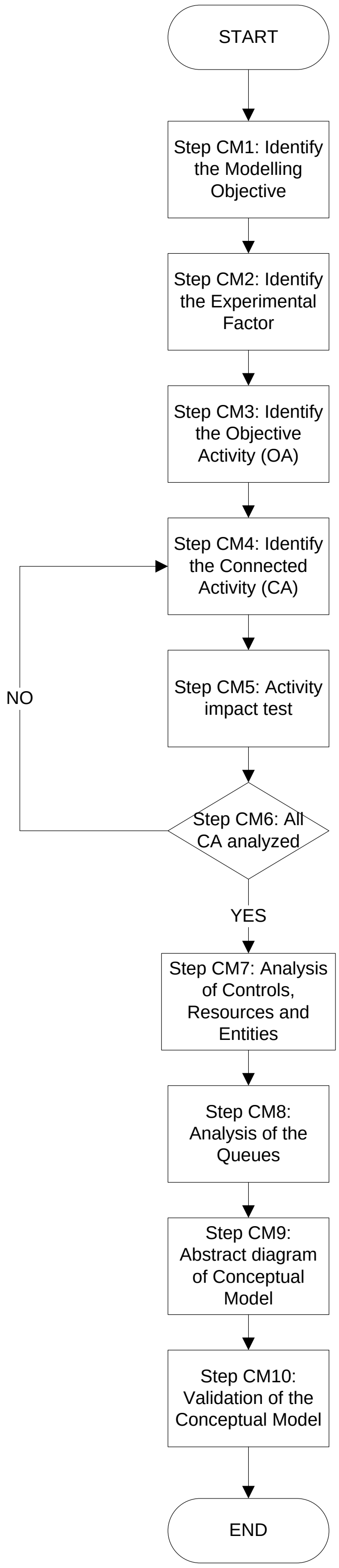

START
Step CM1: Identify the Modelling Objective
Step CM2: Identify the Experimental Factor
Step CM3: Identify the Objective Activity (OA)
Step CM4: Identify the Connected Activity (CA)
Step CM5: Activity impact test
NO
Step CM6: All CA analyzed
YES
Step CM7: Analysis of Controls, Resources and Entities
Step CM8: Analysis of the Queues
Step CM9: Abstract diagram of Conceptual Model
Step CM10: Validation of the Conceptual Model
END

Figure 5.10: Steps in Abstraction of Conceptual Model

Some steps are iterative (as shown) and will be explained accordingly in the following sections. An extended table, called the Scope Table (updated from the columns of Activity Table), is shown in figure 5.11 which is used to record all the information obtained in the different steps of part B. Similar to the Activity Table, the Scope Table is also updated as we move through the different steps of abstraction procedure. The first four columns in the Scope Table are simply copied from the Activity Table (obtained in Step SD5) before moving on with the abstraction phase.

Scope Table:

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?

Figure 5.11: Table for recording information in all steps of Part B

5.7.2.1 Step CM1: Identify the Modelling Objectives

This is the most critical part of the whole procedure. The modelling objectives need to be clearly identified at this stage before starting any abstraction. The user needs to be sure what they require (output/response) from their model. For example, if the objective is to measure the waiting time of certain entities in a queue, then this should be listed down as the primary focus other than any general project objectives (for example, time frame to develop the model, flexibility, interoperability, visual requirements etc.). This is because later stages of abstraction are strongly related to correct identification of the specific modelling objectives. If unsure about the modelling objectives then discussion with managers and other stakeholders needs to be conducted and the problem brainstormed in meetings to reach a consensus about the modelling objectives. In case of multiple modelling objectives, steps CM1 to step CM10 needs to be repeated for each modelling objective.

Bank Problem: Modelling Objective

In the case of the bank problem, the modelling objective is to study the output (response) for the customers' queue developing at the cashier (in this case, waiting times and not the queue size) and to check if this is within an acceptable limit (5 mins).

5.7.2.2 Step CM2: Identify the Experimental Factors

The experimental factors are the "input" to the model and needs to be identified as they play an important role in abstraction. They are the main parameters to test during experimentation with the model. This again can be precisely identified through discussion with managers and by knowing the allowable changes to the model.

Bank Problem: Experimental Factor

The only experimental factor identified for the bank problem is the cashier labeled "R2" from Activity Table (2) since the manager is interested to know if the investment in hiring an extra cashier can decrease the waiting time of the customers in the cashier queue to within an acceptable limit of 5 mins.

5.7.2.3 *Step CM3: Identify the Objective Activity (OA)*

The next step is to identify the OA. This is very closely related to the modelling objective and could be identified through discussion around the SD. This activity is the key activity that is needed to measure the identified response in terms of modelling objective discussed in step CM1. This can be the immediate activity linked to any of the modelling objectives, e.g. no of people in the queue, waiting time of the people in the queue, resources needed etc. It should be noted that this activity needs very careful selection as this is the starting point of the whole abstraction process. For example, an activity is considered an OA if the immediate queue behind that activity is to be considered for measuring the response (waiting time for instance) in the model.

Bank Problem: Objective Activity (OA)

In the bank problem, the OA is identified as A3 (cashier service) from Activity Table (2). By looking at the SD of figure 6 in Step SD6, it can easily be figured out that A3 is central to the modelling objective as the modelling objective requires accounting for the people in the cashier queue and no other activity other than A3 is responsible for developing queues for customers seeking cashier service.

5.7.2.4 *Step CM4: Identify the Connected Activities (CA)*

The Connected Activities (CA) are identified at this stage. The CA is any activity other than the OA (including source or sink activity). Only one CA is identified at a time and the step is repeated if a new CA is to be identified (after step CM6). The SD layout obtained in Step SD6 can be consulted for the identification of CAs. The main purpose of identifying the CAs is to compare each CA against OA during the abstraction procedure (step CM5). An important point to note is that no strict pattern is necessary for identification of the CAs, but it is better to move from left to right and top to bottom in the SD diagram to avoid any activity being missed. Also it is better to start with the source activities and end with the sink activities. Once an activity is identified, it is marked in the SD layout and subsequent steps are performed.

Bank Problem: Connected Activity (CA)

In the bank problem, the first CA identified for the bank problem is A1 (Customer arrival) being the source activity in the SD layout. This is also the first activity appearing from left in the layout.

5.7.2.5 Step CM5: Conduct CA impact test

After the identification of the CA, seek to answer the following question:

Does the CA *impact* on the OA?

Answering "yes" or "no", it is decided whether to include or exclude the CA from the conceptual model. At this point, the term "impacts" needs a detailed explanation to guide the user to a careful response to this question. Impact in a broad sense means any CA that has a significant impact on the accuracy of the model (i.e. on outputs/responses) with respect to the modelling objectives. A user should keep different aspects of "impact" in mind before applying this analysis. Also it is required by the user to understand the problem situation carefully (and hold meetings or discussions with managers if unsure) to answer this part. Some guidance on how to answer this question is provided below.

CA is considered to have an impact on the OA if:

The exclusion of the CA would result in an insufficiently accurate model

Some examples of these scenarios could be as follows:

- CA is a bottleneck activity in the model
- CA directly affect the flow of entities in the OA
- CA contains any shared resource, constraint or control
- CA is causing some variations in the OA (cycle time, maintenance, break-downs, etc.)
- CA is related with experimental factors
- CA is related with the modelling objective or OA in the whole process

As the analysis goes along, the results are recorded in the Scope Table. The activities are either kept or removed from the model depending upon the analysis of *impact* explained above. If the answer is “yes”, the activity is included in the Scope Table, but removed if the answer is “no”. The justification of the decision is also provided in the Scope Table.

Bank Problem: CA impact test

For the case of the bank problem, the first CA was identified to be A1 (customer arrival) in step CM4 and the OA was identified to be A3 (cashier service) in step 2. Having identified these two activities, analysis in step CM5 is carried out.

After careful examination we can decide that CA (A1) significantly impacts the OA (A3).

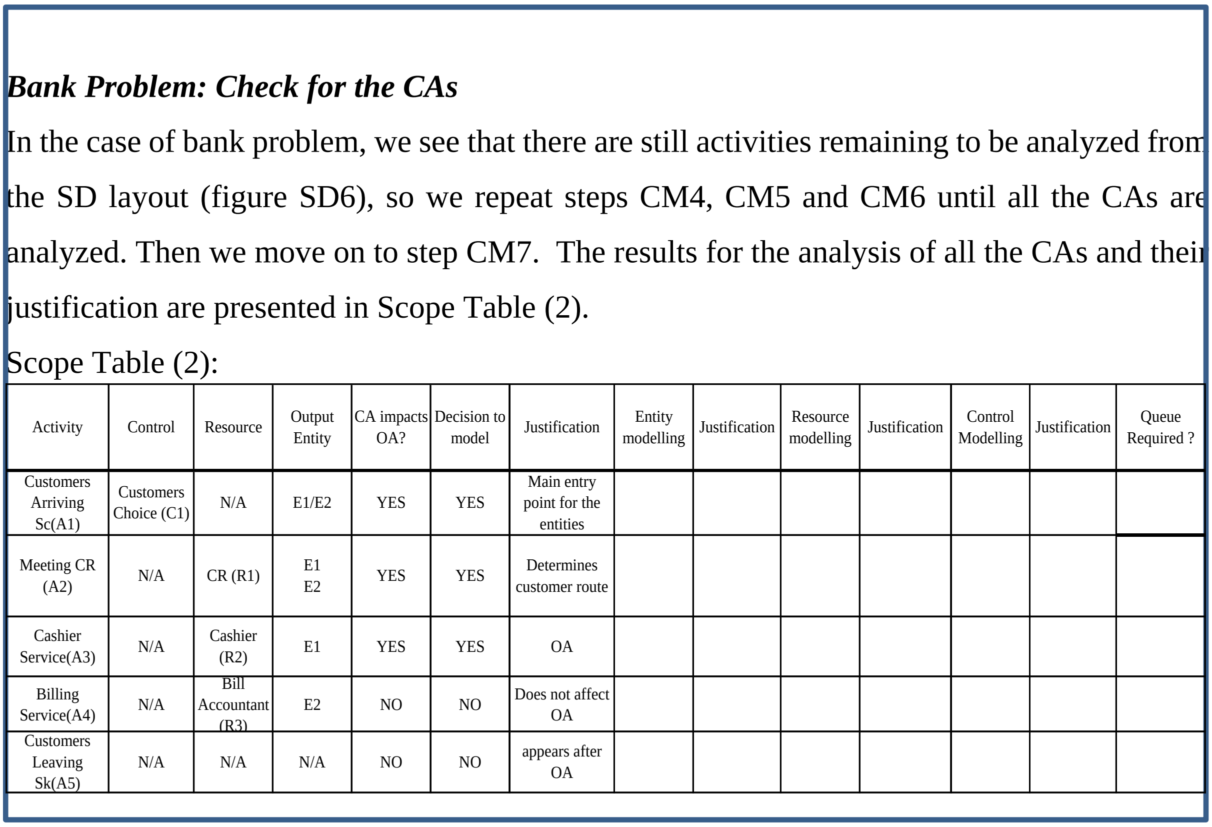
Hence A1 is added in Scope Table (1) as shown below with the justification. The justification in this case is that this activity is a source activity and initiates the arrival of entities in the system.

Scope Table (1): Activity A1 is added in the scope table after CA impact test

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities							
Meeting CR (A2)	N/A	CR (R1)	E1 E2										
Cashier Service(A3)	N/A	Cashier (R2)	E1										
Billing Service(A4)	N/A	Bill Accountant (R3)	E2										
Customers Leaving Sk(A5)	N/A	N/A	N/A										

5.7.2.6 Step CM6: Check if all CAs have been analysed

At this step, the SD layout is consulted again to see if all the CAs have been analyzed. If “yes” then the iterative process stops and we need to move to step CM7 else we need to repeat steps CM4, CM5 and CM6.



Bank Problem: Check for the CAs

In the case of bank problem, we see that there are still activities remaining to be analyzed fron the SD layout (figure SD6), so we repeat steps CM4, CM5 and CM6 until all the CAs are analyzed. Then we move on to step CM7. The results for the analysis of all the CAs and thei justification are presented in Scope Table (2).

Scope Table (2):

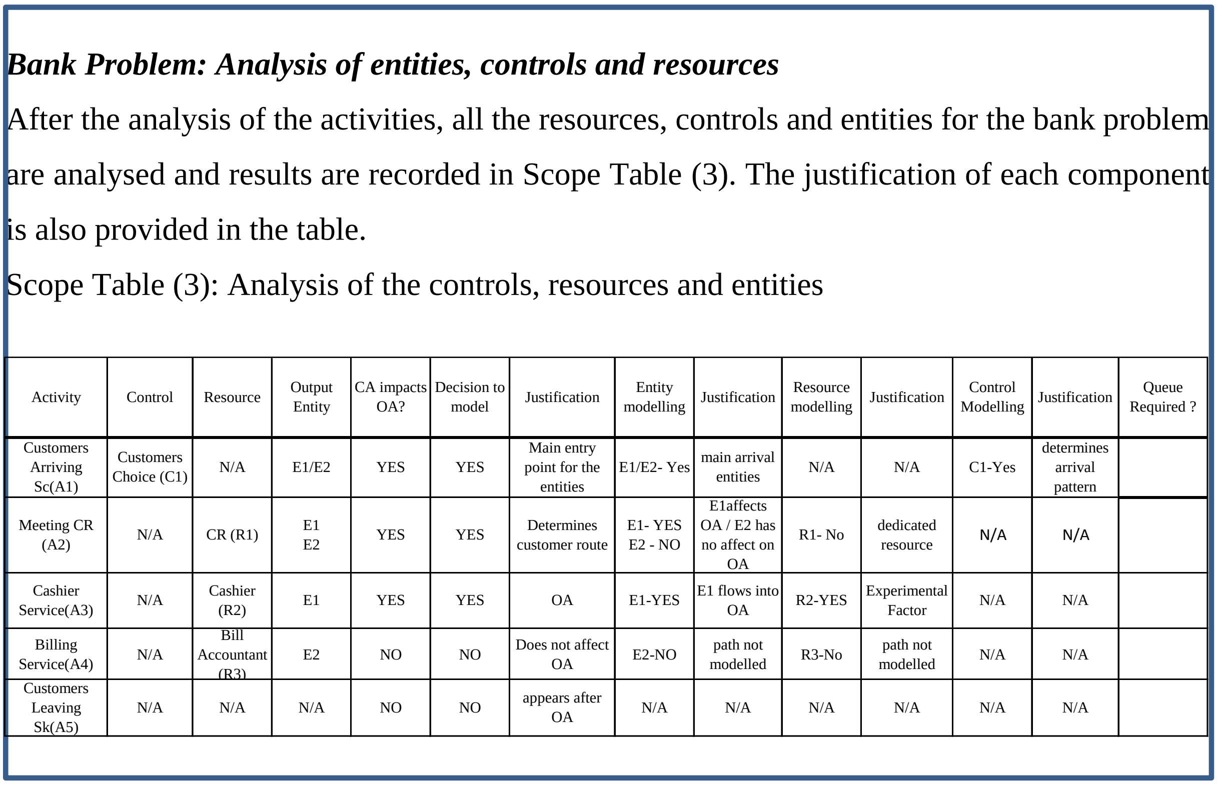
Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities							
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route							
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA							
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA							
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA							

5.7.2.7 Step CM7: Analysis of Entities, Resource and Controls

In this step, any entity, resource or control associated with the selected activity is analysed for inclusion or exclusion from the model. This is also recorded in separate columns of the Scope Table. This mainly requires considering if any of these particular components have some role in the model and it is valuable to keep them in model with respect to the modelling objectives. Separate columns are used to justify the inclusion/exclusion of the resources, entities and controls in the Scope Table. Some examples of reasons for the exclusion of these components are:

- Any resource or constraint that is linked to an activity which has been removed in the previous step will be removed.
- A resource can be excluded if it is a dedicated resource and always available
- An entity can be removed if it is not on the path affecting an OA

- A constraint can be removed from modelling if it is an unlikely event to happen during a specified model run (example non-shift times or environmental constraint)



Bank Problem: Analysis of entities, controls and resources

After the analysis of the activities, all the resources, controls and entities for the bank problem are analysed and results are recorded in Scope Table (3). The justification of each component is also provided in the table.

Scope Table (3): Analysis of the controls, resources and entities

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities	E1/E2- Yes	main arrival entities	N/A	N/A	C1-Yes	determines arrival pattern	
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route	E1- YES E2 - NO	E1affects OA / E2 has no affect on OA	R1- No	dedicated resource	N/A	N/A	
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA	E1-YES	E1 flows into OA	R2-YES	Experimental Factor	N/A	N/A	
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA	E2-NO	path not modelled	R3-No	path not modelled	N/A	N/A	
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA	N/A	N/A	N/A	N/A	N/A	N/A	

5.7.2.8 *Step CM8: Analysis of the queues*

To complete the Scope Table the requirement for queues is analyzed for each activity. The requirement for a queue is obvious if an activity is not a “source” or “sink” in the complete SD. The other cases of having/not having a queue need to be identified from the SD or through discussion with managers at the meetings and brainstorming all the aspects of the problem, for example enough capacity is available, infinite queue length permitted, experimental factors, model responses etc.. This information is recorded in the last column of the Scope Table. Also it needs to be identified whether single or multiple queues are needed for any activity. This justification can be put within the same column as inclusion or exclusion of the queue. In the case of a single activity that is fed by multiple queues, all queues are listed with labels besides the activity row, and likewise if the same queue is serving multiple activities then same label of the queue is assigned under different activities row.

Bank Problem: Queue analysis

In the case of the bank problem, the queue analysis against each selected activity is carried out and results are recorded in last column of Scope Table (4). There is a need for separate single queues for services before meeting CR and waiting for the cashier service.

Scope Table (4): Analysis of the Queues

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities	E1/E2- Yes	main arrival entities	N/A	N/A	C1-Yes	determines arrival pattern	No (Entry point)
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route	E1- YES E2 - NO	E1affects OA / E2 has no affect on OA	R1- No	dedicated resource	N/A	N/A	YES(Single)
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA	E1-YES	E1 flows into OA	R2-YES	Experimental Factor	N/A	N/A	YES(Single)
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA	E2-NO	path not modelled	R3-No	path not modelled	N/A	N/A	No (path not modelled)
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA	N/A	N/A	N/A	N/A	N/A	N/A	No (exit point)

5.7.2.9 Step CM9: Make an abstracted diagram of the Conceptual Model

After completion of the Scope Table, we can now proceed with the final step of drawing the layout of the conceptual model. This is drawn by taking information from the completed Scope Table (Step CM8) and completed SD layout in part A (Step SD6). All the components that have been analyzed and removed along with a justification in the completed scope table (Step CM8) will be removed in the final abstracted layout of the conceptual model. Any path that has not been modeled as recorded in the Scope Table will be removed. Also any control, resource, entity and activity that has not been included in the model as recorded in the Scope Table will be removed from the final abstracted diagram of conceptual model. This diagram is very easily drawn by removing any component from SD diagram produced in Step SD6, and connecting back the arrows in between without redrawing the whole figure.

Bank Problem: Abstracted Conceptual Model

The complete abstracted conceptual model of the bank problem is shown in figure 5.11 as per justified through Scope Table (4). Any component decided not to be modeled in Scope Table (4) has been removed and figure 5.6 (complete SD) has been updated accordingly to produce figure 5.12.

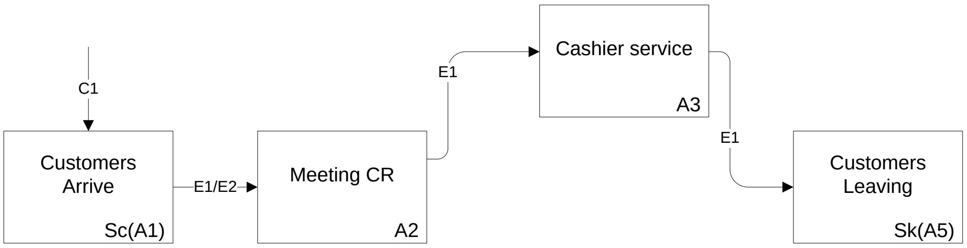


Figure 5.12: Abstracted lay out of the conceptual model for the bank problem

5.7.2.10 Step CM10: Validation of Conceptual Model

The final step is to perform a validation of the conceptual model generated as a result of the abstraction stage. Direct comparison of the SADT SD diagrams with the SADT CM diagrams would enable the abstractions in the model to be identified, debated and validated with subject matter experts.

Bank Problem: Validation of the conceptual model

Validation of the bank problem conceptual model is achieved by comparing the final abstracted layout of the conceptual model with the SD. It can easily be seen that the only branch removed from the SD part in the abstracted model is the Billing service branch which serves no variation on the OA and hence the modelling objective is not affected. Also any controls, resources and entities that are removed in the abstracted model are again verified through the justification provided in the scope table.

5.8 Application of SADT CM on Panorama TV case

In this section, we present the application of SADT CM on the Panorama Televisions production case taken from Robinson (2014). The problem has been directly copied form Robinson's book without any modification; just one paragraph about the manufacturing site and the associated figure has been left out. First, the complete problem description is provided, followed by the application of SADT CM.

5.8.1 Problem Description

"Panorama Televisions have been involved in the manufacturing of electrical goods since the early days of the radio. They now concentrate on the production of high quality, premium priced televisions for the international market. There are four televisions in their product range: small, medium, large and internet enabled. Last year, to meet increased demand, Panorama invested in a new television assembly plant. Also, after some negotiation with the unions, all areas of the site moved to continuous working over a five day week. However, the plant has never achieved its target throughput of 500 units per day. In fact, daily throughput is only just over 400 units. The plant is shown in figure 5.13.

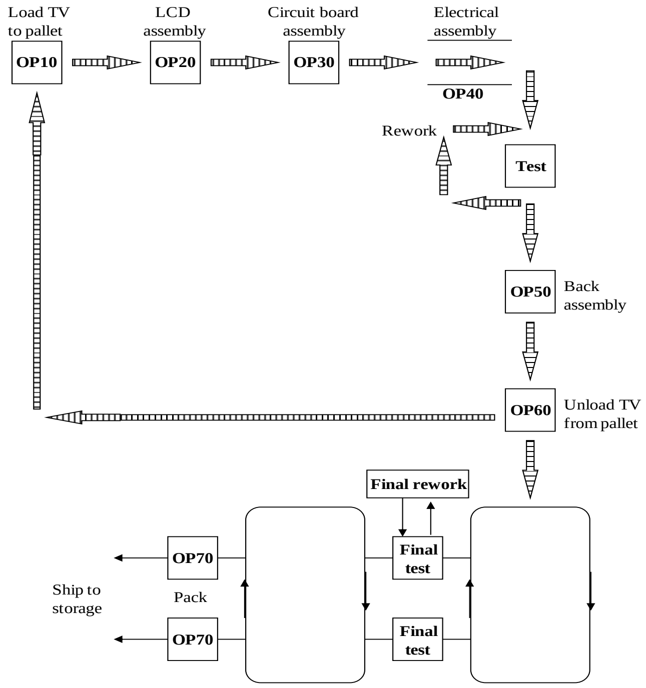


Figure 5.13: *Panorama Televisions: Television Assembly Plant, Robinson (2014)*

“Plastic moulded boxes are loaded to a pallet by an operator at OP10. A production schedule, which is based on projected demand, determines the type of box to be loaded (small, medium, large or for internet enabled). At OP20 the LCD is assembled to the box before the circuit board is added at OP30. The televisions travel on a conveyor and five manual operators assemble the electrical equipment, OP40. The television is then tested and any failures go to the rework area. Good televisions have the back assembled at OP50 and are unloaded from the line at OP60 by an operator. The empty pallets are returned by conveyor to OP10 and the televisions are stored on a circular sling conveyor. A television is taken from the conveyor when a final test booth becomes available. Televisions failing this test are sent for final rework. Televisions passing are stored on another sling conveyor and are packed at OP70. Packed televisions are transported to the warehouse by forklift truck. The final test and packing area are often short of work and there is enough spare capacity to achieve 500 units per day. The management at Panorama believes that the throughput problem is a result of the number of stoppages on the main assembly line. There are a significant number of breakdowns on automated machines, and set-ups are required every time there is a change of product in the production schedule. There is only one maintenance engineer per shift who is required to attend all machine breakdowns and set-ups. There seems to be little opportunity to improve the efficiency of the machines, nor can the production schedule be changed since it is driven by customer demand. The solution being considered is to increase the buffering between the operations to dampen the effects of stoppages. Design engineers have considered this proposal and believe that, due to physical constraints on space, the buffering could be increased by a maximum of 200%. This will also require further pallets to be bought. In fact, there is some uncertainty as to whether enough pallets are currently being used. Increasing the number of pallets may provide a solution without the need for further storage. Extra storage is expensive, so before investing Panorama want to be sure it is necessary. Also, special pallets have to be used at a cost of $1,000 each, so it is important to minimize the number required. Target throughput must be achieved, but expenditure should be kept to a minimum. The management at Panorama is looking for some proposals on how to improve the line within ten working days”

Robinson (2014, pp. 311-314)

5.8.2 Application of SADT CM

Now all the steps of SADT CM are performed to arrive at the solution (conceptual model) for the Panorama TV production case.

5.8.2.1 Step SD1: Identification of entities

The Entity Table is produced as shown in Table 5.1 in this step.

Table 5.1: Entity Table for Panorama TV case

Entity	**Label**
TV	E1
Pallet	E2
Completed TV	E3

5.8.2.2 Step SD2-SD4: Identification of Activities, Controls and Resources

The combined outcomes of steps SD2 to SD4 are shown in table 5.2(a). All the activities, resources and controls identified in Panorama TV case are recorded along with the labels for each component.

Table 5.2(a): information recorded in steps SD2-SD4 for Panorama TV case

Activity	**Control**	**Resource**
TV arrives into system Sc(OP9)	Production Schedule (C1)	Forklift truck (T1)
Boxes loaded to pallet (OP10)	n/a	Shift operator (R1)
LCD assembly (OP20)	n/a	Maintenance operator (M1)
Circuit Board Assembly (OP30)	n/a	Maintenance operator(M1)
Electrical Assembly (OP40)	n/a	Group of shift operators (R2-5)
Test area one (T1)	n/a	Maintenance operator (M1)
Rework area one (RT1)	n/a	Shift operator (R6)
Back Assembly (OP50)	n/a	Maintenance operator (M1)
Finished TV Unloaded from line (OP60)	n/a	Shift operator (R7)
Test Area Two (T2)	n/a	Maintenance operator (M1)
Rework area two (RT2)	n/a	Shift operator (R8)
Packaging (OP70)	n/a	Shift operator (R9)
Finished TV Transported to Warehouse (OP80)	n/a	Forklift truck (T2)
Product exit form system SK(OP90)	n/a	n/a

5.8.2.3 Step SD5: Drawing SADT branches

In this step, first table 5.2(a) is updated with the final column of “output entity” as shown in table 5.2 (b), and then corresponding SADT CM branches are created as shown in figure 5.14

Table 5.2(b): Output Activity in Panorama TV case

Activity	Control	Resource	Output Entity
TV arrives into system Sc(OP9)	Production Schedule (C1)	Forklift truck (T1)	E1
Boxes loaded to pallet (OP10)	n/a	Shift operator (R1)	E1/E2
LCD assembly (OP20)	n/a	Maintenance operator (M1)	E1/E2
Circuit Board Assembly (OP30)	n/a	Maintenance operator(M1)	E1/E2
Electrical Assembly (OP40)	n/a	Group of shift operators (R2-5)	E1/E2
Test area one (T1)	n/a	Maintenance operator (M1)	E1/E2 E1/E2
Rework area one (RT1)	n/a	Shift operator (R6)	E1/E2
Back Assembly (OP50)	n/a	Maintenance operator (M1)	E2/E3
Finished TV Unloaded from line (OP60)	n/a	Shift operator (R7)	E3 E2
Test Area Two (T2)	n/a	Maintenance operator (M1)	E3 E3
Rework area two (RT2)	n/a	Shift operator (R8)	E3
Packaging (OP70)	n/a	Shift operator (R9)	E3
Finished TV Transported to Warehouse (OP80)	n/a	Forklift truck (T2)	E3
Product exit form system SK(OP90)	n/a	n/a	n/a

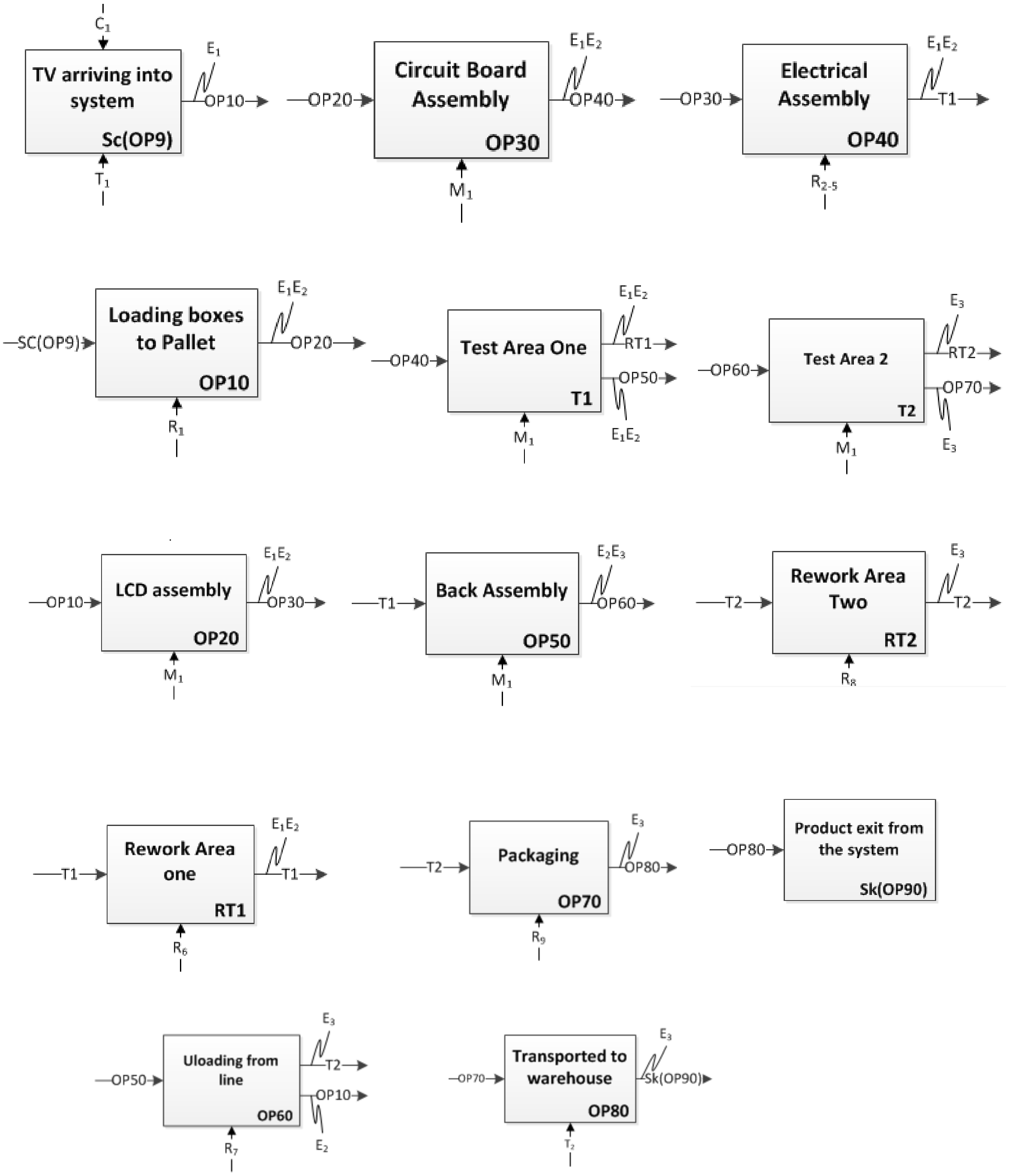


Figure 5.14: SADT CM branches for Panorama TV case

5.8.2.4 Step SD6: Generation of complete SD

The complete SD for the Panorama TV case is shown in figure 5.15. This is obtained by combining all SADT CM branches developed in step SD5.

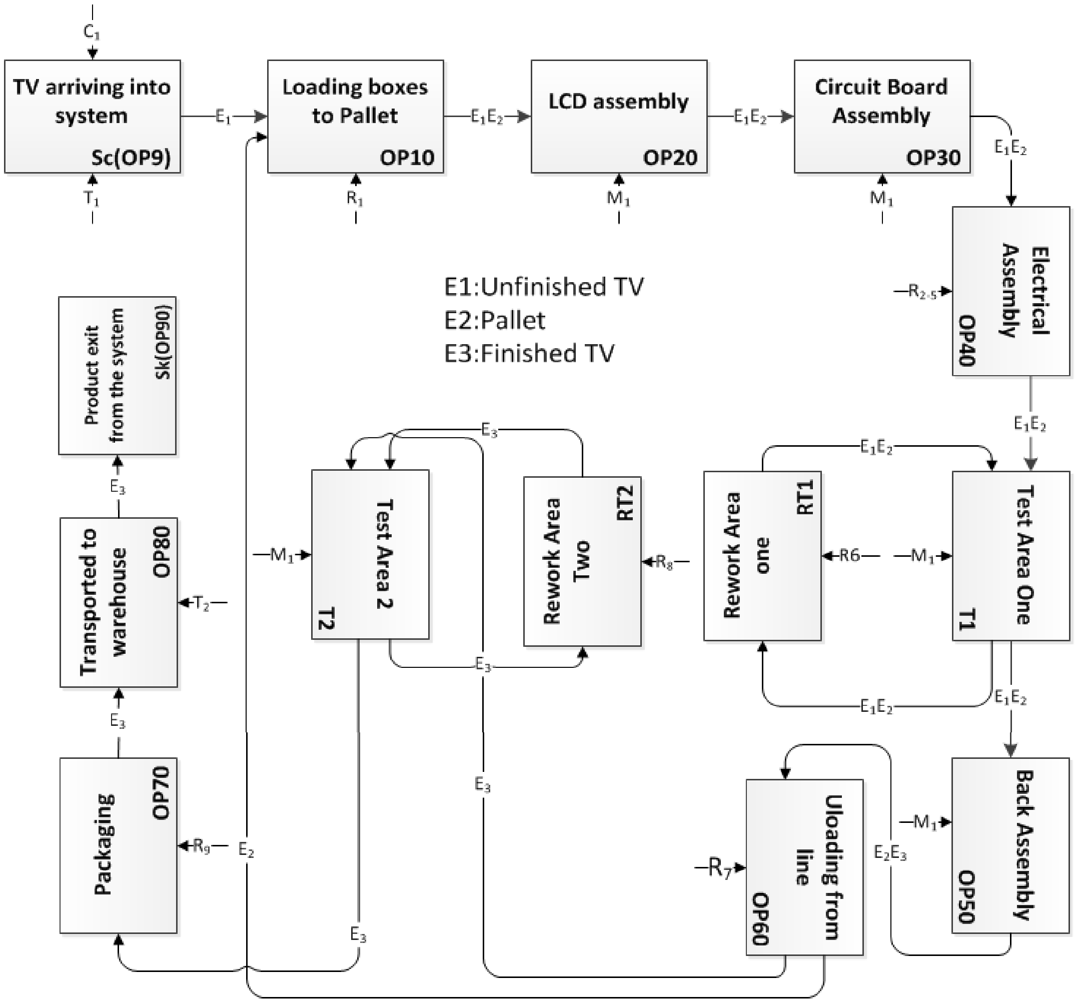


Figure 5.15: Complete SD from SADT CM branches

5.8.2.5 *Step SD7: Verification and Validation of SD*

All SADT CM branches are cross-checked with the entity and activity tables (5.1, 5.2 (a, b)) to ensure that every activity has been drawn into the branches. Also, resources and controls are checked against table 5.2 (a, b). The complete SD diagram is visually checked to ensure that all branches have been connected correctly along with any shared resources and controls.

5.8.2.6 *Step CM1: Identify the modelling objective*

The modelling objective for Panorama TV case is identified as follows:
"To determine whether 500 units of throughput per day can be achieved"

5.8.2.7 *Step CM2: Identify the Experimental Factor*

The two experimental factor identified in Panorama TV case are as follows:

1) The number of pellets
2) The size of the buffers (conveyers) between the operations

5.8.2.8 *Step CM3: Identify the Objective Activity (OA)*

The OA in the Panorama TV case is identified as OP60. The reason for identifying OP60 as OA is because it is linked most closely to the number of finished TVs produced in a day. Also it is the last activity which determines the throughput of finished TVs as before this activity TVs can fail the testing procedure and could be returned.

5.8.2.9 *Step CM4: Identify the Connected Activity (CA)*

The first CA identified for Panorama TV case is SC (OP9). The reason to select Sc (OP9) as the first CA is that it appears as the first source in the system and also appears first in the complete SD layout of figure 5.11.

5.8.2.10 Step CM5: Conduct CA impact test

After conducting the CA impact test, it can easily be seen that CA (Sc (OP9)) significantly impacts the OA (OP60), since it directly affects the flow of entities in the OA being the source activity. (This result is then added in the scope table of 5.3 and the justification for this decision is also recorded. These results are recorded in the columns “CA impacts OA?”, “Decision to model” and “Justification” next to the decision.

5.8.2.11 Step CM6: Check if all CAs have been analysed

Since we are left with many CAs after the first iteration, the whole procedure from step CM4 to CM6 is repeated for the Panorama TV case and the results are updated in table 5.3. Once we are done with all the CAs, then we can move on to step CM7.

5.8.2.12 Step CM7: Analysis of Entities, Resources and Controls

All the analysis of entities, resources and controls are performed after the CA impact test. The results are recorded in “Entity Modelling”, “Resource Modelling”, and “Control Modelling” respectively in table 5.3 along with the justification in each case.

5.8.2.13 Step CM8: Analysis of the queues

All the activities are analysed for the queue requirement and the results are recorded in the last column of table 5.3. The justification for the decision and number of queues required is also provided.

5.8.2.14 Step CM9: Make an abstracted diagram of the Conceptual Model

The final abstracted layout of the conceptual model is shown in figure 5.16. This diagrammatic representation of the conceptual model is derived from table 5.3 by removing any components of the SD that is excluded in this table.

5.8.2.15 Step CM10: Validation of Conceptual Model

The final validation of the conceptual model in the Panorama TV case is performed after comparing the complete SD layout of figure 5.14 and conceptual model layout of figure 5.16. Any component removed from the conceptual model is double checked with managers and system experts before proceeding to actual model development.

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
TV arrives into system Sc(OP9)	Production Schedule (C1)	Forklift truck (T1)	E1	YES	YES	Initiates arrival of entities in the system	E1- Yes	Main entity initializing the flow of product	T1- No	Assuming materials are always avaialble	C1-Yes	Determines production schedule	No
Boxes loaded to pallet (OP10)	n/a	Shift operator (R1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R1-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
LCD assembly (OP20)	n/a	Maintenance operator (M1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Circuit Board Assembly (OP30)	n/a	Maintenance operator(M1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Electrical Assembly (OP40)	n/a	Group of shift operators (R2-5)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R2-5-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Test area one (T1)	n/a	Maintenance operator (M1)	E1E2 E1E2	YES	YES	contributes in the throughput of the final tested TVs	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Rework area one (RT1)	n/a	Shift operator (R6)	E1E2	YES	YES	contributes in the throughput of the final tested TVs	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R6-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Back Assembly (OP50)	n/a	Maintenance operator (M1)	E2E3	YES	YES	directly contrbutes to entities in OA	E2E3- Yes	E2 are experimental factors. E2E3 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Finished TV Unloaded from line (OP60)	n/a	Shift operator (R7)	E3 E2	YES	YES	OA	E3- Yes E2 - NO	E3 finished Tvs (determines thorughput) E2 returned conveyers not modelled	R7-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Test Area Two (T2)	n/a	Maintenance operator (M1)	E3 E3	NO	No	Does not effect OA	E3-No	Path not modelled	M1-No	path not modelled	n/a	n/a	No (path not modelled)
Rework area two (RT2)	n/a	Shift operator (R8)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	R8-No	Dedicated Resource	n/a	n/a	No (path not modelled)
Packaging (OP70)	n/a	Shift operator (R9)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	R9-No	Dedicated Resource	n/a	n/a	No (path not modelled)
Finished TV Transported to Warehouse (OP80)	n/a	Forklift truck (T2)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	T2-No	Does not effect throughput of finshed TVs	n/a	n/a	No (path not modelled)
Product exit form system SK(OP90)	n/a	n/a	n/a	NO	No	Does not effect OA	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Table 5.3: Scope table for Panorama TV case (for recording results in steps CM4 to CM8)

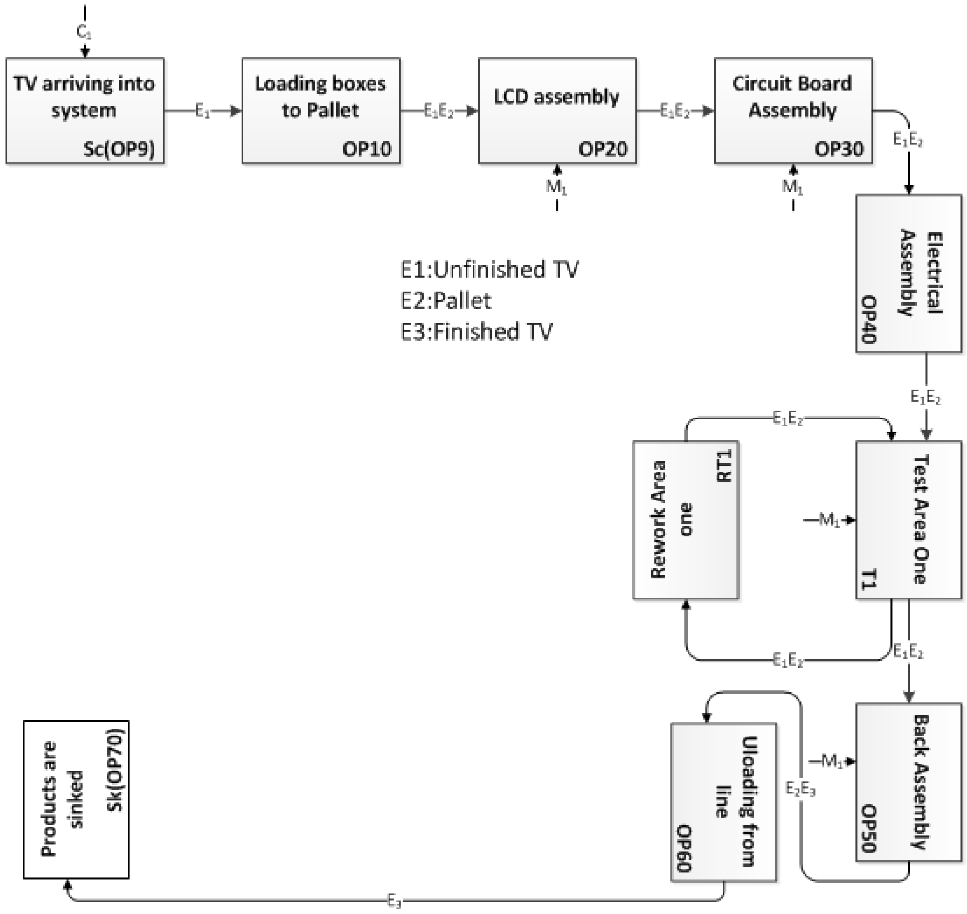


Figure 5.16: Abstracted layout of conceptual model for Panorama TV case

5.9 Discussion

The chapter presents a structured approach for CM using the Structured Analysis and Design Technique (SADT) from SRE. In this chapter, SADT is shown to be successful in adaptation for M&S CM. The flexible notation and style of SADT makes it possible to be used in M&S CM. The adapted framework has undergone some modification from SADT style and is renamed as SADT CM. This modification is mainly in the notation and hierarchal structure of the approach, i.e. bottom-up approach instead of top-down. The framework has been explained with the help of a simple bank problem. Both parts of CM, i.e. SD and abstraction as explained in Robinson (2008a) are addressed in the framework. It is shown that the structural and visual approach of SADT CM makes it easy to move from the SD to the abstracted conceptual model.

In the last section, the SADT CM framework is applied on the Panorama TV case and the results from both parts of framework are presented.

The next chapter reports on the detailed testing of SADT CM including pilot test and the final test.

6 Testing of SADT CM

This chapter reports on the detail testing of the CM framework SADT CM, presented in chapter 5. The testing procedures involved pilot testing, modifications, final test and analysis of the results for the framework.

The chapter is structured as follows. The next section describes the experimentation script which was used for the testing. This is followed by sections that describe the test cases, method of analysis of results, pilot testing and final testing. The last section provides a brief summary of the chapter.

6.1 Experiment Script

In this section, the experiment script is presented which was used during the pilot testing and final testing.

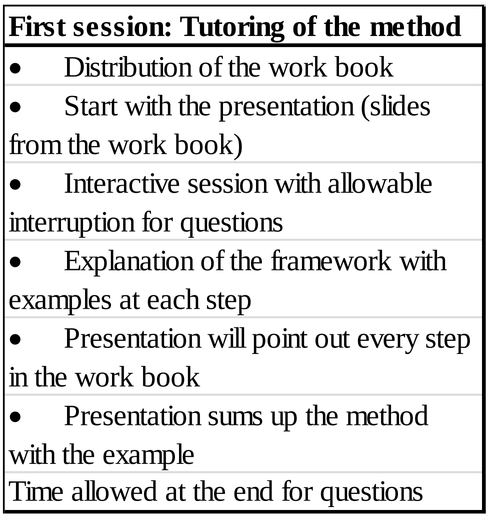
SADT CM framework experiment

Total time: 3 hours
Sessions: two
First session: Tutoring on the SADT CM framework (90 mins)
Second session: practice with a test case (70 mins)
Questionnaire feedback: (20 mins)

The activities of the two sessions of the experiment and feedback part are summarised in tables 6.1, 6.2, and 6.3. As stated in the script, two separate parts were planned for evaluating the framework. The first part was dedicated for teaching the framework to the participants. This part is detailed in table 6.1 showing different activities performed in this part. The tutoring of the framework is supported through a comprehensive user-guide workbook provided to the participants at the time of pilot testing. The user guide is included in the appendix of the thesis. This user guide omits the theoretical and detailed explanation of the SADT CM included in chapter 5, and lists only the key steps with examples. PowerPoint slides from the work book were also used for tutoring along with simple examples from work book. In the final testing, this workbook was provided one week in advance of the testing. The reason is to get them familiar with the overall idea of the framework before the main testing. This helped to speed up the process of tutoring as the framework was large to cover within the prescribed time limit. The session was aimed to be interactive which allowed interruptions for any queries and also provided some time at the end for questions.

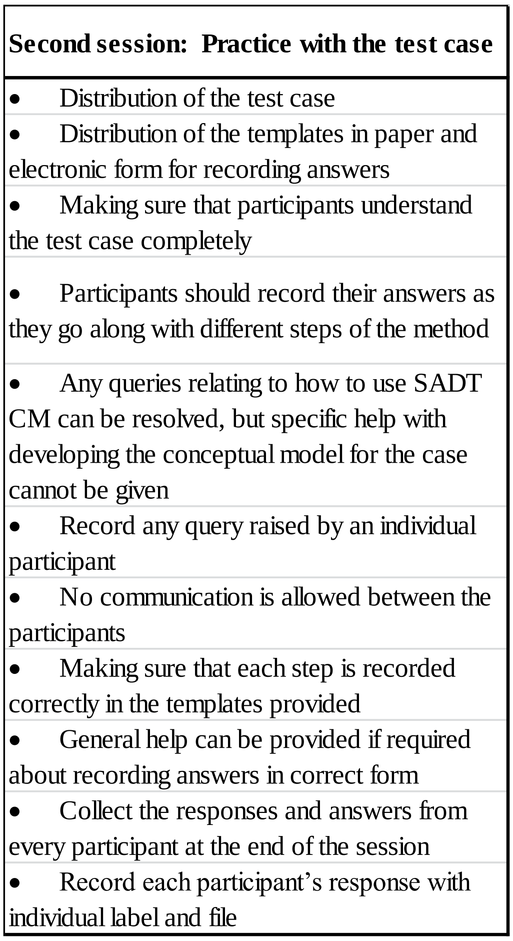
The second part was the actual application of the framework on the test cases provided to the participants. The activities of this session are listed in table 6.2. Different test cases were provided to each participant during the pilot test and in the final test. This made it possible to work with more than one participant during the testing and also helped to observe how participants managed to use SADT CM for different simulation problems.

Table 6.1: First session of SADT CM testing



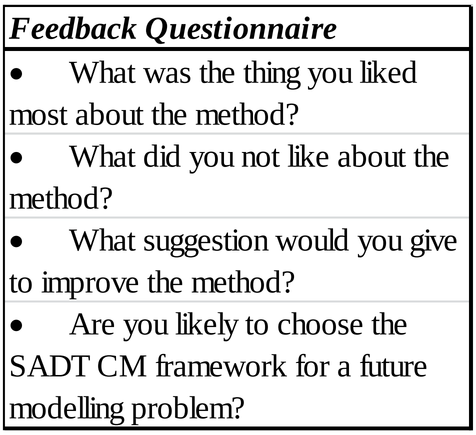
First session: Tutoring of the method
• Distribution of the work book
• Start with the presentation (slides from the work book)
• Interactive session with allowable interruption for questions
• Explanation of the framework with examples at each step
• Presentation will point out every step in the work book
• Presentation sums up the method with the example
Time allowed at the end for questions

Table 6.2: second session of SADT CM testing



Second session: Practice with the test case
• Distribution of the test case
• Distribution of the templates in paper and electronic form for recording answers
• Making sure that participants understand the test case completely
• Participants should record their answers as they go along with different steps of the method
• Any queries relating to how to use SADT CM can be resolved, but specific help with developing the conceptual model for the case cannot be given
• Record any query raised by an individual participant
• No communication is allowed between the participants
• Making sure that each step is recorded correctly in the templates provided
• General help can be provided if required about recording answers in correct form
• Collect the responses and answers from every participant at the end of the session
• Record each participant’s response with individual label and file

Table 6.3: feedback questionnaire in SADT CM experiment



Feedback Questionnaire
• What was the thing you liked most about the method?
• What did you not like about the method?
• What suggestion would you give to improve the method?
• Are you likely to choose the SADT CM framework for a future modelling problem?

The last part of SADT CM testing is about collecting participants' responses and feedback on the framework. The feedback questionnaire is included in table 6.3. This small questionnaire needed to be answered by every participant either at the end of the session, or within two days of the session.

6.2 Description of the test cases

The following three test cases were used for the pilot tests.

Panorama TV case

This test case was taken from Robinson (2014) whose complete problem description is provided in appendix 2. Same content from the test case had been used as provided in section 5.8.1.

Wardeon Cinema case

This test case was taken from Robinson (2014) whose complete problem description is provided in appendix 2. Mostly same test case content has been used as provided in section 4.3.1.1 except a minor difference. The difference is that for the PREmod framework (chapter 4), additional constraints about location and compatibility of the software system was added in the problem description to test the approach, which was not added for SADT CM testing.

King's Road Nursery problem

This test case is similar to the Happy Faces Day Care problem of section 4.3.2.1. Happy Faces Day Care is a North American derivative of the King's Road problem and is also discussed in Robinson (2012). The complete problem description is provided in appendix 2. For the Happy Faces Day Care problem the street names were changed and the flow traffic at the nursery is different. The on-street parkings for residents are specifically marked in the layout diagram given in the problem description. The nursery parking is proposed at the front of the building facing King's Road. The concerns are specific about parking spaces and exiting queues on King's Road and there is no extension of the study included in the problem description.

6.3 Analysis of the results

The analysis of all the tests were performed through comparison with the reference solution included in appendix 3 of all the test cases. These reference solutions were generated as a result of the application of SADT CM on these test cases by the author. As explained in section 3.3, these reference solutions are generated as a part of the framework development and iteratively every step is checked thoroughly and verified before adding the step into the framework. The solution of that part is then taken as the reference solution. In the case of the pilot tests, the comments of the participants were used to modify the framework before conducting the final test. In the case of the final tests, all key results were recorded and analysed along with the responses for the questionnaire from all the participants.

Next we present the pilot testing of SADT CM and then the final test.

6.4 Pilot Testing of SADT CM

The pilot testing was conducted with two participants and the experiment script described in the previous section was followed. These were the same participants who participated in the pilot testing of the PREmod framework. The idea was to take the feedback on strengths of the SADT CM when compared with PREmod. Participants were tutored with the small ATM problem which is discussed in chapter 5 and then they were asked to apply the framework on a test problem. The main results and comments were recorded and necessary modifications were made before conducting the final test. We present the main results and comments from both pilot tests in the following section.

6.4.1 Pilot testing 1

Background

Participant “A” was an OR PhD student from an OR modelling background and had a very good familiarity and experience of discrete-event simulation modelling. This participant had been working for industrial projects and firms for a few years before starting her research in OR simulation modelling.

Test case

Panorama TV case was used in this pilot test.

Participant responses

The participant successfully developed the entity, activity and scope tables along with the SADT CM branches for the system description and the abstraction stages. The key responses are summarised below.

- *What was the thing you liked most about the method?*

“I liked the fact that conceptual modelling can be structured! Eventually people will stop considering it as a process of art. In a more complex situation than the one given to me, I felt that by following these steps, I could easily end up with the ‘right’ model. Without following rules, it could have taken me more time and effort to come up with the same model”

- *What you did not like about the method?*

“Rewriting entities, activities and constraints for the abstraction parts seems time consuming to me. In addition, drawing activities individually did not have any specific value for me. Also it may be found confusing by some the change in arrows meaning (i.e. from activities arrows to entities arrows)”

- *What suggestion would you give to improve the method?*

“I think that it is worth seeing it as software. Electronic version could save time. Perhaps then the step of output entities is worth keeping as is, so that individual activities and whole model are drawn automatically (and simultaneously with filling the activities) by the programme. Therefore, you have an immediate visual validation tool and also it could be fantastic if the final [model] could be directly exported to simulation software. Tasks that require dropping a number of activities should be preferred so people appreciate the part of abstraction”

- *Are you likely to choose the SADT CM framework for a future modelling problem?*

"If it was electronic, yes! As it is now, in complex situations, I think I would consider following some steps of the method, but I would prefer to do simultaneously the abstraction to save some time as well as I would skip the output entities and drawing the individual branches"

6.4.2 Pilot testing 2

Background:

Participant "B" was from an OR and analytics background with a strong interest in e-commerce, business analytics, prescriptive analytics and big data. He was studying for a PhD in OR and analytics at the time of the pilot test. The participant did have a background of OR modelling using simulation and statistical methods and had good familiarity with discrete-event simulation modelling.

Test case

King's Road nursery problem was used for this pilot test.

Participant Responses

The participant came up with a slightly different version of the conceptual model; however, the key results including the scope table and abstracted layout did not deviate much from the reference solution. The key responses from this pilot test are summarised below.

- *What was the thing you liked most about the method?*

"The earlier stages were a little confusing at first, trying to decide what should be included or not, but after that it was useful for viewing the elements and how they were connected. Up to SD5 was useful for fleshing this out"

- *What you did not like about the method?*

“The process of making the final diagram (SD6) became very messy - too many interconnections. It became confusing to write and to read. What also strikes me is that the data is already there so it would be feasible for these diagrams to be automatically generated, which would take a lot of the headache away. I felt in the example, given the identification of experimental factor, was not as obvious as it could be. Same can be said of the objective activities - the problem given infers this is in fact the queues which are not particularly discussed in the slides. It takes a (admittedly small) leap to realise it is the exits from Kings Rd., but then also the literature suggests 1x objective activity whereas the example gives 2x objective activities. It becomes relative clear early on in CM5 that the only activity I was going to exclude is residents coming in to park, and then it feels like a lot of work to just be excluding one thing that I probably could have identified without the tool”

- *What suggestion would you give to improve the method?*

“I think the example/question given would be the obvious place to start. It is unsatisfying for there to be so little change (only one thing removed) after a relatively involved process. Even doing the other question [the first concern] would lead to many other factors being removed which may be more satisfying. It is a balance obviously, but the last thing you would want is for the user to be frustrated by the experience as this will affect how they respond to the tool and also how likely they are to use it future”

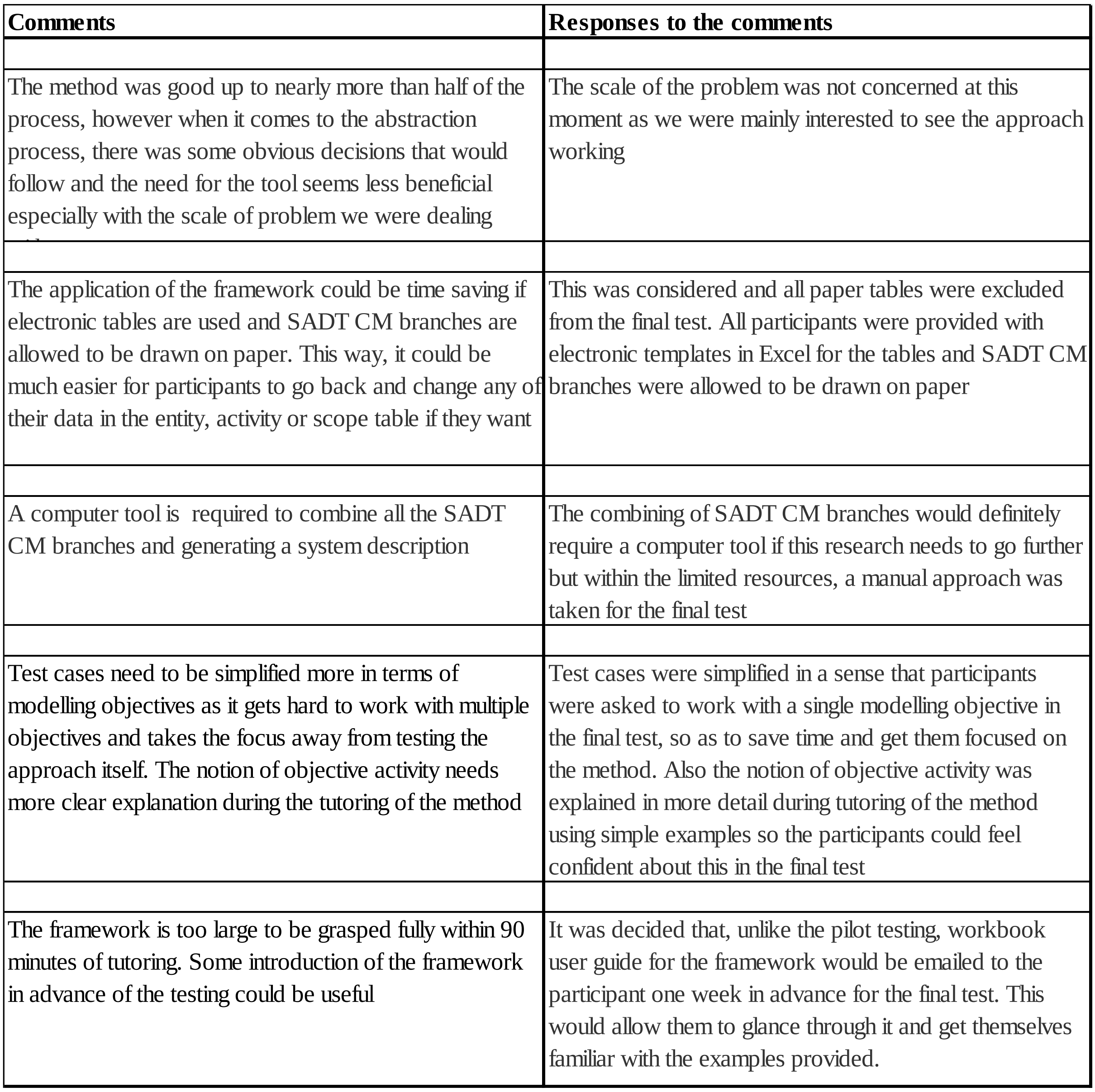
- *Are you likely to choose the SADT CM framework for a future modelling problem?*

“My feeling is that I wouldn't want to use it all the way through for this specific problem. I would possibly start using it and then stop when I have worked out what I need. My feeling is it will be better suited to some problems than others, and/or it would be something you may use to help frame the problem/system, but not necessarily complete. There is a lot of benefit in the fleshing out and mapping, but it then becomes diminishing returns as the work vs. new insight balance starts to go the other way. That said, if this was more automated - i.e. if I wasn't doing the same tasks multiple times, and the diagrams built themselves, etc. I suspect I would be far more likely to use it”

6.4.3 Reflections from pilot testing

After comparing the pilot responses, the overall impression was taken and any key modifications required were made before the final test. The important changes after the pilot tests in response to the comments are shown in table 6.4. Next we present the proceedings of the final testing in the next section.

Table 6.4: Comments from pilot testing and responses to the comments



Comments	**Responses to the comments**
The method was good up to nearly more than half of the process, however when it comes to the abstraction process, there was some obvious decisions that would follow and the need for the tool seems less beneficial especially with the scale of problem we were dealing	The scale of the problem was not concerned at this moment as we were mainly interested to see the approach working
The application of the framework could be time saving if electronic tables are used and SADT CM branches are allowed to be drawn on paper. This way, it could be much easier for participants to go back and change any of their data in the entity, activity or scope table if they want	This was considered and all paper tables were excluded from the final test. All participants were provided with electronic templates in Excel for the tables and SADT CM branches were allowed to be drawn on paper
A computer tool is required to combine all the SADT CM branches and generating a system description	The combining of SADT CM branches would definitely require a computer tool if this research needs to go further but within the limited resources, a manual approach was taken for the final test
Test cases need to be simplified more in terms of modelling objectives as it gets hard to work with multiple objectives and takes the focus away from testing the approach itself. The notion of objective activity needs more clear explanation during the tutoring of the method	Test cases were simplified in a sense that participants were asked to work with a single modelling objective in the final test, so as to save time and get them focused on the method. Also the notion of objective activity was explained in more detail during tutoring of the method using simple examples so the participants could feel confident about this in the final test
The framework is too large to be grasped fully within 90 minutes of tutoring. Some introduction of the framework in advance of the testing could be useful	It was decided that, unlike the pilot testing, workbook user guide for the framework would be emailed to the participant one week in advance for the final test. This would allow them to glance through it and get themselves familiar with the examples provided.

6.5 Final Testing of SADT CM

This section presents the proceedings of the final test conducted for the evaluation of the SADT CM framework. We start with the details of the test, resources required, test proceedings and then present the solution for the tests.

Test details

The final test for SADT CM was conducted in November, 2014 at Loughborough University. The total time of the test was three hours and three participants were invited with good familiarity of discrete-event simulation. In the following, we present the proceedings for the final test briefly.

Test Material

SADT CM workbook (hard copy, provided one week in advance), electronic version of all the templates which were to be used during the testing (emailed one hour before the start of the test).

Test proceedings

Each solution was recorded carefully using electronic templates in Excel files; this include recording of responses for two main tables during the SADT CM process: Entity – Activity Table (from system description), Scope table (from abstraction part). SADT CM branches and a complete system description were drawn clearly using paper and pencil by the participants. . These drawings were translated to Microsoft Visio drawings by the author after the test for documentation and analysis purposes. No alteration was done to the diagrams during this translation and the paper counter-part has also been saved for reference. The whole test was conducted as per the strict rules explained in the experiment script. Three different test cases were used to test generality of the method. All the data (solution of the test cases) and feedback was securely collected at the end of the test which was compared against reference solutions for evaluation of the method.

All the test results (including original solutions on Excel templates, SADT CM diagrams and final abstracted diagram of the conceptual model are provided in the next section. All manual diagrams have been converted to Visio drawings by the author for ease of reading. No alteration was done to the diagrams during the translation.

6.5.1 Participant A

Participant A was a PhD student in both the Management Science and Operations Management Group at the School of Business and Economics and in Aeronautical and Automotive Engineering in Loughborough University. She started her PhD research about one year before the test and mainly focuses on developing combined simulation and optimisation models that can be used to support optimal staff allocation for a police force.

6.5.1.1 Test case

Panorama TV case was used for participant A.

6.5.1.2 Results

The results form participant A are summarised below.

Entity-Table

The entity table produced form participant A is shown in table 6.5.

Table 6.5: Entity table (participant A)

Entity	**Labels**
Plastic moulded box	E1
LCD	E2
Circuit board	E3
Electrical equipment	E4
Back	E5

Activity Table

Activity table of 6.6 was produced from participant A.

Table 6.6: Activity table (participant A)

Activity	**Control**	**Resource**	Output Entity
A1 Load TV to pallet (OP10)	N/A	R1(Pallet)	E1
A2 LCD assembly (OP20)	N/A	R1 + R2(LCD assembly machine)	E1+E2
A3 Circuit board assembly (OP30)	N/A	R1 + R3(Citcuit board assembly machine)	E1+E2+E3
A4 Electrical assembly (OP40)	N/A	R1 + R4(Five manual operator)	E1+E2+E3+E4
A5 Test	N/A	R1 + R5(Test resource)	E1+E2+E3+E4
A6 Rework	pass / fail (C1)	R1 + R6(Rework resource)	E1+E2+E3+E4
A7 Back assembly (OP50)	N/A	R1 + R7(Back assembly machine)	E1+E2+E3+E4 +E5
A8 Unloaded TV from pallet (OP60)	N/A	R1 + R8(Unloaded TV resource)	E1+E2+E3+E4 +E5
A9 Wait on circular sling conveyor 1	N/A	R9(Circular sling conveyor 1 capacity)	E1+E2+E3+E4 +E5
A10 Final test	Test booth avaialble(C 2)	R10(Final test booth)	E1+E2+E3+E4 +E5
A11 Final rework	Pass or fail (C3)	R11(Final rework resource)	E1+E2+E3+E4 +E5
A12 Wait on circular sling conveyor 2	N/A	R12(Circular sling conveyor 2 capacity)	E1+E2+E3+E4 +E5
A13 Pack (OP70)	Packs OK or not ? (C4)	R13(Pack resource)	E1+E2+E3+E4 +E5
A14 Ship to storage	N/A	R14(forklift truck)	E1+E2+E3+E4 +E5
A15 Store in warehouse	N/A	R15(Warehouse capacity)	E1+E2+E3+E4 +E5

SADT CM branches

SADT branches created by participant A is shown in figure 6.1.

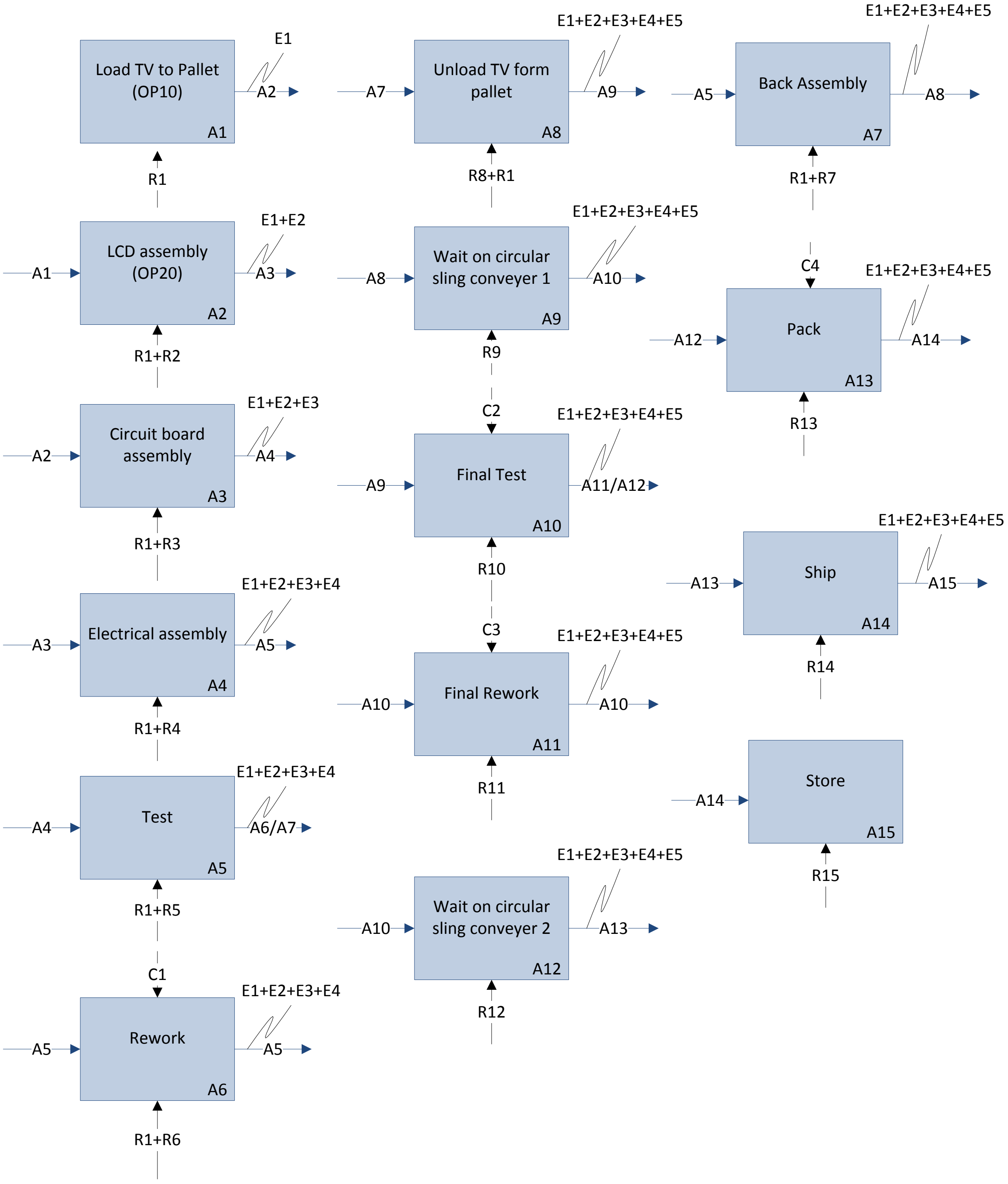


Figure 6.1: SADT CM branches (Participant A)

Complete system description

Figure 6.2 presents the complete system description generated by Participant A. This was achieved by combining all branches in figure 6.1.

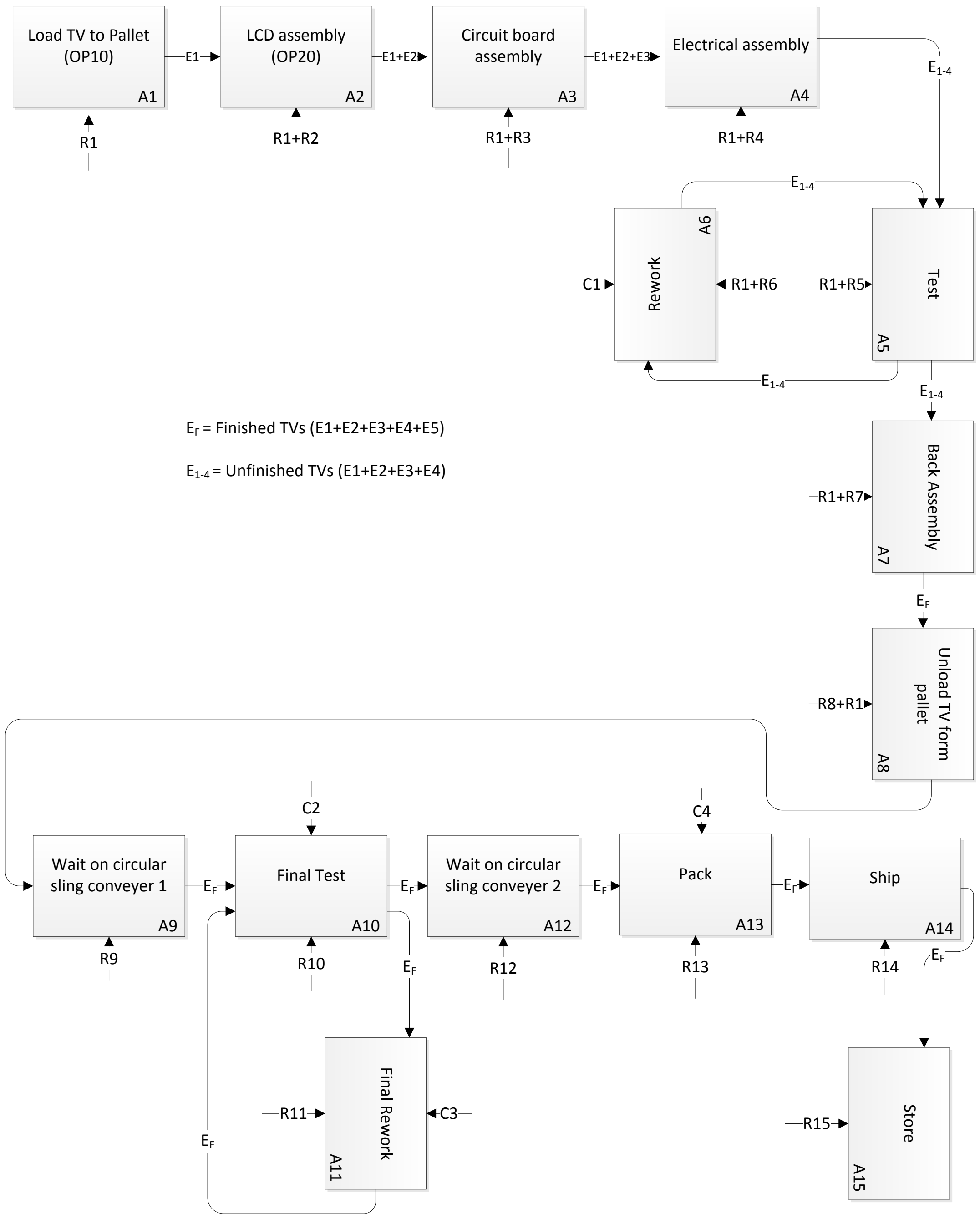


Figure 6.2: Complete system description (participant A)

Modelling Objective, experimental factors and objective activity

Modelling objective: To study the number of TVs produced daily.

Experimental factors: Pallets and storage spaces

Objective Activity (OA): Unload TV from pallet (A8). The reason provided was that the final test and packing area are often short of work and hence would not affect the daily production of the TVs.

Scope table

The abstraction phase was performed after identification of the above factors and the scope table was produced by conducting all required tests as prescribed in section 5.7.2. The resulting scope table produced by participant A is shown in table 6.7.

Abstracted diagram of the conceptual model

As per table 6.7, participant A produced the abstracted diagram of figure 6.3.

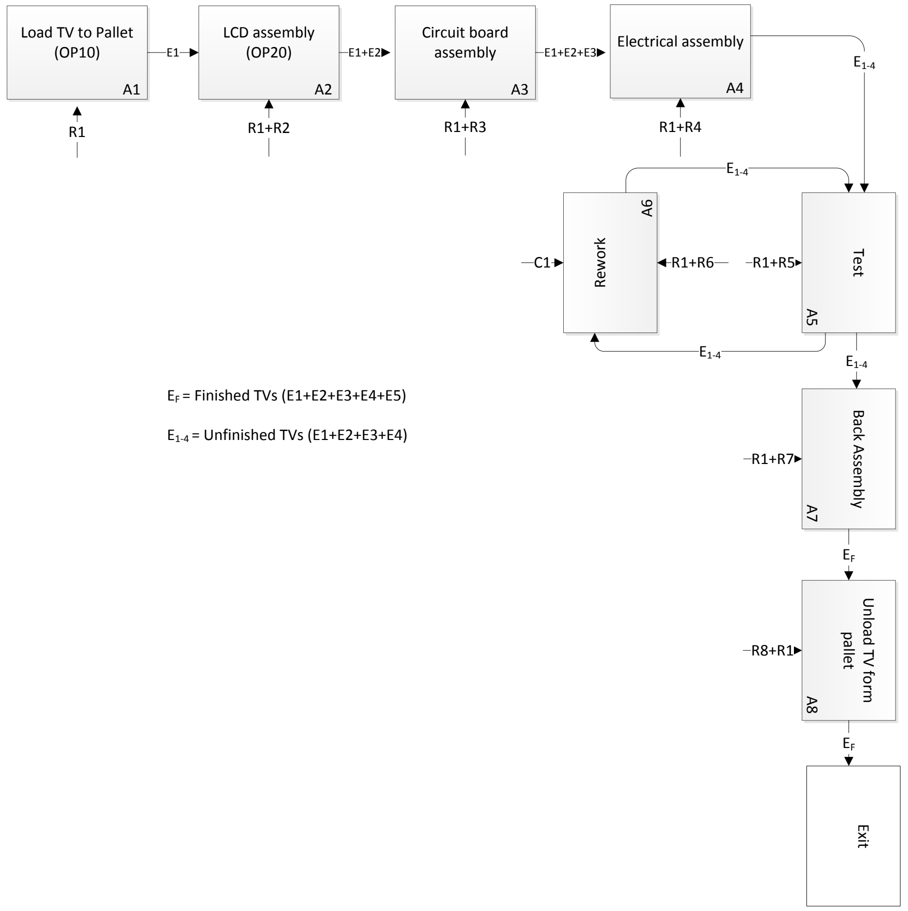


Figure 6.3: Abstracted diagram of conceptual model (participant A)

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
A1 Load TV to pallet (OP10)	N/A	R1(Pallet)	E1	Y	Y	Entry point	Y	Contributes to OA	Y	Exp. Factor	N/A	N/A	N
A2 LCD assembly (OP20)	N/A	R1 + R2(LCD assembly machine)	E1+E2	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A3 Circuit board assembly (OP30)	N/A	R1 + R3(Circuit board assembly	E1+E2+E3	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A4 Electrical assembly (OP40)	N/A	R1 + R4(Five manual operator)	E1+E2+E3 +E4	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A5 Test	N/A	R1 + R5(Test resource)	E1+E2+E3 +E4	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A6 Rework	pass / fail (C1)	R1 + R6(Rework resource)	E1+E2+E3 +E4	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	Y	Affects no. of productive	Y
A7 Back assembly (OP50)	N/A	R1 + R7(Back assembly machine)	E1+E2+E3 +E4+E5	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A8 Unloaded TV from pallet (OP60)	N/A	R1 + R8(Unloaded TV resource)	E1+E2+E3 +E4+E5	Y	Y	Affects OA entities	Y	Contributes to OA	Y	Includes Exp. Factor	N/A	N/A	Y
A9 Wait on circular sling conveyor 1	N/A	R9(Circular sling conveyor 1 capacity)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A10 Final test	Test booth available(C2	R10(Final test booth)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A11 Final rework	Pass or fail (C3)	R11(Final rework resource)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A12 Wait on circular sling conveyor 2	N/A	R12(Circular sling conveyor 2 capacity)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A13 Pack (OP70)	Packs OK or not (C4)	R13(Pack resource)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A14 Ship to storage	N/A	R14(forklift truck)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N
A15 Store in warehouse	N/A	R15(Warehouse capacity)	E1+E2+E3 +E4+E5	N	N	Activity after OA	N	Activity not modelled	N	Activity not modelled	N	Activity not modelled	N

Table 6.7: Scope Table (Participant A)

6.5.1.3 Result analysis

Participant A achieved a very close match with the final abstracted layout of the conceptual model as obtained in the reference solution. The differences were not significant. Entities were identified separately for each sub-component of the TV, but this was made clear in the SADT CM branches, entity table and scope table. Some additional resources were identified which were retained in the abstracted layout. The reason identified was not to consider those resources as dedicated resources due to the inclusion of the pallet as a resource also. This is acceptable if the experimental factor is correctly identified as this was the case. The objective activity was identified same as in the reference solution. The overall feedback was found to be highly satisfactory from the participant.

6.5.1.4 Responses

The following responses were recorded from participant A after the test:

- *What was the thing you liked most about the method?*

"The method is easy enough to apply to any system you may need to abstract their conceptual models. Thinking about the system with the unit of activities, people will think clearly and have deep understandings about how the system works."

- What you did not like about the method?

"I think the method is too messy since it is not necessary to draw all the activities one by one, especially for small systems. Before simulation designing, someone (for example me) will draw a draft flow chart to help make clear the processes flow in the simulation system. I think the flow chart can do most of the job of your activity flow does. Another problem I have in this experiential process is that I think, I prefer to have smaller and clearer systems. It will save more time for the test."

- *What suggestion would you give to improve the method?*

"The method is quite straightforward and easy to be conducted. Based on my example, I think every simulation system can be separated into different sub-blocks if there are no loops within each block. For example, the TV production line, the initial 3 activities before the first rework is actually did the same thing that is assembling something to the TV. When analyzing the system before we build the conceptual model, I cannot see any differences if I see these activities as a block."

- *Are you likely to choose the SADT CM framework for a future modelling problem?*

"Yes, if necessary. Since my current project is only about the simple queuing system which concerns more about the system performance. I think I will not use this method in my current project. I think the SADT CM framework is a useful tool for analyzing the big and complex systems."

6.5.2 Participant B

Participant B was a PhD student at the school of Business and Economics, Loughborough University. She had been teaching in operations research and supply chain management areas for about 5 years in Indonesia and a year in the UK. Her PhD research was about developing an agent-based model for the analysis of competition and collaboration behavior in supply chains. She had good familiarity with discrete-event simulation.

6.5.2.1 Test case

The test case used was Wardeon cinema problem.

6.5.2.2 Results

The key results from participant B are summarised as follows. Entity and Activity table are shown in table 6.8 and 6.9 respectively.

Entity-Table

Table 6.8: Entity table (Participant B)

Entity	Labels
Customer's call	E1 (information/inquiries, with return) E2 (information/inquiries, with no return) E3 (ticket booking, with no return) E4 (operator, with return) E5 (opertor, with no return)

Activity Table

Table 6.9: Activity Table (Participant B)

Activity	Control	Resource	Output Entity
Call coming (A1)	Customer's requirement (C1)	N/A	E1/E2/E3/E4/E5
Interacting with ACD (A2)	N/A	ACD or call router (R1)	E1/E2 E3 E4/E5
Providing basic information (A3)	N/A	Information line (4 lines - R2)	E1 E2
Ticket booking - CSR (A4)	N/A	CSR 4 lines, 3 staff - R3)	E3
Customer service - Ticket sales (A5)	N/A	Ticket sales line (4 lines-R4)	E4 E5
Call leaving or finish (A6)	N/A	N/A	N/A

SADT CM branches

SADT CM branches from participant B is shown in figure 6.4.

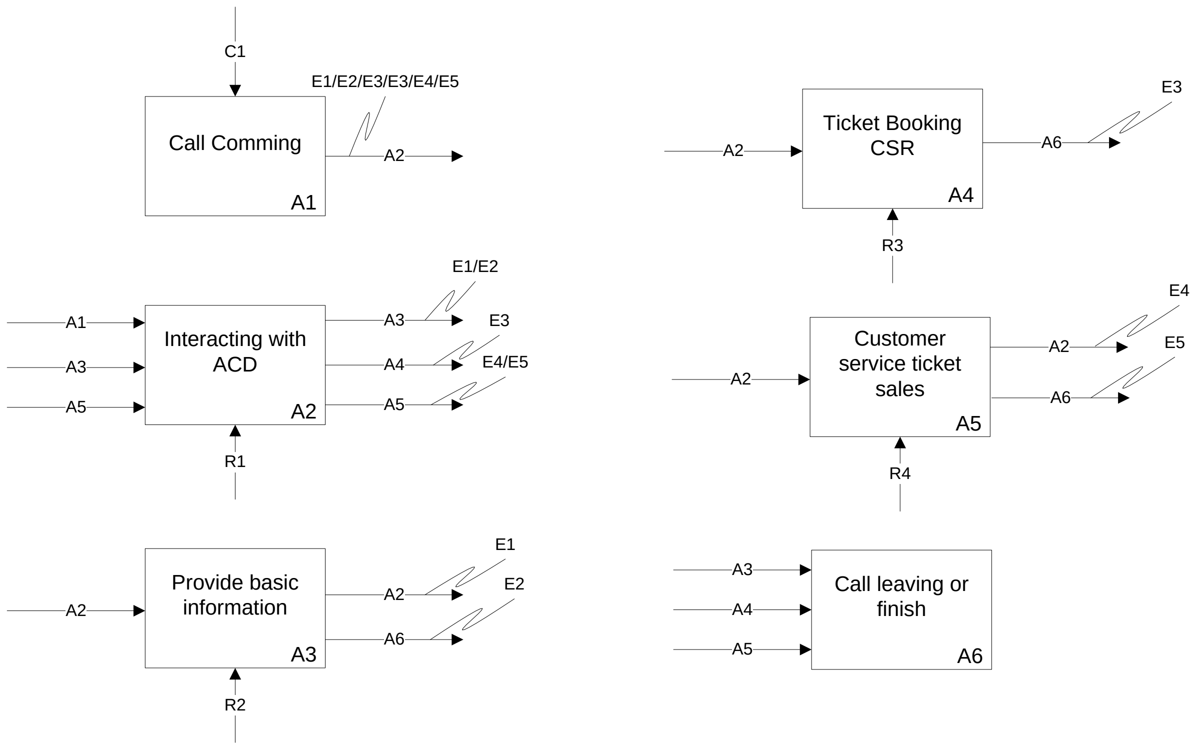


Figure 6.4: SADT CM branches (Participant B)

Complete system description

Complete system description from participant B is shown in figure 6.5

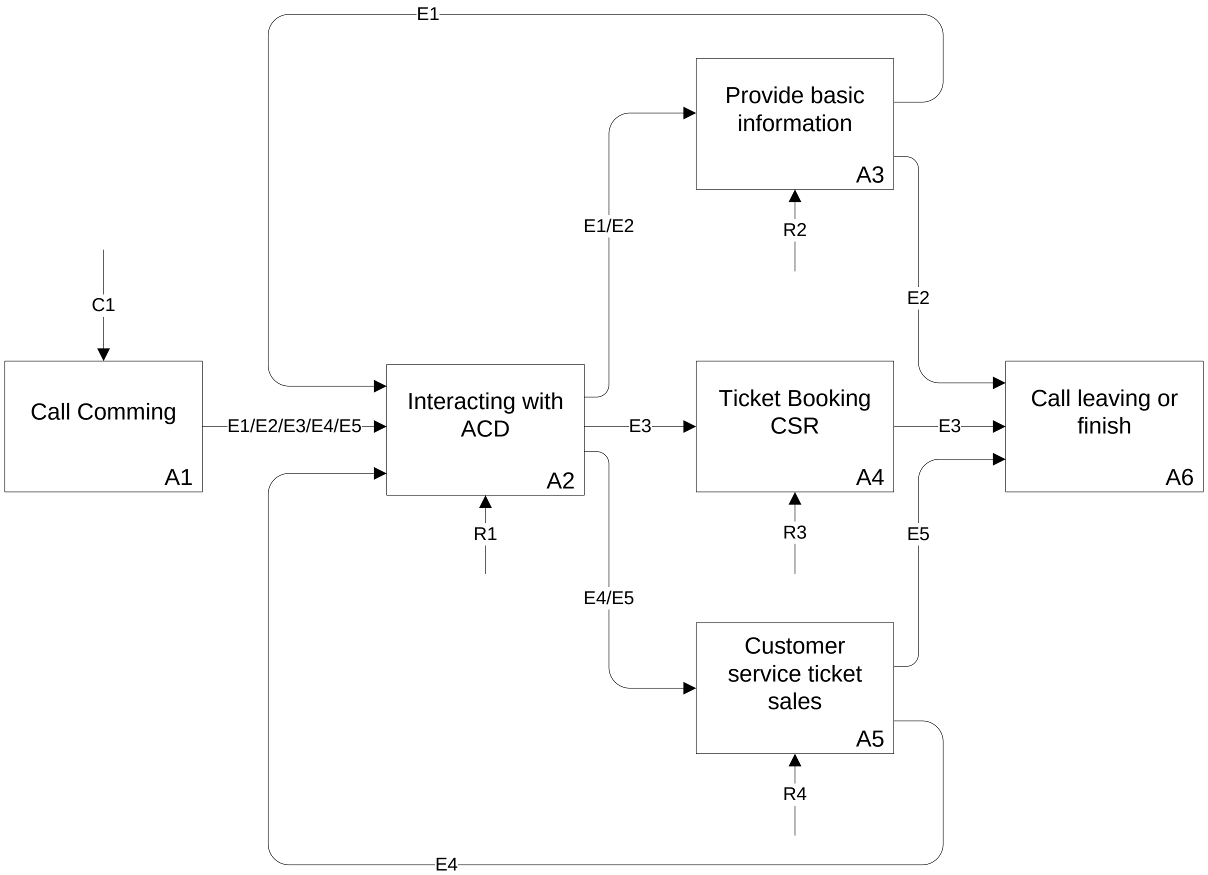


Figure 6.5: Compete system description (Participant B)

Modelling Objective, experimental factors and objective activity

Modelling objective: To examine whether the resource in the information line in the new system is sufficient to achieve the management target.

Experimental factors: Resource needed, particularly in information lines to cover both peak and off-peak condition

Objective activity: A3, as this is the main activity whose resources need to be examined.

Scope table

Scope table from participant B is shown in table 6.10

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
Call coming (A1)	Customer's requirement (C1)	N/A	E1/E2/E3/ E4/E5	yes	yes	Initialize the call process	Yes, but E3	A4, which results in E3, will be removed from the model	N/A	N/A	Yes	Affects different type of call arrivals	No (system entry point)
Interacting with ACD (A2)	N/A	ACD or call router (R1)	E1/E2 E3 E4/E5	yes	yes	Main processing unit for each call	Yes, but E3	A4, which results in E3, will be removed from the model	Yes	capacity of call router to handle the calls can affect OA	N/A	N/A	yes
Providing basic information (A3)	N/A	Information line (4 lines - R2)	E1 E2	yes	yes	Objective Activity (OA)	yes for all	Affects entities in OA	Yes	Experimental factor	N/A	N/A	yes
Ticket booking - CSR (A4)	N/A	CSR 4 lines, 3 staff - R3)	E3	No	No	It has no effect on returning service queue that influences the demand on the OA	No for E3	A4, which results in E3, will be removed from the model	No	A4, which results in E3, will be removed from the model	N/A	N/A	Yes but can be ignored since activity is not modelled
Customer service - Ticket sales (A5)	N/A	Ticket sales line (4 lines-R4)	E4 E5	yes	yes	Calls that return back can influence demand on OA	yes for all	May have an effect on entities in the OA	Yes	Capacity of ticket sales line will affect processing time of calls and hence no. of calls routing back to ACD	N/A	N/A	yes
Call leaving or finish (A6)	N/A	N/A	N/A	yes	yes	To ensure proper exit of call form the system	N/A	N/A	N/A	N/A	N/A	N/A	No (system exit point)

Table 6.10: Scope table from participant B

Abstracted layout of conceptual model

Figure 6.6 shows the abstracted layout of conceptual model.

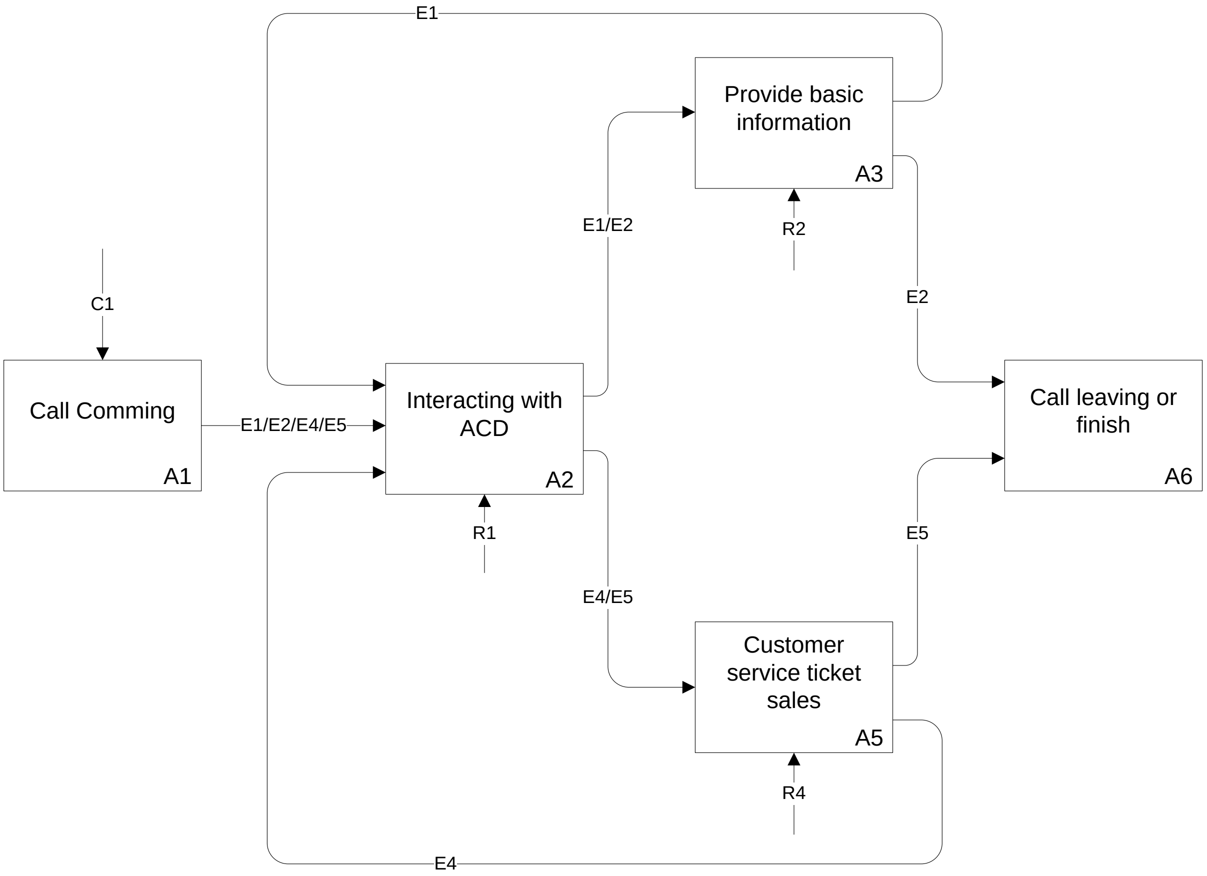


Figure 6.6: Abstracted layout of conceptual model (Participant B)

6.5.2.3 *Result analysis*

Most of the components have been correctly identified and in line with the reference solution. These include modelling objective, experimental factors, entities and the abstracted layout of the conceptual model. Five entities were identified instead of the three in the reference solution. This is because the participant considered separate entities for two return paths of the calls. This was acceptable as the entities were correctly identified in SADT CM branches, system description and abstracted layout. The participant retained the constraint and resources in the abstracted layout as she justified the impact of these components on the objective activity. This was mainly due to the resources being considered as a capacity factor of the activities rather than dedicated operators which were not identified by the participant other than the three CSR staff members. This approach is also correct since dedicated operators would eventually be removed in the abstracted conceptual model and hence would not affect the objective activity. The participant found the framework highly satisfactory with the exception of the need for automating the drawing of the diagrams.

6.5.2.4 Responses

- ***What was the thing you liked most about the method?***

“The introduced method simplifies the problem situation that is concerned with entity flow and sequential processes. In my view, the most interesting part of this approach is the visualisation part. It helps both modellers and readers to understand the problem situation much faster. It also helps modellers to recognise which activities is the core process and which activities are not relevant to the modelling objectives. However, the visualisation only provides the map of sequential processes for the entity flow, it does not show which points the queue problem exists.”

- ***What you did not like about the method?***

“Understanding how the tabulation works at the first time. However, it does help the next steps of conceptualising the model.”

- ***What suggestion would you give to improve the method?***

“The method is great, but it would be better if the method is developed into simple software.”

- ***Are you likely to choose the SADT CM framework for a future modelling problem?***

“If I have to analyse in queuing-based problem, I think I would adopt the method”

6.5.3 Participant C

Participant C was a PhD student at Loughborough University working within the special interest group of control and reliability at the Department of Aeronautical and Automotive Engineering. Her current work focuses on detailed research on police demand modelling and predictive positioning for effective service provision, a project carried out with Loughborough University and Leicestershire Police Force, sponsored by ESRC. She had good familiarity with discrete-event simulation and her background was in aeronautical engineering. She was a suitable candidate for the test as she had knowledge of both engineering and OR practices.

6.5.3.1 Test case

King’s Road Nursery problem was used for participant C.

6.5.3.2 Results

The key results from participant C are summarised below.

Entity-Table

Table 6.11: Entity table (Participant C)

Entity	Labels
Cars (Residents)	E1
Cars (Nursery)	E2
Cars (commuters)	E3

Activity Table

Table 6.12: Activity table (Participant C)

Activity	Control	Resource	Output Entity
Car enters kings road Sc(A1)	Driver decision (C1)	N/A	E1/E2/E3
Driving on Kings road Activity(A2)	N/A	N/A	E1/E2/E3
Car enters King's Meadow Activity(A3)	N/A	N/A	E1/E2
Car parks on kings road Activity(A4)	N/A	Road parking (R1)	E1/E2
Car parks on drive on King's Road Activity(A5)	N/A	Drive kings road (R2)	E1
Car parks in nursery parking Activity(A6)	N/A	Nursery parking (R3)	E2
Car parks King's Meadow on road Activity(A7)	N/A	Road parking Kings meadow (R4)	E1/E2
Car parks King's Meadow on drive Activity(A8)	N/A	Drive Kings meadow (R5)	E1
Car joins Kings road from Kings meadow Activity(A9)	N/A	N/A	E1/E2
Car queues to turn out of Kings road onto Highland	N/A	N/A	E1/E2/E3
Car queues to turn out of Kings road onto Lowland Road Activity(A11)	N/A	N/A	E1/E2/E3
Car exits road by Highland road Sk(12)	N/A	N/A	N/A
Car exits road by Lowland road Sk(13)	N/A	N/A	N/A

SADT CM branches

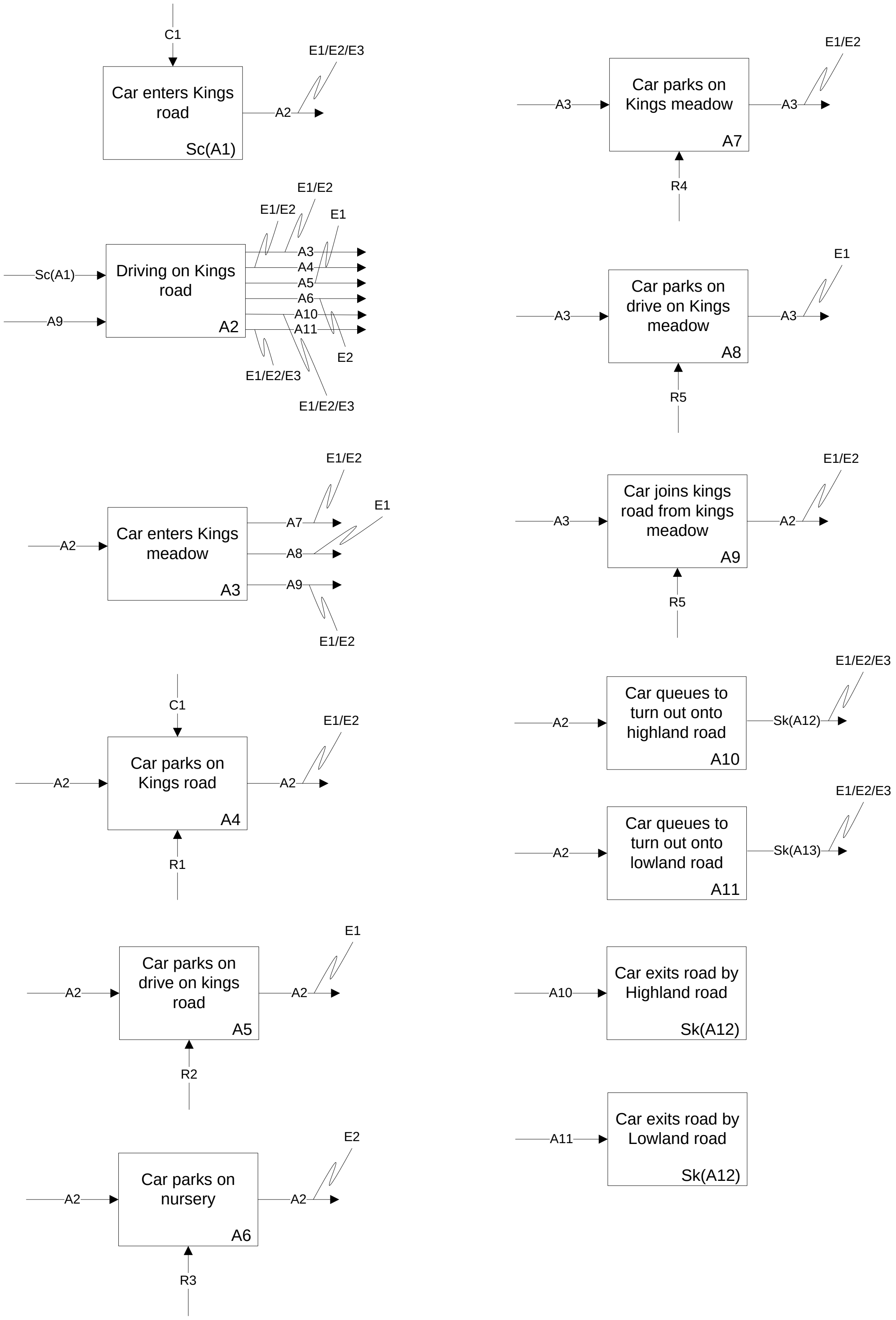
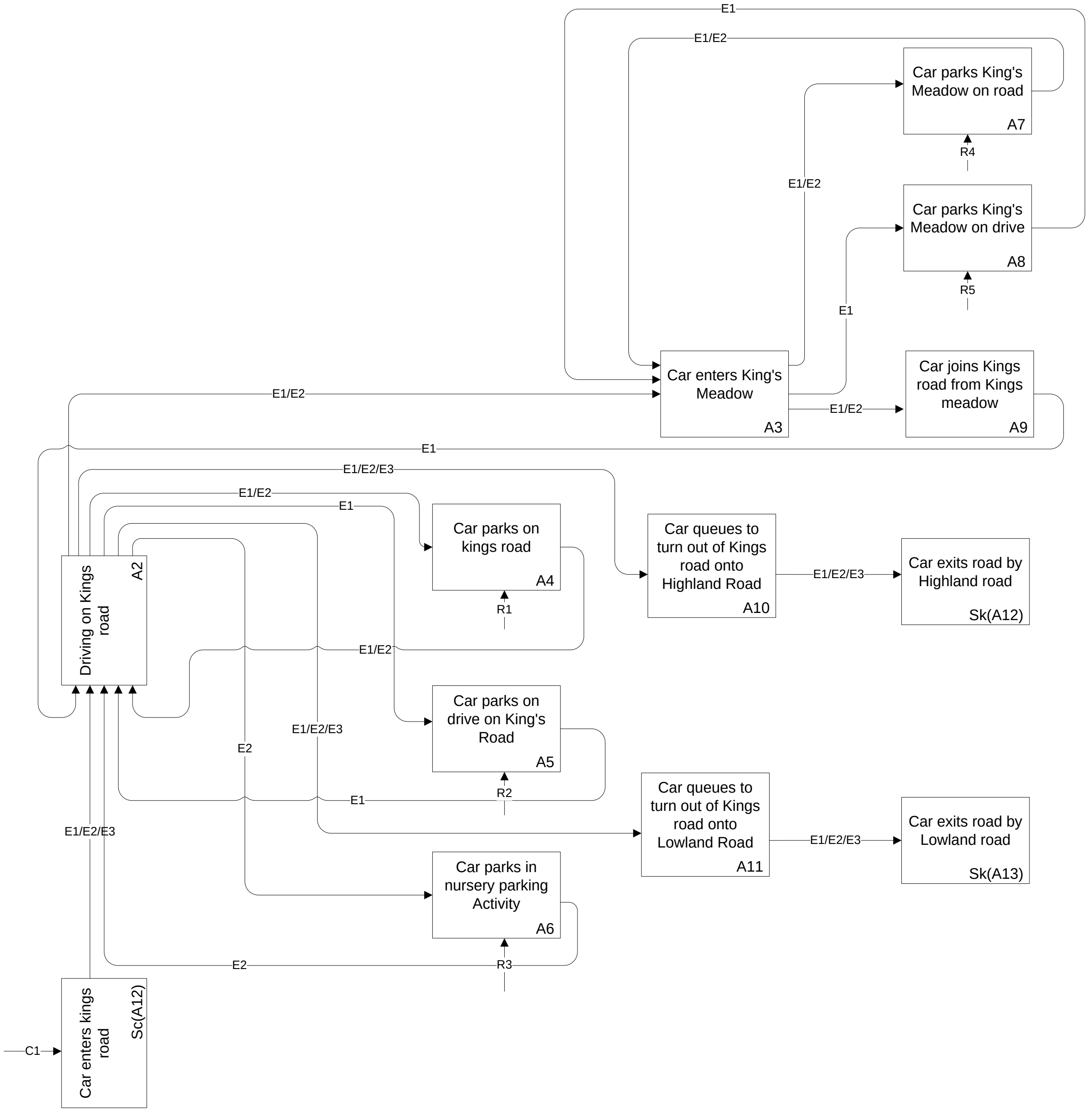


Figure 6.7: SADT CM branches (Participant C)

Complete system description



E1
E1/E2
Car parks King's Meadow on road
A7
R4
E1/E2
Car parks King's Meadow on drive
A8
R5
E1
Car enters King's Meadow
A3
Car joins Kings road from Kings meadow
A9
E1/E2
E1/E2
E1
E1/E2/E3
E1/E2
E1
Car parks on kings road
A4
R1
Car queues to turn out of Kings road onto Highland Road
A10
E1/E2/E3
Car exits road by Highland road
Sk(A12)
Driving on Kings road
A2
E1/E2
E1/E2/E3
E2
Car parks on drive on King's Road
A5
R2
E1
Car queues to turn out of Kings road onto Lowland Road
A11
E1/E2/E3
Car exits road by Lowland road
Sk(A13)
E1/E2/E3
Car parks in nursery parking Activity
A6
E2
R3
Car enters kings road
Sc(A12)
C1

Figure 6.8: Complete system description (Participant C)

Modelling Objective, experimental factors and objective activity

Modelling objective: To study the effect of traffic flow on King's Road, in particular by determining the waiting time for the queue developing to exit onto Highland Road.

Experimental factor: Presence of nursery

Objective activity: Cars turning to exit onto Highland Road

Scope table

The complete scope table form participant C is shown in table 6.13.

Abstracted layout of conceptual model

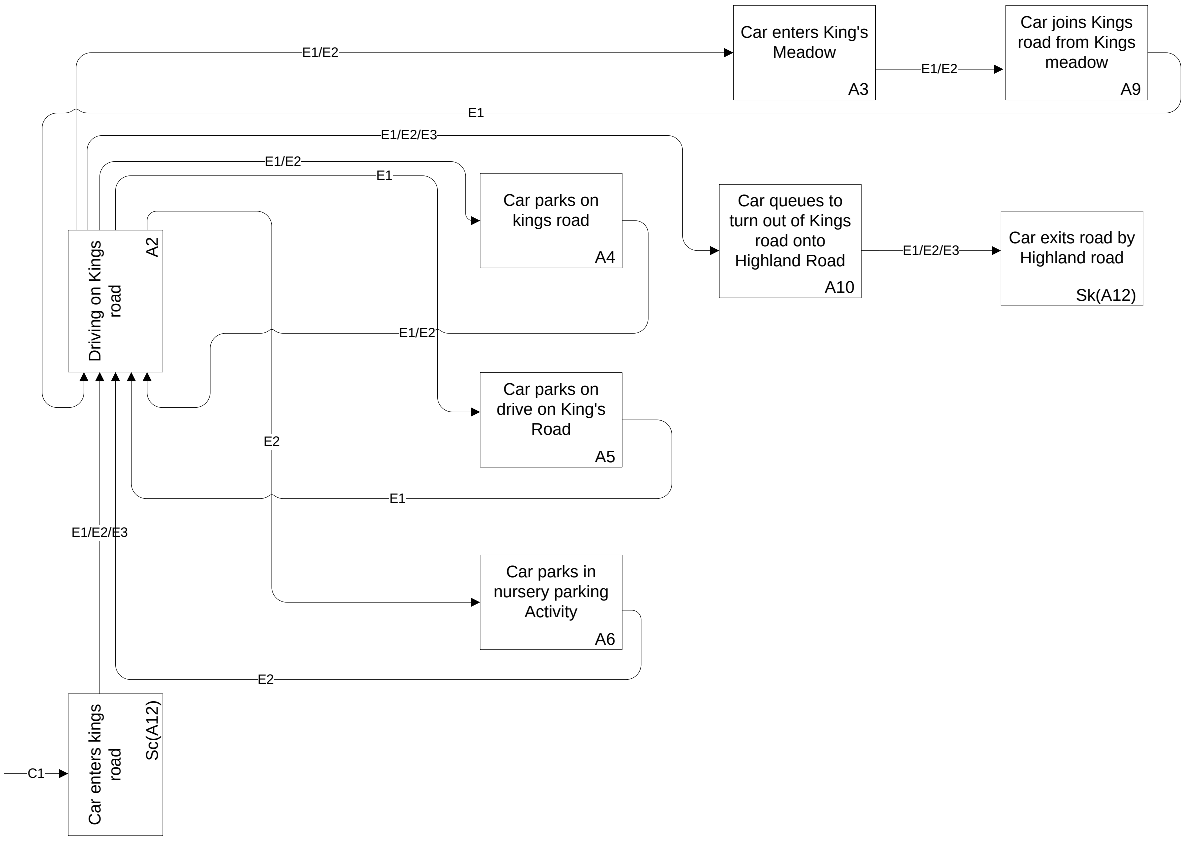

Car enters King's Meadow
A3
E1/E2
Car joins Kings road from Kings meadow
A9
E1/E2
E1
E1/E2/E3
E1/E2
E1
Car parks on kings road
A4
Car queues to turn out of Kings road onto Highland Road
A10
E1/E2/E3
Car exits road by Highland road
Sk(A12)
Driving on Kings road
A2
E1/E2
Car parks on drive on King's Road
A5
E2
E1
E1/E2/E3
Car parks in nursery parking Activity
A6
E2
C1
Car enters kings road
Sc(A12)

Figure 6.9: Abstracted layout of conceptual model

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
Car enters kings road Sc(A1)	Driver decision (C1)	N/A	E1/E2/E3	YES	YES	Starting of enteries in to Kings road	YES	All Entities make up the flow of traffic on Kings road	N/A	N/A	Yes	Customer's choice determines their direction	NO (entry point)
Driving on Kings road Activity(A2)	N/A	N/A	E1/E2/E3	YES	YES	Resposible for all traffic into OA	YES	Entities are flowing into OA	N/A	N/A	N/A	N/A	Yes
Car enters King's Meadow Activity(A3)	N/A	N/A	E1/E2	YES	YES	Affects traffic flowing back into kings road	YES	Entities are flowing into OA	N/A	N/A	N/A	N/A	YES
Car parks on kings road Activity(A4)	N/A	Road parking (R1)	E1/E2	YES	YES	Affects traffic flowing back into kings road	YES	Entities are flowing into OA	NO	Parking takes cars off road but doesn't effect queue	N/A	N/A	YES
Car parks on drive on King's Road Activity(A5)	N/A	Drive kings road (R2)	E1	YES	YES	Affects traffic flowing back into	YES	Entities are flowing into OA	NO	Parking takes cars off road but doesn't effect	N/A	N/A	YES
Car parks in nursery parking Activity(A6)	N/A	Nursery parking (R3)	E2	YES	YES	Affects traffic flowing back into	YES	Entities are flowing into OA	NO	Parking takes cars off road but doesn't effect	N/A	N/A	YES
Car parks King's Meadow on road Activity(A7)	N/A	Road parking Kings meadow (R4)	E1/E2	NO	NO	They are not on Kings road	NO	Activity not modelled	NO	Parking takes cars off road but doesn't effect queue	N/A	N/A	NO
Car parks King's Meadow on drive Activity(A8)	N/A	Drive Kings meadow (R5)	E1	NO	NO	They are not on Kings road	NO	Activity not modelled	NO	Parking takes cars off road but doesn't effect queue	N/A	N/A	NO
Car joins Kings road from Kings meadow Activity(A9)	N/A	N/A	E1/E2	YES	YES	Affects traffic flowing back into	YES	Entities are flowing into OA	N/A	N/A	N/A	N/A	YES
Car queues to turn out of Kings road onto Highland Road Activity(A10)	N/A	N/A	E1/E2/E3	YES	YES	Objective Activity (OA)	YES	Entities are flowing into OA	N/A	N/A	N/A	N/A	YES
Car queues to turn out of Kings road onto Lowland Road Activity(A11)	N/A	N/A	E1/E2/E3	NO	NO	They are queuing in the other direction	NO	Does not affect entities in OA	N/A	N/A	N/A	N/A	NO
Car exits road by Highland road Sk(12)	N/A	N/A	N/A	Yes	Yes	May affect the entities in OA	NO	Does not affect entities in OA	N/A	N/A	N/A	N/A	NO (Exit point)
Car exits road by Lowland road Sk(13)	N/A	N/A	N/A	NO	NO	They are leaving via a different exit	NO	Does not affect entities in OA	N/A	N/A	N/A	N/A	NO

Table 6.13: Scope table (Participant C)

6.5.3.3 Result analysis

In comparison with the reference solution, the modelling objective, experimental factors and objective activity were identified correctly. Three entities were identified and all SADT CM branches and the system description were drawn correctly. One difference was in the identification of the resources. The participant identified all car parks as resources within the parking activities. Since the approach allowed these resources to be identified in all the diagrammatic parts of the framework; this was equally acceptable and could provide an additional dimension to the experimental factors in the model. Also this could increase the accuracy of the model if detailed analysis for the car parking activity was to be considered.

6.5.3.4 Responses

Following were the key responses recorded from participant C.

- ***What was the thing you liked most about the method?***

“The method divided the problem into very simple steps such as deciding the entities and then built on this. This was useful to make the problem more accessible and help you make decisions in a clear order.”

- ***What you did not like about the method?***

“The stage of drawing all the boxes out individually and then putting them together into a diagram was very time consuming.”

- **What suggestion would you give to improve the method?**

“It will be helpful to have the process of drawing the boxes and then putting the diagram together more automated.”

- **Are you likely to choose the SADT CM framework for a future modelling problem?**

“Yes I will use the method to structure future problems”

6.6 Summary

This chapter covered the detailed testing of the SADT CM framework presented in chapter 5. Overall, two pilot tests and three final tests were conducted to evaluate the approach. After these tests, it was conclusive that SADT CM has clear potential for conceptual modelling in discrete event simulation problems. In contrast to the PREmod framework (chapter 3), SADT CM framework benefits from the diagrammatic approach during the generation of the system description and abstraction phases in conceptual modelling. The participants of the tests gave satisfactory responses for most parts of the framework. The main limitation of the approach as identified during the testing was the manual construction of the system description from SADT CM branches. However, the participants still find it useful to draw the branches and labelling the details for structuring and breaking down the problem in visual form. This helped them in quickly abstracting the required conceptual model by working through the diagrammatic layout of the system description as every component of the model was clearly visible. It was also conclusive that different conceptual model developed for the same problem can be easily represented by the framework as the approach allows flexibility for structuring information in both diagrammatic and tabular formats.

Most of the components of the participants' solution were found to be similar to those of the reference solutions. The similarities among the solutions of the participants were found within the identification of entities, activities and modelling objectives. The differences were mainly in the identification of the resources and constraints. Resources were identified differently, since either they were related to particular activities or as separate experimental factors. In some cases, specific activities (such as parking spaces) were identified as separate resource to be modelled with the parking activity. Also, at other places, some entities in the reference solution were identified by the participants as a resource (e.g. pallets in the case of Panorama TV problem), since they were identified as an experimental factor and retained in the abstracted solution. All these differences do not create significant deviation from the reference solution. Moreover, it is shown that despite these differences, the framework is capable of accommodating different versions, while abstracting an appropriate conceptual model in terms of the modelling objectives.

The key feedback given from the participants reflects on the ease, flexibility and structured approach of the framework. Mostly, they found the framework easy to follow and apply along with the flexibility of developing templates in Excel at different stages of the framework. The idea of focusing on the objective activity kept participants' concentration around the root cause of modelling the whole problem, alongside providing options to visually structure the problem by creating SADT CM branches. This combination of the visual and structural approach was appreciated by the participants. Although one of the major concern was the lack of a computer tool to develop and combine the SADT CM branches, still participants found it helpful to draw the SADT CM branches manually as it helped them to breakdown and structure the problem.

The next chapter provides some discussion of the research work presented in this thesis and identify areas of the future direction.

7 Discussion and future work

This chapter summarises the research presented in this thesis along with a brief discussion about the research outcomes, limitations of the study and identification of future directions for the research.

7.1 Summary of the research

The research focuses on the adaptation of CS methods into M&S CM, specifically discrete-event simulation. As discussed in chapter 2, previously little research exists that explores such adaptation of CS methods. This adaptation of CS methods into M&S CM is supported by a comprehensive cross-disciplinary review of CM between M&S and CS, presented in chapter 2. It is identified that SRE methods are closest in nature to M&S CM and hence it is the most suitable candidate area for adaptation. It is also identified that among four key components of a conceptual model (definition, purpose, process and testing), the process of CM (or framework to develop a conceptual model) needs the most urgent attention. Hence, the adaptation of SRE methods into M&S CM framework should focus on the process/framework for CM. The testing component also needs significant attention and this is highlighted as future work in section 7.4. Meanwhile, perspectives on the other two components (definition and purpose) have some level of consensus in the M&S community and approaches are already available for research community.

Chapter 4 proposes a CM framework called PREmod based on SRE method, PREview, (Sommerville and Sawyer, 1997). PREmod is shown to have some potential for generating a system description using view point orientation, however, it lacks severely in dealing with the abstraction part due to its subjective nature. The absence of a diagrammatic layout in PREmod also makes it less useful for the abstraction phase, as modellers can get lost in the procedure of view point removals in the absence of any diagrammatic layout and lack of structural support.

Chapter 5 then proposes another CM framework called SADT CM, based on SRE method SADT (Ross, 1977). SADT CM is shown to overcome the limitations of PREmod and proves to be useful both in the system description and abstraction phase. The diagrammatic and visual approach of creating SADT CM branches and combining them to get a full system description is shown to help participants greatly in understanding the problem description.

Chapter 6 reports the detailed testing of SADT CM. The tests demonstrate that SADT CM is useable both in the pilot and final testing. The responses of the participants recorded from these test experiments also leads to a clear conclusion about the framework being suitable for discrete-event simulation problems.

7.2 Research outcomes

The overall aim of the thesis is stated in chapter 1 as:

"To explore how CM approaches in CS could be adopted and adapted for M&S CM"

This aim is achieved by devising and testing a M&S CM framework based on CS methods. We now briefly discuss each research question of this thesis as presented in section 1.3 and provide comments about how this thesis addresses them.

- ***RQ1: Which part of CS is most appropriate for adaptation to M&S CM?***

As discussed in section 3.1, this research question informs the direction of this research, where out of the CS domain, SE is identified to be the most closely aligned with that of CM within M&S. This analysis is presented in chapter 2. Furthermore, the development of the actual frameworks for CM based on SE methods confirms their suitability for CM processes within M&S. Although these frameworks focus on the process of CM, it is anticipated that by using similar adaptation from CS methods, testing of CM within M&S could also be done in a more rigorous and structured manner.

- ***RQ2: Which methods from CS could best be adopted for M&S CM?***

The two proposed SRE methods in chapter 3 used for M&S CM in this thesis are PREView and SADT. Their CM derivatives are called PREmod and SADT CM respectively. Both frameworks are developed and tested for suitability in M&S CM as explained in chapters 4, 5 and 6. It is shown in this thesis that both PREview and SADT have potential to be used in M&S CM. Although PREmod did not qualify with the overall aims of the thesis, because it provided little support for abstraction, SADT CM provided satisfactory results with respect to feasibility and usability testing. It is shown that SADT CM has potential for use in discrete-event simulation in order to structure a problem by defining all its components (entities, resources and controls) at a lower-level of modelling. This helps to identify individual interactions amongst these components along with any resource or constraint requirement. It has been shown through usability testing of the SADT CM that the essential criteria discussed in section 3.3 have been successfully evaluated. This is confirmed through participants' comments about the understanding of the SADT CM framework and through application of the framework on the given modelling problem. Both key properties of the DES CM framework (system description and abstraction of conceptual model) have been explicitly evaluated in the test experiment. Moreover, the results also confirmed that a good level of understanding was developed to structure the physical components of a DES problem along with comprehensive generation of tabular information regarding the interaction between different components was also achieved. Lastly, the idea to drive the framework with focus on the modelling objective was seen helpful by the participants so as to arrive at valid conceptual models.

The generic notations and communicative style of SADT allows this information to be integrated into a complete system description (SD). The next phase of CM, the abstraction of the conceptual model from the defined SD, is also shown to be feasible using this approach. It is demonstrated that the structured SD defined using the SADT approach, allows an easier abstraction and design of simulation models. In addition, CM validation might be carried out in a more efficient and transparent way using SADT CM. Direct comparison of the SADT SD diagrams with the SADT conceptual model diagrams would enable the abstractions in the model to be identified, debated and validated with subject matter experts.

- ***RQ3: How should these methods be adapted for M&S CM***

This research question is successfully answered through the outcome of the evaluation of both frameworks. As per the research methodology described in chapter 3, two parts of the evaluation are conducted. The outcome of the first part is the reference solution developed by the author using PREmod and SADT CM. These solutions are documented in section 4.3 and appendix 3. The reference solutions were generated iteratively along with the development of the framework as explained in section 3.3. The outcome of the second part is the results and responses produced by the participants using these frameworks during pilot testing and main testing. These results and responses are documented in chapter 4 (PREmod) and chapter 6 (SADT CM). Since results from pilot testing for PREmod were not satisfactory in terms of usability of the framework, the final test was only conducted with SADT CM.

The test experiment on SADT CM confirms the usability of the framework and also provides insight into how different levels of modellers would use this approach on different test cases.

Although test participants' responses did not indicate that they would likely be adopting the whole framework as it is devised, there was an agreement that more than half of the SADT CM framework (whole of SD and initial 2-3 steps of the abstraction part that mainly deals with the objective activity) could be easily utilised in structuring the modelling problem. One of the key responses was about the structured framework that gave participants a new way of structuring a discrete-event simulation problem without going into building the computer model. The test experiment also helped evaluate the generality of the framework as the test was conducted using cases from different domains, and with participants having different expertise and backgrounds in discrete-event simulation modelling.

7.3 Contribution of the thesis

The main contribution of this thesis is as follows "*Development of a CM framework through adaptation of a well-known SRE method called Structured Analysis and Design Technique (SADT)"*

In addition to this, a secondary contribution is to highlight the existing practices of CM in CS which also identifies the dearth of standards within M&S CM on a comparative basis. This contribution is supported through a comprehensive review of CM within M&S and CS across four components of a conceptual model: definition, purpose, process and testing. There are few studies which report the possibility of bringing CS methods into M&S CM (e.g. Arthur and Nance , 2007), but none of these researchers make a detailed comparison of the similarities and differences between the fields of M&S and CS. The review presented in this thesis helps to identify the areas in both disciplines where any such integration might be appropriate.

The main contribution is achieved through development of the framework for M&S CM based on SADT method from SRE. There is little evidence of the adoption of CS methods by M&S despite the potential benefits. Various authors (Turnitsa et al., 2010; Tolk and Turnitsa, 2012; Guizzardi and Wagner, 2012; Tolk et al., 2013; Arthur and Nance, 2007) have discussed the possibility of such integration, however, their research has either focused on typical CS theories that require some level of subject matter expertise, or describe general guidelines rather than a specific useable framework. This thesis contributes in this area through the development of a structured framework for M&S CM influenced from the software engineering method (so as to gain similar benefits as in the latter filed) and provides step-by-step guidance for modellers to follow this approach. A comprehensive user guide (appendix 1) is also developed as part of this research to support users of the framework on M&S CM.

7.4 Limitations of the study and future work

There are some limitations associated with this research work. Firstly, the research has focused on a single component of CM; namely the process of developing a conceptual model. It is identified in this research that CM quality testing procedures also requires attention from the research community. This is an area for future research. There are various possibilities to handle this issue. The first approach could be to develop a precise, semi-formal framework that can be used by modellers to evaluate their conceptual models as CM progresses. This view is towards a process view of the quality as stated in Wand and Weber (1990). At each stage the modeller would be verifying different dimensions of the conceptual model from the real world. The second approach could be a product-oriented approach as stated in Lindland et al. (1994). The conceptual model could be verified against several *goals* and *means* to achieve semantic, syntactic and pragmatic qualities.

Secondly, the methodology of review in this thesis has taken four components of a conceptual model; namely, definition, purpose, process and testing. These findings could be extended by using a framework based on meta-models (upper ontologies) for each CM domain. This would enable a broad comparison of the different fields so that more components of CM (relations, types, attributes, properties etc.) across all domains can be compared. Such an approach could bring greater precision in identifying the potential areas of improvements for CM in M&S. Moreover, a future work could be to explore the other two main domains of M&S namely, Agent-based simulation and system dynamics for any such adaptation of methods from CS.

One of the limitations of SADT CM approach is handling the systems with a large number of activities and resources where the potential combinations of individual models can prove to be cumbersome. Addressing this would require the use of an automated support tool where all the individual activities could be fed into it and the SD could be generated automatically. This could be achieved by writing a computer program in an object-oriented language (for example, JAVA or C++), using activities as objects under different classes (for example, main line, testing and packaging class). A user input would be required to indicate different parameters for each object (entity, constraint, resource and control). In the final step, using the graphical library of the language, different objects (activities) under different classes could be combined to produce a complete SD in a diagrammatic form. However, in the absence of such a tool, the manual approach of SADT CM would still be useful because the individual models can be created manually without having a complete knowledge of the system.

Finally, it would be useful to perform additional testing of the SADT CM framework with experienced modellers from industry or consultancy firms to get an insight into how this approach compares with their usual CM practices.

References

Ahmed, F., Robinson, S. & Tako, A. 2014. CONCEPTUAL MODELLING: LESSONS FROM COMPUTER SCIENCE. *In: B.Tjahjono, C. H., S. Onggo, and D-J. van der Zee* (ed.) *7th Simulation Workshop Conference (SW14).* The Abbey Hotel Golf and Country Club, Worcestershire,England B98 9BE. UK.

Andrade, J., Ares, J., Garcia, R., Pazos, J., Rodriguez, S. & Silva, A. 2004. A methodological framework for viewpoint-oriented conceptual modeling. *IEEE Transactions on Software Engineering,* 30**,** 282-294.

Andrade, J., Ares, J., GarcÃ-a, R., Pazos, J., RodrÃ-guez, S. & Silva, A. 2006. Definition of a problem-sensitive conceptual modelling language: foundations and application to software engineering. *Information and Software Technology,* 48**,** 517-531.

Arbez, G. & Birta, L. G. 2011. The ABC Conceptual Modelling Framework. *In:* Robinson, S., Brooks, R., Kotiadis, K. & Van der zee, D.-J. (eds.) *Conceptual modeling for Discrete Event Simulation.* Boca Raton London Newyork: CRC Press.

Ares, J. & Pazos, J. 1998. Conceptual modelling: an essential pillar for quality software development. *Knowledge-Based Systems,* 11**,** 87-104.

Arthur, J. D. & Nance, R. E. 2007. Investigating the use of software requirements engineering techniques in simulation modelling. *Journal of Simulation,* 1**,** 159-174.

Balci, O. 1994. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research,* 53**,** 121-173.

Balci, O., Arthur, J. D. & Nance, R. E. 2008. Accomplishing reuse with a simulation conceptual model. *Proceedings of the 40th Conference on Winter Simulation.* Miami, Florida: Winter Simulation Conference.

Balci, O. & Nance, R. E. 1985. Formulated problem verification as an explicit requirement of model credibility. *Simulation,* 45**,** 76-86.

Balci, O. & Ormsby, W. F. 2007. Conceptual modelling for designing large-scale simulations. *J of Sim,* 1**,** 175-186.

Basili, V. R. 1992. Software modeling and measurement: the Goal/Question/Metric paradigm. Techreport UMIACS TR-92-96, University of Maryland at College Park, College Park, MD, USA.

Batra, D. & Davis, J. G. 1992. Conceptual data modelling in database design: similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies,* 37**,** 83-101.

Battista, G. D., Kangassalo, H. & Tamassia, R. 1989. Definition libraries for conceptual modelling. *Data & Knowledge Engineering,* 4**,** 245-260.

Batty, M., Desyllas, J. & Duxbury, E. 2003. “The Discrete Dynamics of Small-Scale Spatial Events: Agent-based Models of Mobility in Carnivals and Street Parades”. *International Journal of Geographical Information Science*, 17**,** 673-697

Beck, K. 1999. *Extreme Programming Explained : Embrace Change, XP Series,* Reading, MA, Addison-Wesley.

Boger, M., Baier, T., Wienberg, F. & Lamersdorf, W. Extreme modeling. eXtreme Programming and Flexible Processes in Software Engineering, June (2000) 2000.

Borshev, A. & Filippov, A. 2004. “From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools”. *Proceedings of the 22nd International Conference of the System Dynamics Society*

Buchanan, B. G., Barstow, D., Bechtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T. & Waterman, D. A. 1983. Constructing an expert system. *In:* Hayes-Roth, F., Waterman, D. A. & Lenat, D. B. (eds.) *Building Expert Systems.* USA: Addison-Wesley, Reading.

Bunge, M. 1977. Ontology I: The Furniture of the World (vol. 3). Dordrecht: Reidel.

Cetinkaya, D., Verbraeck, A. & Seck, M. 2009. “Towards A Component Based Conceptual Modeling Language for Discrete Event Simulation”. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.403.8115 (accessed June 2015)

Checkland, P. 1995. Model validation in soft systems practice. *Systems Research,* 12**,** 47-54.

Chen, P. P.-S. 1976. The entity-relationship model:toward a unified view of data. *ACM Trans.Database Syst.,* 1**,** 9-36.

Chesterton, G. K. 1935. *The point of the pin, The Scandal of Father brown Cassell,* London.

Chwif, L., Banks, J., de Moura Filho, J. P. & Santini, B. 2013. A framework for specifying a discrete-event simulation conceptual model. *J of Sim,* 7**,** 50-60.

Congram, C. & Epelman, M. 1995. “How to Describe Your Service: An Invitation to The Structured Analysis and Design Technique”. *International Journal of Service Industry Management,* 6**,** 6-23

Crockett, H. D., Guynes, J. & Slinkman, C. W. 1991. Framework for development of conceptual data modelling techniques. *Information and Software Technology,* 33**,** 134-142.

Datri, A., De Marco, M. & Casalino, N. 2008. *Interdisciplinary Aspects of Information Systems Studies: The Italian Association for Information Systems,* Springer Science & Business Media

Davies, I., Green, P., Rosemann, M., Indulska, M. & Gallo, S. 2006. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering,* 58**,** 358-380.

Delugach, H. S. 1992. Specifying multiple-viewed software requirements with conceptual graphs. *Journal of Systems and Software,* 19**,** 207-224.

Demraoui, L., Behja, H., Zemmouri, E. M. & Benabbou, R. Towards an approach to integration of the viewpoint concept into the CWM. Next Generation Networks and Services (NGNS), 2014 Fifth International Conference on, 28-30 May 2014 2014. 352-355.

Dickover, M. E., McGowan, C. L. & Ross, D. T. 1977. Software design using: SADT. *Proceedings of the 1977 annual conference.* ACM.

Dromey, R. G. 1995. A model for software product quality. *Software Engineering, IEEE Transactions on,* 21**,** 146-162.

Engels, G., Gogolla, M., Hohenstein, U., Hulsmann, K., Lohr-Richter, P., Saake, G. & Ehrich, H.-D. 1992. Conceptual modelling of database applications using an extended ER model. *Data & Knowledge Engineering,* 9**,** 157-204.

Evermann, J. & Wand, Y. 2005. Ontology based object-oriented domain modelling: fundamental concepts. *Requirements Engineering,* 10**,** 146-160.

Gemino, A. & Wand, Y. 2004. A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering,* 9**,** 248-260.

Gemino, A. & Wand, Y. 2005. Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering,* 55**,** 301-326.

Genesereth, M. & Nilsson, N. 1986. *Logical Foundations of Artificial Intelligence,* California, Morgan Kaufmann Publishers.

Goamez, A., Juristo, N., Montes, C. & Pazos, J. 1997. La Ingenieraa del Conocimiento. CEURA, Madrid.

Greenspan,S.J.,Mylopoulus, J. & Borgida, A. 1982. Capturing more world knowledge in the requirements specification. Proceedings of the 6th international conference on Software engineering. Tokyo, Japan: IEEE Computer Society Press.

Guizzardi, G. & Wagner, G. 2005. Towards ontological foundations for agent modelling concepts using the unified fundational ontology (UFO). *Agent-Oriented Information Systems Ii,* 3508**,** 110-124.

Guizzardi, G. & Wagner, G. 2012. Conceptual simulation modeling with onto-UML. *Proceedings of the Winter Simulation Conference.* Berlin, Germany: Winter Simulation Conference.

Heavey, C. & Ryan, J. Process Modelling Support for the Conceptual Modelling Phase of a Simulation Project. Simulation Conference, 2006. WSC 06. Proceedings of the Winter, 3-6 Dec. 2006 2006. 801-808.

Hoppenbrouwers, J., van der Vos, B. & Hoppenbrouwers, S. 1997. NL structures and conceptual modelling: grammalizing for KISS. *Data Knowl.Eng.,* 23**,** 79-92.

Innis, G. & Rexstad, E. 1983. Simulation model simplification techniques. *SIMULATION,* 41**,** 7-15.

Insfran, E., Pastor, O. & Wieringa, R. 2002. Requirements Engineering-Based Conceptual Modelling. *Requirements Engineering,* 7**,** 61-72.

Insfran, E., Pelechano, V. & Pastor, O. 2002. Conceptual Modeling in the eXtreme. *Information and Software Technology,* 44**,** 659-669.

Jeong, K.-Y. 2000. “Conceptual Frame for Development of Optimized Simulation-based Scheduling Systems”. *Expert Systems with Applications*, 18, 299-306

Karagoz, N. A. & Demirors, O. 2011. Conceptual Modelling Notations and Techniques. *In:* Robinson, S., Brooks, R., Kotiadis, K. & Van der zee, D.-J. (eds.) *Conceptual modeling for Discrete Event Simulation.* Boca Raton London Newyork: CRC Press.

Kellner, M. I., Madachy, R. J. & Raffo, D. M. 1999. Software process simulation modeling: Why? What? How? *Journal of Systems and Software,* 46**,** 91-105.

Kim, C.-H., Weston, R. H., Hodgson, A. & Lee, K.-H. 2003. “The Complementary use of IDEF and UML Modelling Approaches”. *Computers in Industry,* 50**,** 35-56

Kleijnen, J. P. C. 1995. Verification and validation of simulation models. *European Journal of Operational Research,* 82**,** 145-162.

Kotiadis, K. 2007. Using soft systems methodology to determine the simulation study objectives. *Journal of Simulation,* 1**,** 215-222.

Kotiadis, K. & Robinson, S. 2008. Conceptual modelling: knowledge acquisition and model abstraction. *Proceedings of the 40th Conference on Winter Simulation.* Miami, Florida: Winter Simulation Conference.

Kotiadis, K., Tako, A. A. & Vasilakis, C. 2013. A participative and facilitative conceptual modelling framework for discrete event simulation studies in healthcare. *J Oper Res Soc*.

Kotonya, G. 1999. Practical Experience with Viewpoint-Oriented Requirements Specification. Springer London.

Kotonya, G. & Sommerville, I. 1996. Requirements engineering with viewpoints. *Software Engineering Journal,* 11**,** 5-18.

Kung, C. H. 1989. Conceptual Modeling in the Context of Development. *Software Engineering, IEEE Transactions on,* 15**,** 1176.

Kung, C. H. & Solvberg, A. Activity modeling and behavior modeling. Proc. of the IFIP WG 8.1 working conference on Information systems design methodologies: improving the practice, 1986 1986 Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co, 145-171.

Landry, M., Malouin, J.-L. & Oral, M. 1983. Model validation in operations research. *European Journal of Operational Research,* 14**,** 207-220.

Law, A. M. 1991. Simulation model's level of detail determines effectiveness. *Industrial Engineering,* 23**,** 16-18.

Law, A. M. 2007. *Simulation modeling and analysis,* Boston;London, McGraw-Hill.

Lehaney, B. & Paul, R. J. 1996. The use of soft systems methodology in the development of a simulation of out-patient services at watford genereal hospital. - 864.

Lglesias, C. A., GARIJO, M. & GONZÁLEZ, J. C. 1999. A survey of agent-oriented methodologies. Intelligent Agents V: Agents Theories, Architectures, and Languages. Springer

Lindland, O. I., Sindre, G. & Solvberg, A. 1994. UNDERSTANDING QUALITY IN CONCEPTUAL MODELING. *Ieee Software,* 11**,** 42-49.

Lucas P., Hommerson A., Galan J.C., Marcos M., Coltell O., Mouzon O., Polo C., Rosenbrand K., Wittenberg J., van Croonenborg J. 2005. "New model of guideline process". Specific Targeted Research Project Information Society Technology, Universitat Jaume I, IST-FP6-508794

March, S. T. & Allen, G. N. 2009. Challenges in Requirements Engineering: A Research Agenda for Conceptual Modeling. *Design Requirements Engineering: A Ten-Year Perspective,* 14**,** 157-165.

Marin, B., Giachetti, G., Pastor, O. & Abran, A. 2010. A Quality Model for Conceptual Models of MDD Environments.

Menzel, C. & Mayer, R. J. 1998. The IDEF family of languages. *Handbook on architectures of information systems.* Springer.

Mineau, G. W., Missaoui, R. & Godinx, R. 2000. Conceptual modeling for data and knowledge management. *Data & Knowledge Engineering,* 33**,** 137-168.

Mohagheghi, P., Dehlen, V. & Neple, T. 2009. Definitions and approaches to model quality in model-based software development â€" A review of literature. *Quality of UML Models,* 51**,** 1646-1669.

Montevechi, J., Da Silva Costa, R. F., Leal, F., DE Pinho, A. F., Marins, F., Marins, F. F. & De Jesus, J. T. 2008 . "Combined Use of Modeling Techniques for the Development of the Conceptual Model in Simulation Projects". *Proceedings of the 2008 Winter Simulation Conference* .S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, eds. 987-995, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Montevechi, J., Leal, F., De Pinho, A. F., Da Silva Costa, R. F., De Oliveira, M. L. M. & Da Silva, A. L. F. 2010. "Conceptual Modeling in Simulation Projects by Mean Adapted IDEF: An Application in a Brazilian Tech Company". *Proceedings of the 2010 Winter Simulation Conference*. B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan,eds. 1624-1635, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc

Montevechi, J. A. B. & Friend, J. D. 2012. Using a soft systems methodology framework to guide the conceptual modeling process in discrete event simulation. *Proceedings of the Winter Simulation Conference.* Berlin, Germany: Winter Simulation Conference.

Moody, D. L. 2005. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering,* 55**,** 243-276.

Mylopoulos, J. 1992. Conceptual Modeling and Telos. *In:* Loucopoulos, P. & Zicari, R. (eds.) *Conceptual modeling, databases, and case : an integrated view of information systems development.* Newyork: Wiley.

M&S Magazine. 2010. Cocmeptual Modelling corner project : The Society for modelling and Simulation International , 2010 (http://www.scs.org/magazines/2010-10/index_file/ConceptualModelingCorner.htm), assessed Feb 2014)

Nance, R. E. 1979. "Model Representation in Discrete Event Simulation: Prospects for Developing Documentation Standards". Academic Press, Inc., New York.

Nathan, B. & Wood, J. 1991. "The Use of IDEF0 to Document a Methodology-a Novices Point of View". *Automation and Robotics Research Institute.* Fort Worth, Texas

Nelson, H. & Monarchi, D. 2007. Ensuring the quality of conceptual representations. *Software Quality Journal,* 15**,** 213-233.

Nelson, H. J., Poels, G., Genero, M. & Piattini, M. 2012. A conceptual modeling quality framework. *Software Quality Journal,* 20**,** 201-228.

Onggo, B. S. S. 2009. Towards a unified conceptual model representation: a case study in healthcare. *J of Sim,* 3**,** 40-49.

Oral, M. & Kettani, O. 1993. The facets of the modeling and validation process in operations research. *European Journal of Operational Research,* 66**,** 216-234.

Pace, D. K. Conceptual model development for C4ISR simulations. Proceedings of the 5th international command and control research and technology symposium, 2000. 24-26

Page-Jones, M. 1980. *The practical guide to structured systems design,* Yourdon Press.

Parnas, D. L. & Darringer, J. A. SODAS and a methodology for system design. Proceedings of the November 14-16, 1967, fall joint computer conference, 1967 1967 New York, NY, USA. ACM, 449-474.

Parsons, J. & Cole, L. 2005. What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data & Knowledge Engineering,* 55**,** 327-342.

pidd , M. 1999. *Just Modeling Through: A rough guide to modeling. Interfaces March/April.*

Pidd, M. 2004. *Computer Simulation in management science,* Chichester, John Wiley & Sons.

Pidd, M. Making sure you tackle the right problem: Linking hard and soft methods in simulation practice. Winter Simulation Conference, 9-12 2007. 195-204.

Prakash, N., Rolland, C. & Pernici, B. Product Modeling for Requirements Engineering Process Modeling£. Information System Development Process: Proceedings of the IFIP WG8. 1 Working Conference on Information System Development Process, Como, Italy, 1-3 September, 1993, 2014. Elsevier, 231.

Pratt, D., J. Mize, and M. Kamath, 1993, "A case for bottom-up modeling" . *2nd Industrial Engineering Research Conference Proceedings,* IIE, Norcross. GA, pp. 430-434

Qi, Y.-d., Xie, X.-f., Zhu, A.-h. & Yan, X.-b. Analysis of contribution of conceptual model quality to software reliability. Computer Application and System Modeling (ICCASM), 2010 International Conference on, 2010 2010. V10-386-V10-389.

Robinson, S. 1997. Simulation model verification and validation: increasing the users' confidence. *Proceedings of the 29th conference on Winter simulation.* Atlanta, Georgia, USA: IEEE Computer Society.

Robinson, S. 2002. Modes of simulation practice: approaches to business and military simulation. *Simulation Modelling Practice and Theory,* 10**,** 513-523.

Robinson, S. 2004a. Discrete-event simulation: from the pioneers to the present, what next? *J Oper Res Soc,* 56**,** 619-629.

Robinson, S. 2004b. *Simulation: the practice of model development and use,* John Wiley & Sons.

Robinson, S. 2007. The future's bright the future's[hellip]Conceptual modelling for simulation! *J of Sim,* 1**,** 149-152.

Robinson, S. 2008a. Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society,* 59**,** 278-290.

Robinson, S. 2008b. Conceptual modelling for simulation Part II: a framework for conceptual modelling. *J Oper Res Soc,* 59**,** 291-304.

Robinson, S. Choosing the right model: Conceptual modeling for simulation. Simulation Conference (WSC), Proceedings of the 2011 Winter, 11-14 Dec. 2011 2011. 1423-1435.

Robinson, S. Tutorial: Choosing what to model — Conceptual modeling for simulation. Simulation Conference (WSC), Proceedings of the 2012 Winter, 9-12 Dec. 2012 2012. 1-12.

Robinson, S. Conceptual modeling for simulation. Simulation Conference (WSC), 2013 Winter, 8-11 Dec. 2013 2013. 377-388.

Robinson, S., Brooks, R., Kotiadis, K. & Van der zee, D.-J. 2011. *Conceptual modeling for Discrete Event Simulation,* Boca Raton London Newyork, CRC Press.

Robinson, S. 2014. *Simulation: the practice of model development* and use (2nd Edition), Palgrave Macmillan.

Rolland, C. & Cauvat, C. 1992. Trends and Perspectives in Conceptual Modelling. UniversitÃƒÂ© PanthÃƒÂ©on-Sorbonne (Paris 1).

Rolland, C. & Prakash, N. 2000. From conceptual modelling to requirements engineering. *Annals of Software Engineering,* 10**,** 151-176.

Ross, D. T. 1977. Structured Analysis (SA): A Language for Communicating Ideas. *Software Engineering, IEEE Transactions on,* SE-3**,** 16-34.

Ross, D. T. 1985. Applications and Extensions of SADT. *Computer,* 18**,** 25-34.

Ross, D. T. & Schoman, K. E. 1977. Structured Analysis for Requirements Definition. *IEEE Trans. Softw. Eng.*, 3**,** 6-15.

Santarek, K. & Buseif, I. M. 1998. "Modelling and Design of Flexible Manufacturing Systems using SADT and Petri Nets Tools". *Journal of Materials Processing Technology,* 76**,** 212-218

Sargent, R. G. 2005. Verification and validation of simulation models. *Proceedings of the 37th conference on Winter simulation.* Orlando, Florida: Winter Simulation Conference.

Sargent, R. G., Mize, J. H., Withers, D. H. & Zeigler, B. P. 1993. "Hierarchical Modeling for Discrete Event Simulation (panel)". *Proceedings of the 1993 Winter Simulation Conference.* G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles, eds. 569 – 572, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sargent, R. G. Verification and validation of simulation models. Simulation Conference (WSC), Proceedings of the 2011 Winter, 11-14 Dec. 2011 2011. 183-198.

Sehruben, L. 1979. Designing correlation induction strategies for simulation experiments. Current Issues in Computer Simulation, edited by NR. Adam and A. Dogremaci Academic Press, Inc., New York.

Shanks, G., Tansley, E. & Weber, R. 2003. Using ontology to validate conceptual models. *Communications of the Acm,* 46**,** 85-89.

Siau, K. 2004. Informational and computational equivalence in comparing information modeling methods. *Journal of Database Management (JDM),* 15**,** 73-86.

Sommerville, I. & Sawyer, P. 1997. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering,* 3**,** 101-130

Sommerville, I., Sawyer, P. & Viller, S. 1998. Viewpoints for Requirements Elicitation: A Practical Approach. *Proceedings of the 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice.* IEEE Computer Society.

Sowa, J. F. 1984. *Conceptual Structures: Information Processing in Mind and Machine,* Reading, Addison-Wesley.

Stavely, A. M. 1983. Modeling and projection in software development. *Journal of Systems and Software,* 3**,** 137-146.

Steen, M. W. A., Akehurst, D. H., ter Doest, H. W. L. & Lankhorst, M. M. Supporting viewpoint-oriented enterprise architecture. Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International title={, 2004 2004. 201.

Sugumaran, V. & Storey, V. C. 2002. Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering,* 42**,** 251-271.

Sutcliffe, A. G., Maiden, N. A. M., Minocha, S. & Manuel, D. 1998. Supporting scenario-based requirements engineering. *Software Engineering, IEEE Transactions on,* 24**,** 1072-1088.

Tako, A. A., Vasilakis, C. & Kotiadis, K. A participative modelling framework for developing conceptual models in healthcare simulation studies. Simulation Conference (WSC), Proceedings of the 2010 Winter, 5-8 Dec. 2010 2010. 500-512.

Thayer, R. & Dorfman, M. 1997. Software requirements engineering glossary.

Tolk, A. & Muguira, J. A. The levels of conceptual interoperability model. Proceedings of the 2003 Fall Simulation Interoperability Workshop, 2003. Citeseer

Tolk, A., Diallo, S. Y., Padilla, J. J. & Herencia-Zapana, H. 2013. Reference modelling in support of M&S[mdash]foundations and applications. *J of Sim,* 7**,** 69-82.

Tolk, A. & Turnitsa, C. CONCEPTUAL MODELING WITH PROCESSES. *In: C. Laroque, J. H., R. Pasupathy, O. Rose,A.M. Uhrmacher,* ed. Winter Simulation Conference, 2012 US.

Turnitsa, C., Padilla, J. J. & Tolk, A. Ontology for Modeling and Simulation. Simulation Conference (WSC), Proceedings of the 2010 Winter, 5-8 Dec. 2010 2010. 643-651.

Van der zee, D.-J. 2011. Developing Participative Simulation Models : Framing Decomposition Principles for joint understanding. *In:* Robinson, S., Brooks, R., Kotiadis, K. & Van der zee, D.-J. (eds.) *Conceptual modeling for Discrete Event Simulation.* Boca Raton London Newyork: CRC Press.

van der Zee, D.-J. An integrated conceptual modeling framework for simulation ; Linking simulation modeling to the systems engineering process. 9-12 Dec. 2012 2012. 1-12.

van der Zee, D. J. & Van der Vorst, J. G. A. J. Guiding principles for conceptual model creation in manufacturing simulation. Simulation Conference, 2007 Winter, 9-12 Dec. 2007 2007. 776-784.

van Lamsweerde, A. Goal-oriented requirements engineering: a guided tour. Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on title={Goal-oriented requirements engineering: a guided tour, 2001 2001. 249.

Van Rensburg, A. & Zwemstra, N. 1995. “Implementing IDEF Techniques as Simulation Modelling Specifications”. *Computers & Industrial Engineering,* 29**,** 467-471.

Wand, Y., Monarchi, D. E., Parsons, J. & Woo, C. C. 1995. THEORETICAL FOUNDATIONS FOR CONCEPTUAL MODELING IN INFORMATION-SYSTEMS DEVELOPMENT. *Decision Support Systems,* 15**,** 285-304..

Wand, Y. & Weber, R. 1990. AN ONTOLOGICAL MODEL OF AN INFORMATION-SYSTEM. *Ieee Transactions on Software Engineering,* 16**,** 1282-1292.

Wand, Y. & Weber, R. 2002. Research commentary: Information systems and conceptual modeling - A research agenda. *Information Systems Research,* 13**,** 363-376.

Ward, S. C. 1989. Arguments for constructively simple models. *Journal of the operational research society***,** 141-153.

Whitman, L., Huff, B. & Presley, A. 1997. “Structured Models and Dynamic Systems Analysis: The Integration of the IDEF0/IDEF3 Modeling Methods and Discrete Event Simulation”.

Proceedings of the 1995 Winter Simulation Conference. S.Andradbttir, K. J. Healy, D. H. Withers, and B. L. Nelson , eds. 518 - 524, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc

Appendices

Appendix 1: User Guide on SADT CM

A1.1 Introduction

This workbook explains the procedure for applying the SADT CM framework of Conceptual Modelling (CM) on Discrete Event Simulation (DES) problems. A software engineering method known as Structured Analysis and Design Technique (SADT) has been adopted and modified to be used for this purpose. This being the reason, the methodology is called SADT CM. It is expected that by using SADT CM, a more structured approach for CM is learned by the modellers. Since at present, most of the techniques are subjective in nature, SADT CM aims to provide a better structuring of the problem and allows a mechanism to perform the abstraction process in CM which is largely absent in other approaches. This work book is aimed at the modellers who have the basic knowledge of DES and wants to model the problem in a precise manner so as to gain benefits during the initial stages of simulation study. In the subsequent sections, each step of the framework is discussed with details and guidelines for the modeller. SADT CM framework is divided into two main parts as per the CM artefacts presented by Robinson (2011) and shown in figure A1.1. These are as follows:

1 - Generation of System Description

2 - Abstraction of Conceptual Model

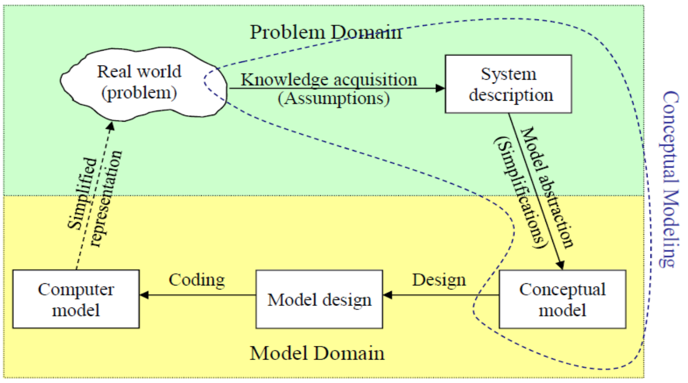


Figure A1.1: Artefacts of Conceptual Modelling (Robinson 2011)

While System Description (SD) helps in knowledge acquisition from the real world problem domain, the abstraction phase helps to simplify this SD into a conceptual model by removing any irrelevant details required for modelling. SADT CM helps to achieve these stages in a structured framework. In first phase, it collects information about a problem situation and the main components involved with respect to that problem (Activities, Resources, Entities and Controls). In second phase, it provides means to analyse this information by removing any unnecessary element of SD through abstraction. Before the explanation of the actual framework for both phases, notation of SADT CM and a simple bank problem case that will be used to illustrate the framework are presented.

A1.2 Notation of SADT CM

SADT CM block is shown in figure A1.2.

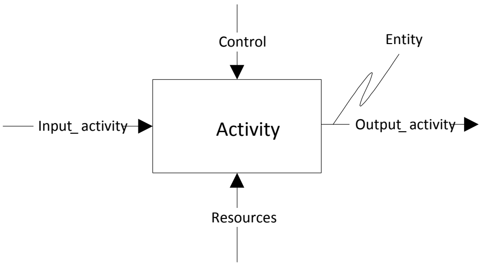


Figure A1.2: Notation of SADT CM.

The definitions of the labels used are as follows (figure A1.2):

- ***Activity***:

This corresponds to the activity under consideration (transformation activity). This is the central part of the analysis. Therefore, it is important to identify all activities carefully in the system. There is only *one* transformation activity in a single SADT CM block. The label uses a notation of alpha-numeric style indicating the operation or activity (e.g. OP1, and A1 to indicate operation 1 and activity 1 in the system).

- ***Input activity***:

This determines all activities preceding the activity. Arrows can be replicated in case of more than one input activity.

- ***Output activity*:**

This determines all activities following the activity. Arrows can be replicated in case of more than one output activity.

- ***Resources*:**

Any resource (or group of resources) required by the activity is indicated by the bottom arrow of the box. The notation used is “R_x” where “x” represents the ID of a resource (or group of resources). In case of more than one ungrouped resource, “x” can indicate this value as a range for e.g. R_{1-3} means three individual resources (R_1, R_2, and R_3) are required on a particular activity. The values of “x” are unique unless used for shared resources (or a group of resources) on different activities. Arrows can be replicated for two completely different types of resources on a single activity, for example, the operators and transportation resources can be both represented by R_x and T_x on two different arrows respectively.

- ***Controls*:**

This represents any constraint imposed on the activity by the environment, management, production plans, customer demand etc. A simple notation “C_X” is used, where “x” denotes the ID number of the constraint. This label will also be the same for activities sharing any similar constraints. Arrows can be replicated for multiple constraints on the same activity.

- ***Entities*:**

This label represents the entities flowing out from the activity and following this route. These are represented by E_x where “x” represents the ID of the entity. In case of more than one entity type on single branch, entities are denoted by E_xE_y where “X” and “Y” represents the IDs of two different entities. Other acceptable notations for different type of entities on one branch are Ex\Ey and Ex+Ey. The labels are used in alpha-numeric style to indicate different components of SADT CM block. For example, E_1, E_2 for entities, OP_1 and $OP_{2,}$ for operations, and R_1 and R_2 for resources are used. Where more than one resource or control required for a single activity, same multiple entity style notation is used. For shared resources, a range can also be specified, for example, R_{1-5} (to indicate 5 resources required for an activity). It is also acceptable to use regular fonts in alpha-numeric fonts without using subscript. For e.g. R1 can be used instead of R_1 and E1E2 in place of E_1E_2.

A1.3 Bank problem

Customers enter the bank where they need to go to the required service counter. They first meet a Customer Representative (CR) who directs them to either the cash counter or bill payment counter. Both counters are served by dedicated bank tellers. The bank is facing a queuing problem at the cashier counter. The manager of the bank is interested to know about the number of customers served by the cashier during any busy hour of the day (their waiting times also) and whether it is beneficial to invest in another cashier for this service. The manager can take the decision to hire a new cashier if the average waiting time of the customers is more than five minutes.

A1.4 SADT CM Framework

Now we explain the SADT CM framework with reference to the bank problem. The SADT CM framework consists of the following steps:

Part A: Generation of System Description (SD)

- Step SD1: Identify all the entities in the problem
- Step SD2: Identify all activities in the problem
- Step SD3: List any control associated with each activity
- Step SD4: List any resources associated with each activity
- Step SD5: Draw the SADT CM branches for each activity
- Step SD6: Combine all SADT branches to generate a complete SD
- Step SD7: Verification and validation of SD

Part B: Abstraction of Conceptual Model from SD

- Step CM1: Identify the modelling objective
- Step CM2: Identify the experimental factor
- Step CM3: Identify the Objective Activity(OA)

- Step CM4: Identify the Connected Activity (CA)
- Step CM5: Conduct CA impact test
- Step CM6: Check if all CAs have been analyzed
- Step CM7: Analysis of Controls, Resources and Entities
- Step CM8: Analysis of the queues
- Step CM9: Make abstracted diagram of the conceptual model
- Step CM10: Verification and validation of conceptual model

Generation of System Description (SD)

The generation of the SD is achieved by following the set of steps as listed in Part A. The process starts with identifying the major components of a conceptual model (activities, resources, entities, controls) followed by a construction of the CM SD in diagrammatic format using the SADT CM notations. The whole process of generating a SD can also be understood with the help of the flowchart in figure A1.3. The flowchart explains that after step 4, a thorough re-check is required to ensure that all activities, resources, controls and entities have been identified correctly. A final verification of SD is also performed at the end of part A. The two tables used to record information in all these steps are shown in figure A1.4. They are called entity and activity tables. These tables will be filled-in as we go through each step of part A. Entity Table(s) will be used to record information in step 1 while Activity Table(s) will be used to record information from steps 2 to 5. The first instance of these tables are blank and then subsequently numbered as the information is added at each step.

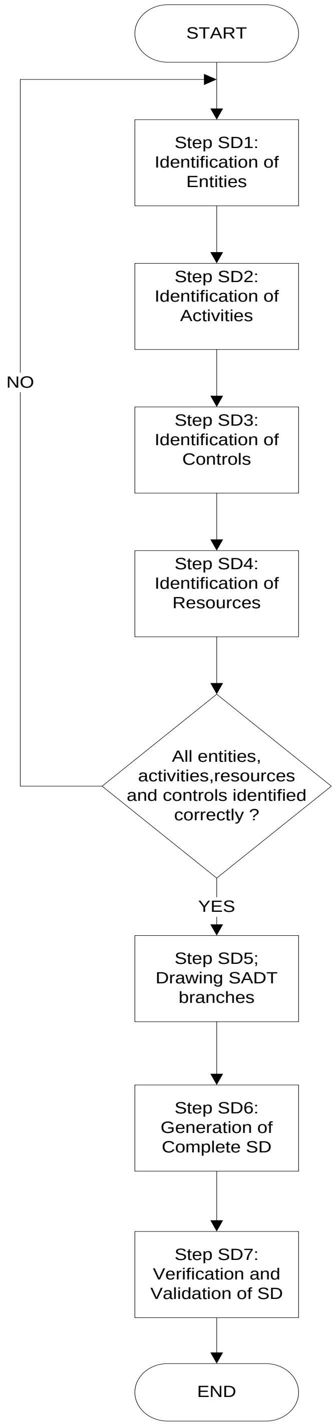

START
Step SD1: Identification of Entities
Step SD2: Identification of Activities
NO
Step SD3: Identification of Controls
Step SD4: Identification of Resources
All entities, activities,resources and controls identified correctly ?
YES
Step SD5; Drawing SADT branches
Step SD6: Generation of Complete SD
Step SD7: Verification and Validation of SD
END

Figure A1.3: Process of generating SD in part A of SADT CM

Entity Table:

Entity	Label

Activity Table:

Activity	Control	Resource	Output Entity

Figure A1.4: Tables for recording information through Steps 1 to 6 in part A

Step SD1: Identification of entities

The first step is to identify all entities in the system. These include any type of raw material, product, moving objects, people, cars, unfinished products, pallets, fixtures, telephone calls, etc. All entities are listed in a table allocating a unique label name to each entity (Ei). The list of the entities can be generated by looking into the problem description and locating any possible entities in the system, through discussion with managers or project leader and checking any other information available (layout diagram, customer notes, operations details etc.). It is not necessary to identify any combined or split entities explicitly at this stage as this will be determined in the later stages when activities and output entities are determined. The Entity Table (figure A1.4) is used for recording the information at this step.

Bank Problem: Entities

The entities identified for the bank problem is recorded in Entity Table (1).

Entity Table (1)

Entity	Label
Customers (Cashier)	E1
Customers (Billing)	E2

Step SD2: Identification of Activities

In this step, all activities are defined and labels are assigned to them. The activities are carefully identified by brainstorming, aiming to identify all possible activities in the problem domain and concerning any other related data given about the problem (for example, lay out diagram, tables or flow charts etc.). Some of the possible general activities can include (but not limited to) the following:

- Arrival (can be initiated by entities like customers, telephone calls, data signals, raw materials etc.)
- Services (for example, machine operations, calls servicing, customers' service, drivers parking or driving etc.)
- Exit (executed usually when any finished product or service leaves the system)

Any Activities identified are placed in the "Activity" column of the Activity Table (figure A1.4). Only one activity must be placed at each field of the column and an appropriate label should be assigned to this within parenthesis. For the entities that form the starting point of the flow of entities in the system, a prefix of "Sc" is attached with their label. This is to indicate that this activity acts as "Source" in the system. The same is repeated for the activities which exits the entities out from the system with the pre-fix of "Sk" added to their label to indicate the "Sink" in the system. The "Activity" column in the Activity table is used to record the information at this step.

Bank Problem: Activities

The activities identified for the Bank problem are listed in Activity Table (1) along with the label information as shown below.

Activity Table (1): Activities in the Bank Problem (Step SD2)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)			
Meeting CR (A2)			
Cashier Service(A3)			
Billing Service(A4)			
Customers Leaving Sk(A5)			

Step SD3: Identification of control related to activities

In this step, any control is identified related to the activities identified in step SD2. Controls can be seen as “constraints” on any single activity that limits the process at this point and hence can affect the output in the complete operations chain. Constraints can take many forms as explained below:

- Physical (tangible or non-tangible: examples include equipment maintenance, set-up times, repair times, material shortages, lack of people, lack of space, product types, arrival patterns, technology, layout, buffer specification etc.)
- Policy (company procedures, union contracts, government regulations: examples include overtime policies, mandatory breaks, maximum working hours, new employee training etc.)
- Paradigm (some beliefs or habits: for example, the belief that “we must always keep our equipment running to lower the manufacturing cost per piece”)
- External (environmental, safety or geological concerns: for example, a particular machine may be sensitive to changing weather and hence require a specific time for running)

A modeler is advised to consider all the above points to determine any possible constraints in the problem. A careful discussion around each activity is required. Some constraints might not be visible at first sight, but can come to light through detailed discussions in the initial meetings of the simulation study. Constraints are identified as “Controls” in SADT CM in a way that it determines whether an activity can proceed or not under certain conditions. In some cases, it controls the pace of an activity in terms of restriction for a condition to be met before processing .Also controls are not to be confused with “Resources” which are explained in the next step. The information obtained in this step is recorded in the “Control” column of Activity Table along with the assignment of a label (C_x).

Bank Problem: Controls

The only control that can be identified through the problem description in the case of the bank problem is the customer arrival pattern (based on their choice of service), since this will largely affect the queuing of people at both the cashier and billing counters. This is recorded along with the label in Activity Table (2) as shown below:

Activity Table (2): Control in the Bank Problem (Step SD3)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)		
Meeting CR (A2)	N/A		
Cashier Service(A3)	N/A		
Billing Service(A4)	N/A		
Customers Leaving Sk(A5)	N/A		

Step SD4: Identification of resources related to activities

In this step, any resources are identified that relate to the activities. The resources fall under the following categories:

- Humans (examples: workers, machine operators, maintenance operators, shift operators, security staff, line managers, customer representatives, service operators, hygiene persons, etc.)
- Equipment (examples: tools, forklifts, computers, barometer, thermometer, pressure gauge, heat sensors, fire detectors etc.)
- Information (examples: databases, schedule sheet, time sheets, capacity and material plan etc.)
- Natural (examples: air, water, land, minerals, etc.)

It is advised to identify the resource in a problem by considering all potential resources described above and relating them to every activity. This can be done by reading the problem description, meeting with the managers, observing the factory personnel, taking data about the organizations resources from the operations managers and discussing any external or natural resources to be used within the processes. It is also important to distinguish between the nature of the resource (shared or dedicated). A shared resource is shared among different activities while a dedicated resource is only required by a single activity and is available at all times for that activity. The resources are documented in the Activity Table in the “Resource” column along with a unique label. If a resource appears more than once in a table (shared resource) then same label will be used at each occasion. A dedicated resource is represented by a unique label. Before moving on to the next step, it is advised to verify all identified components in the Activity Table obtained in this Step (SD4) as this can affect the drawing of the SADT CM branches and any missing component would be hard to locate in the later stages.

Bank Problem: Resources

In the case of the Bank Problem, the resources are identified are the CR, cashier and billing accountant. These are documented in Activity Table (3) as follows:

Activity Table (3): Resources in Bank Problem (Step SD4)

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	
Meeting CR (A2)	N/A	CR (R1)	
Cashier Service(A3)	N/A	Cashier (R2)	
Billing Service(A4)	N/A	Bill Accountant (R3)	
Customers Leaving Sk(A5)	N/A	N/A	

To verify that Activity Table (3) contains all components of the bank problem it can be carefully checked against the problem description to ensure that no activity, control or resource is missing at this stage.

Step SD5: Drawing SADT branches

In this step, all the information we attained from steps SD1 to SD4 is used to create SADT CM branches. Before drawing the SADT CM branches, the last column of the Activity Table is filled-in. The "output entity" column is filled-in by considering any possible entity on its output path along with its label. This can be thought of as an output from an activity. These entities are labeled as explained in section A1.2. These entities will be placed on the "Entity" label of the SADT CM block. These entities must have been identified in step SD1 and no new entity should be identified at this step. The only difference is to jointly or separately list entities from the Entity Table as per the output path of the activity under consideration. Once this is done, SADT branches can be created using the symbols and notations of the SADT block as explained in section A1.2 of this workbook. This will contain input activity, output activity, output entity, control, and resource with respect to the main activity.

Bank Problem: Drawing SADT CM branches

For the Bank Problem, the output entity column is filled up as shown in Activity Table (5) below and corresponding SADT CM branches are shown in figure A1.5.

Activity Table (5): Output Entities for the Bank Problem

Activity	Control	Resource	Output Entity
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2
Meeting CR (A2)	N/A	CR (R1)	E1 E2
Cashier Service(A3)	N/A	Cashier (R2)	E1
Billing Service(A4)	N/A	Bill Accountant (R3)	E2
Customers Leaving Sk(A5)	N/A	N/A	N/A

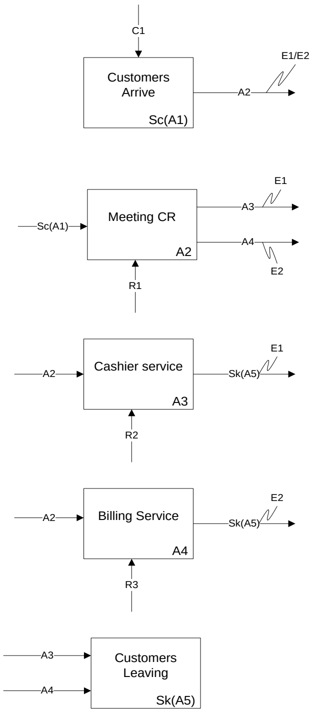


Figure A1.5: SADT CM branches for the bank problem (STEP SD5)

Step SD6: Generation of complete SD

After drawing the branches in Step SD5 individually, they are combined in this step to produce the compete SD. The following steps ensure the correct generation of the complete SD from the individual SADT CM branches.

- Start with a "source" (preceding) activity and follow the output activity.
- Locate the branch with the main activity similar to the "output activity" of the preceding activity. Consider this as the following activity.
- Connect the output activity arrow of the source (preceding) activity to the input activity position on the following activity. Remove any input activity arrow on the following activity.
- Remove any activity label from this new connected arrow and label this arrow with the output entity of the preceding activity.
- Consider this following activity as a new preceding activity.
- Repeat all steps as prescribed above until only the "sink" activity is left.
- Connect the final sink activity in the same manner but without any output activity or entity labels.
- In case of multiple sources and multiple sink in a system, repeat all the above steps until no source and sink are left unconnected.

Bank Problem: Compete SD from SADT CM branches

Figure A1.6 presents the complete SD for the bank problem obtained as a result of observing the steps explained in step SD6.

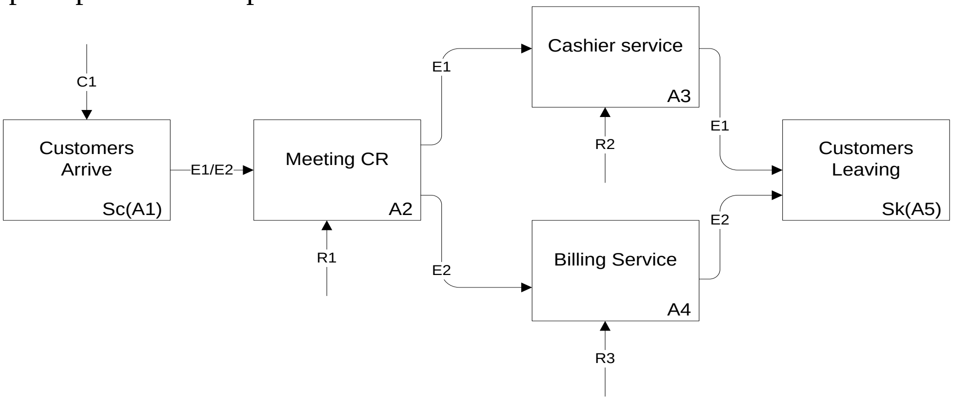


Figure A1.6: complete SD for the bank problem (STEP 6)

Step SD7: Verification and Validation of SD

A verification of the SD can be performed after completing all the steps of generating a SD. This primarily requires a thorough check in the last two steps where the branches of the SADT CM have been created. According to the flowchart in figure A1.3, an intermediate verification should already have been performed before the drawing of the SADT CM branches to ensure that no component from the problem description (activities, entities, resource and control) is missing. The verification of the complete SD will involve consistency checking of the entities from one activity to another and that all activities are linked together. Also, verification should ensure that the correct source and sinks have been put in place and linked to the following and preceding activity. The labels also need to be verified and cross-checked with the information in the completed Entity Table (step SD1) and Activity Table (Step SD5). Similarly all other information from the Activity Table (in step SD5) is used to verify the complete SD diagram. The validation of the SD can be achieved by comparing the problem situation with the SD layout and getting a sense about how closely it reflects the actual problem at this stage. Some discussion is probably required also at this stage to validate the SD. The complete SD diagram is a useful basis for this discussion.

Abstraction of Conceptual Model

The abstraction of the conceptual model from the SD is the next part of the CM process (Part B). We need to refer to completed Activity Table (Step SD5) and completed SD layout (Step SD6) to abstract the conceptual model. The flowchart of figure A1.7 shows the abstraction process. The whole procedure of abstraction goes through the following steps:

- Step CM1: Identify the modelling objective
- Step CM2: Identify the experimental factor
- Step CM3: Identify the Objective Activity(OA)
- Step CM4: Identify the Connected Activity (CA)
- Step CM5: Conduct CA impact test
- Step CM6: Check if all CAs have been analyzed
- Step CM7: Analysis of Controls, Resources and Entities
- Step CM8: Analysis of the queues
- Step CM9: Make abstracted diagram of the conceptual model
- Step CM10: Verification and Validation of conceptual model

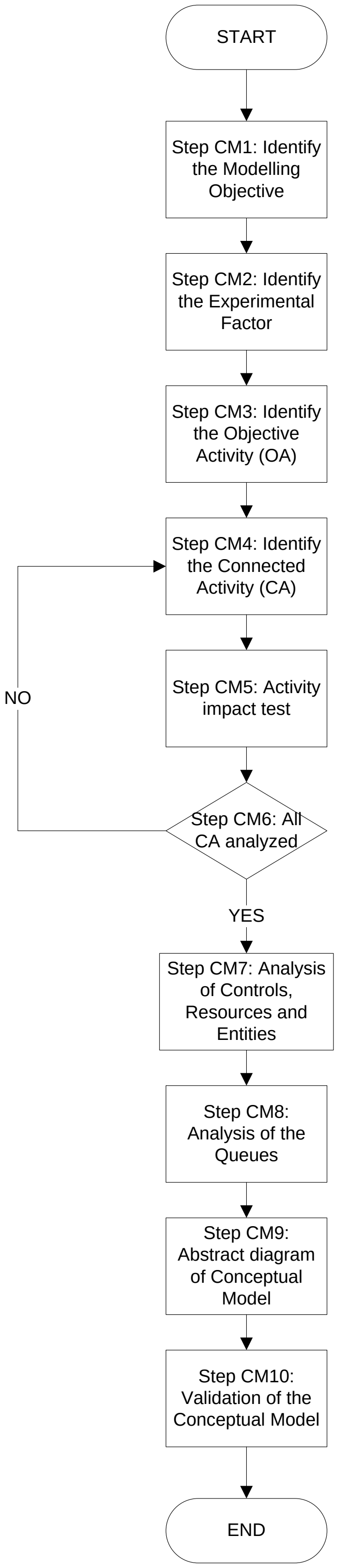

START
Step CM1: Identify the Modelling Objective
Step CM2: Identify the Experimental Factor
Step CM3: Identify the Objective Activity (OA)
Step CM4: Identify the Connected Activity (CA)
Step CM5: Activity impact test
Step CM6: All CA analyzed
NO
YES
Step CM7: Analysis of Controls, Resources and Entities
Step CM8: Analysis of the Queues
Step CM9: Abstract diagram of Conceptual Model
Step CM10: Validation of the Conceptual Model
END

Figure A1.7: Steps in Abstraction of Conceptual Model

Some steps are iterative (as shown) and will be explained accordingly in the following sections. An extended table, called the Scope Table (updated from the columns of Activity Table), is shown in figure A1.8 which is used to record all the information obtained in the different steps of part B. Similar to the Activity Table, the Scope Table is also updated as we move through the different steps of abstraction procedure. The first four columns in the Scope Table are simply copied from the Activity Table (obtained in Step SD5) before moving on with the abstraction phase.

Scope Table:

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?

Figure A1.8: Table for recording information in all steps of Part B

Step CM1: Identify the Modelling Objectives

This is the most critical part of the whole procedure. The modelling objectives need to be clearly identified at this stage before starting any abstraction. The user needs to be sure what they require (output/response) from their model. For example, if the objective is to measure the waiting time of certain entities in a queue, then this should be listed down as the primary focus other than any general project objectives (for example, time frame to develop the model, flexibility, interoperability, visual requirements etc.). This is because later stages of abstraction are strongly related to correct identification of the specific modelling objectives. If unsure about the modelling objectives then discussion with managers and other stakeholders needs to be conducted and the problem brainstormed in meetings to reach a consensus about the modelling objectives.

Bank Problem: Modelling Objective

In the case of the bank problem, the modelling objective is to study the output (response) for the customers' queue developing at the cashier (in this case, waiting times and not the queue size) and to check if this is within an acceptable limit (5 mins).

Step CM2: Identify the Experimental Factors

The experimental factors are the "input" to the model and needs to be identified as they play an important role in abstraction. They are the main parameters to test during experimentation with the model. This again can be precisely identified through discussion with managers and by knowing the allowable changes to the model.

Bank Problem: Experimental Factor

The only experimental factor identified for the bank problem is the cashier labeled "R2" from Activity Table (2) since the manager is interested to know if the investment in hiring an extra cashier can decrease the waiting time of the customers in the cashier queue to within an acceptable limit of 5 mins.

Step CM3: Identify the Objective Activity (OA)

The next step is to identify the OA. This is very closely related to the modelling objective and could be identified through discussion around the SD. This activity is the key activity that is needed to measure the identified response in terms of modelling objective discussed in step

CM1. This can be the immediate activity linked to any of the modelling objectives, e.g. no of people in the queue, waiting time of the people in the queue, resources needed etc. It should be noted that this activity needs very careful selection as this is the starting point of the whole abstraction process. For example, an activity is considered an OA if the immediate queue behind that activity is to be considered for measuring the response (waiting time for instance) in the model.

Bank Problem: Objective Activity (OA)

In the bank problem, the OA is identified as A3 (cashier service) from Activity Table (2). By looking at the SD of figure A1.6 in Step SD6, it can easily be figured out that A3 is central to the modelling objective as the modelling objective requires accounting for the people in the cashier queue and no other activity other than A3 is responsible for developing queues for customers seeking cashier service.

Step CM4: Identify the Connected Activities (CA)

The Connected Activities (CA) are identified at this stage. It is any activity other than the OA (including source or sink activity) that directly connects to the activity under consideration or to any other CA in the system. Only one CA is identified at a time and the step is repeated if a new CA is to be identified (after step CM6). The SD layout obtained in Step SD6 can be consulted for the identification of CAs. The main purpose of identifying the CAs is to compare each CA against OA during the abstraction procedure (step CM5). An important point to note is that no strict pattern is necessary for identification of the CAs, but it is better to move from left to right and top to bottom in the SD diagram to avoid any activity being missed. Also it is better to start with the source activities and end with the sink activities. Once an activity is identified, it is marked in the SD layout and subsequent steps are performed.

Bank Problem: Connected Activity (CA)

In the bank problem, the first CA identified for the bank problem is A1 (Customer arrival) being the source activity in the SD layout. This is also the first activity appearing from left in the layout.

Step CM5: Conduct CA impact test

After the identification of the CA, seek to answer the following question:
Does the CA *impact* on the OA?

By answering “yes” or “no”, it is decided whether to include or exclude the CA from the conceptual model. At this point, the term “impacts” needs a detailed explanation to guide the user to a careful response to this question. Impact in a broad sense means any CA that has a significant impact on the accuracy of the model (i.e. on outputs/responses) with respect to the modelling objectives. A user should keep different aspects of “impact” in mind before applying this analysis. Also it is required by the user to understand the problem situation carefully (and hold meetings or discussions with managers if unsure) to answer this part. Some guidance on how to answer this question is provided below.
CA is considered to have an impact on the OA if:

The exclusion of the CA would result in an insufficiently accurate model

Some examples of these scenarios could be as follows:

- CA is a bottleneck activity in the model
- CA directly affect the flow of entities in the OA
- CA contains any shared resource, constraint or control
- CA is causing some variations in the OA (cycle time, maintenance, break-downs, etc.)
- CA is related with experimental factors
- CA is related with the modelling objective or OA in the whole process

As the analysis goes along, the results are recorded in the Scope Table. The activities are either kept or removed from the model depending upon the analysis of *impact* explained above. If the answer is “yes”, the activity is included in the Scope Table, but removed if the answer is “no”. The justification of the decision is also provided in the Scope Table.

Bank Problem: CA impact test

For the case of the bank problem, the first CA was identified to be A1 (customer arrival) in step CM4 and the OA was identified to be A3 (cashier service) in step 2. Having identified these two activities, analysis in step CM5 is carried out.

After careful examination we can decide that CA (A1) significantly impacts the OA (A3).

Hence A1 is added in Scope Table (1) as shown below with the justification. The justification in this case is that this activity is a source activity and initiates the arrival of entities in the system.

Scope Table (1): Activity A1 is added in the scope table after CA impact test

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities							
Meeting CR (A2)	N/A	CR (R1)	E1 E2										
Cashier Service(A3)	N/A	Cashier (R2)	E1										
Billing Service(A4)	N/A	Bill Accountant (R3)	E2										
Customers Leaving Sk(A5)	N/A	N/A	N/A										

Step CM6: Check if all CAs have been analysed

At this step, the SD layout is consulted again to see if all the CAs have been analyzed. If "yes" then the iterative process stops and we need to move to step CM7 else we need to repeat steps CM4, CM5 and CM6.

Bank Problem: Check for the CAs

In the case of bank problem, we see that there are still activities remaining to be analyzed from the SD layout (figure A1.6 in step SD6), so we repeat steps CM4, CM5 and CM6 until all the CAs are analyzed. Then we move on to step CM7. The results for the analysis of all the CAs and their justification are presented in Scope Table (2).

Scope Table (2):

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities							
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route							
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA							
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA							
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA							

Step CM7: Analysis of Entities, Resource and Controls

In this step, any entity, resource or control associated with the selected activity is analysed for inclusion or exclusion from the model. This is also recorded in separate columns of the Scope Table. This mainly requires considering if any of these particular components have some role in the model and it is valuable to keep them in model with respect to the modelling objectives. Separate columns are used to justify the inclusion/exclusion of the resources, entities and controls in the Scope Table. Some examples of reasons for the exclusion of these components are:

- Any resource or constraint that is linked to an activity which has been removed in the previous step will be removed.
- A resource can be excluded if it is a dedicated resource and always available
- An entity can be removed if it is not on the path affecting an OA
- A constraint can be removed from modelling if it is an unlikely event to happen during a specified model run (example non-shift times or environmental constraint)

Bank Problem: Analysis of entities, controls and resources

After the analysis of the activities, all the resources, controls and entities for the bank problem are analysed and results are recorded in Scope Table (3). The justification of each component is also provided in the table.

Scope Table (3): Analysis of the controls, resources and entities

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities	E1/E2- Yes	main arrival entities	N/A	N/A	C1-Yes	determines arrival pattern	
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route	E1- YES E2 - NO	E1affects OA / E2 has no affect on OA	R1- No	dedicated resource	N/A	N/A	
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA	E1-YES	E1 flows into OA	R2-YES	Experimental Factor	N/A	N/A	
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA	E2-NO	path not modelled	R3-No	path not modelled	N/A	N/A	
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA	N/A	N/A	N/A	N/A	N/A	N/A	

Step CM8: Analysis of the queues

To complete the Scope Table the requirement for queues is analyzed for each activity. The requirement for a queue is obvious if an activity is not a “source” or “sink” in the complete SD. The other cases of having/not having a queue need to be identified from the SD or through discussion with managers at the meetings and brainstorming all the aspects of the problem, for example enough capacity is available, infinite queue length permitted, experimental factors, model responses etc.. This information is recorded in the last column of the Scope Table. Also it needs to be identified whether single or multiple queues are needed for any activity. This justification can be put within the same column as inclusion or exclusion of the queue. In the case of a single activity that is fed by multiple queues, all queues are listed with labels besides the activity row, and likewise if the same queue is serving multiple activities then same label of the queue is assigned under different activities row.

Bank Problem: Queue analysis

In the case of the bank problem, the queue analysis against each selected activity is carried out and results are recorded in last column of Scope Table (4). There is a need for separate single queues for services before meeting CR and waiting for the cashier service.

Scope Table (4): Analysis of the Queues

Activity	Control	Resource	Output Entity	CA impacts OA?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required ?
Customers Arriving Sc(A1)	Customers Choice (C1)	N/A	E1/E2	YES	YES	Main entry point for the entities	E1/E2- Yes	main arrival entities	N/A	N/A	C1-Yes	determines arrival pattern	No (Entry point)
Meeting CR (A2)	N/A	CR (R1)	E1 E2	YES	YES	Determines customer route	E1- YES E2 - NO	E1affects OA / E2 has no affect on OA	R1- No	dedicated resource	N/A	N/A	YES(Single)
Cashier Service(A3)	N/A	Cashier (R2)	E1	YES	YES	OA	E1-YES	E1 flows into OA	R2-YES	Experimental Factor	N/A	N/A	YES(Single)
Billing Service(A4)	N/A	Bill Accountant (R3)	E2	NO	NO	Does not affect OA	E2-NO	path not modelled	R3-No	path not modelled	N/A	N/A	No (path not modelled)
Customers Leaving Sk(A5)	N/A	N/A	N/A	NO	NO	appears after OA	N/A	N/A	N/A	N/A	N/A	N/A	No (exit point)

Step CM9: Make an abstracted diagram of the Conceptual Model

After completion of the Scope Table, we can now proceed with the final step of drawing the layout of the conceptual model. This is drawn by taking information from the completed Scope Table (Step CM8) and completed SD layout in part A (Step SD6). All the components that have been analyzed and removed along with a justification in the completed scope table (Step CM8) will be removed in the final abstracted layout of the conceptual model. Any path that has not been modeled as recorded in the Scope Table will be removed. Also any control, resource, entity and activity that has not been included in the model as recorded in the Scope Table will be removed from the final abstracted diagram of conceptual model. This diagram is very easily drawn by removing any component from SD diagram produced in Step SD6, and connecting back the arrows in between without redrawing the whole diagram.

Bank Problem: Abstracted Conceptual Model

The complete abstracted conceptual model of the bank problem is shown in figure A1.9 as per justified through Scope Table (4). Any component decided not to be modeled in Scope Table (4) has been removed and figure A1.6 (complete SD) has been updated accordingly to produce figure A1.9.

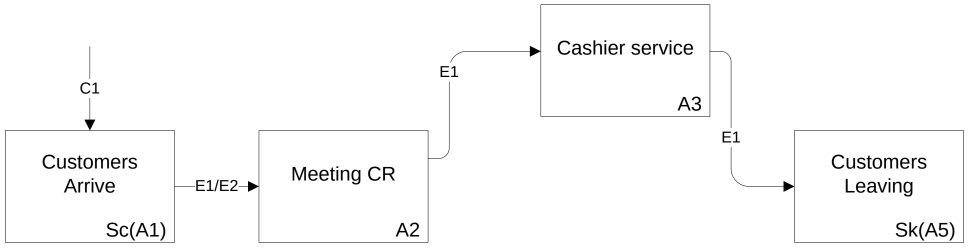


Figure A1.9: Abstracted lay out of the conceptual model for the bank problem

Step CM10: Validation of Conceptual Model

The final step is to perform a validation of the conceptual model generated as a result of the abstraction stage. Direct comparison of the SADT SD diagrams with the SADT CM diagrams would enable the abstractions in the model to be identified, debated and validated with subject matter experts.

Bank Problem: Validation of the conceptual model

Validation of the bank problem conceptual model is achieved by comparing the final abstracted layout of the conceptual model with the scope table. It can easily be seen that the only branch removed from the SD part in the abstracted model is the Billing service branch which serves no variation on the OA and hence the modelling objective is not affected. Also any controls, resources and entities that are removed in the abstracted model are again verified through the justification provided in the scope table.

Multiple Modelling Objectives

In the case of multiple modelling objectives, the modeller needs to repeat the CM stages (of which step 1 is to identify the modelling objective) for each objective they have identified. For each objective there will then be a separate model, so the final stage will be to combine the models. If the models overlap (have the same activities in) then they will probably be best combined, but if there is no overlap, then it would be a case of having separate models for each objective.

Appendix 2: Test cases used for SADT CM testing

Test Case 1: Panorama Televisions

"Panorama Televisions have been involved in the manufacturing of electrical goods since the early days of the radio. They now concentrate on the production of high quality, premium priced televisions for the international market. There are four televisions in their product range: small, medium, large and internet enabled. Last year, to meet increased demand, Panorama invested in a new television assembly plant. Also, after some negotiation with the unions, all areas of the site moved to continuous working over a five day week. However, the plant has never achieved its target throughput of 500 units per day. In fact, daily throughput is only just over 400 units. The plant is shown in figure A2.1

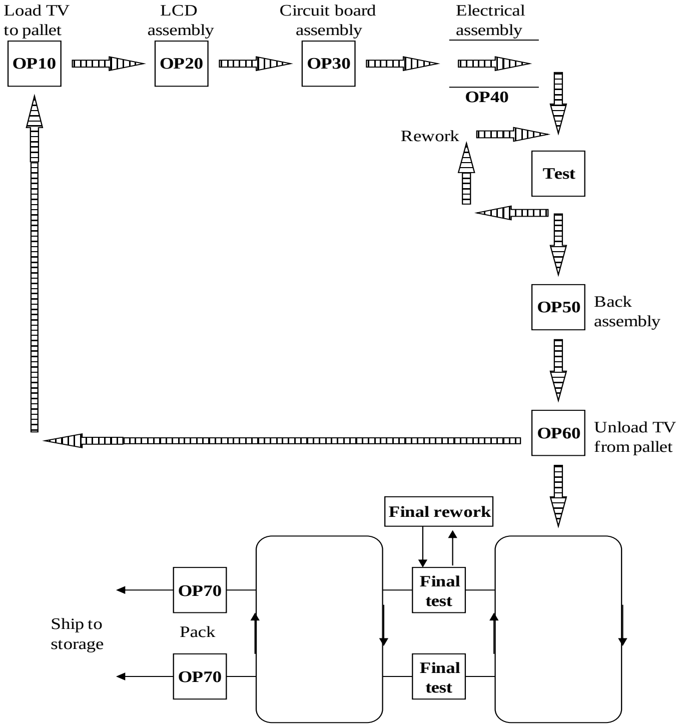


Figure A2.1: *Panorama Televisions: Television Assembly Plant, Robinson (2014)*

Plastic moulded boxes are loaded to a pallet by an operator at OP10. A production schedule, which is based on projected demand, determines the type of box to be loaded (small, medium, large or for internet enabled). At OP20 the LCD is assembled to the box before the circuit board is added at OP30. The televisions travel on a conveyor and five manual operators assemble the electrical equipment, OP40. The television is then tested and any failures go to the rework area. Good televisions have the back assembled at OP50 and are unloaded from the line at OP60 by an operator. The empty pallets are returned by conveyor to OP10 and the televisions are stored on a circular sling conveyor. A television is taken from the conveyor when a final test booth becomes available. Televisions failing this test are sent for final rework. Televisions passing are stored on another sling conveyor and are packed at OP70. Packed televisions are transported to the warehouse by forklift truck. The final test and packing area are often short of work and there is enough spare capacity to achieve 500 units per day. The management at Panorama believes that the throughput problem is a result of the number of stoppages on the main assembly line. There are a significant number of breakdowns on automated machines, and set-ups are required every time there is a change of product in the production schedule. There is only one maintenance engineer per shift who is required to attend all machine breakdowns and set-ups. There seems to be little opportunity to improve the efficiency of the machines, nor can the production schedule be changed since it is driven by customer demand. The solution being considered is to increase the buffering between the operations to dampen the effects of stoppages. Design engineers have considered this proposal and believe that, due to physical constraints on space, the buffering could be increased by a maximum of 200%. This will also require further pallets to be bought. In fact, there is some uncertainty as to whether enough pallets are currently being used. Increasing the number of pallets may provide a solution without the need for further storage. Extra storage is expensive, so before investing Panorama want to be sure it is necessary. Also, special pallets have to be used at a cost of $1,000 each, so it is important to minimize the number required. Target throughput must be achieved, but expenditure should be kept to a minimum. The management at Panorama is looking for some proposals on how to improve the line within ten working days"

Robinson (2014)

Test case 2: Wardeon Cinema Problem

“Wardeon Cinemas own and manage multiplex cinemas in the United Kingdom. Their Nottingham complex has been open for five years and is very successful in attracting clientele, despite fierce competition in the region from other cinema groups. Recently, however, there have been a number of complaints about the telephone enquiries and booking service.

The telephone system was installed at the time the complex was opened. It provides a message service giving information on film times and an option to seek further information or book tickets by speaking to a booking clerk. Initially this service was adequate, but with rising demand the telephone queues are now often full, particularly on the busiest days (Saturdays). As a result, callers either balk (give up having received an engaged tone) or wait for some time and then hang-up (renege) having obtained neither the information nor the booking they require. Meanwhile, the booking clerks are regularly lambasted by irate customers who have waited up to 15 minutes or more for a response. The cinema's managers are obviously concerned about the loss of good will and custom.

Wardeon Nottingham has decided to purchase and install a more modern telephone system. They have obtained various quotes for the system, the favoured choice being from a local company, Dodgey Phones Ltd. The proposed system uses digital technology, allowing customers to use their telephone keypads to choose from an automated menu system. As a result, it is believed that customers will be able to obtain the information they require from the recorded message service more rapidly. Further to this, the system will also provide facilities for booking tickets via the telephone keypad, without the need to speak to a booking clerk. Because some members of the public still wish to speak to a member of staff, particularly if they require further information or would rather book tickets in person, then there will be an option to speak with a customer service representative (CSR). Dodgey have also suggested increasing the capacity of the system from its present level and playing soothing music to customers should they have to wait for a response.

Figure A2.2 shows a schematic of the system proposed by Dodgey. Calls enter the system via a call router (or automatic call distributor, ACD), where the callers enter a number 1, 2 or 3 on their keypad for the service they require (information, ticket sales or CSR respectively). Calls are then routed to the appropriate service. Having completed a call with one of the automated services, callers will be given the option to return to the call router to select another service. The plan is to install four lines for each of the services including the call router. It is unlikely, however, that Wardeon will employ four CSR staff; the current intention is to employ three staff during the busiest periods. Wardeon have the option to purchase extra lines for each service, in blocks of two, albeit at quite some additional expense.

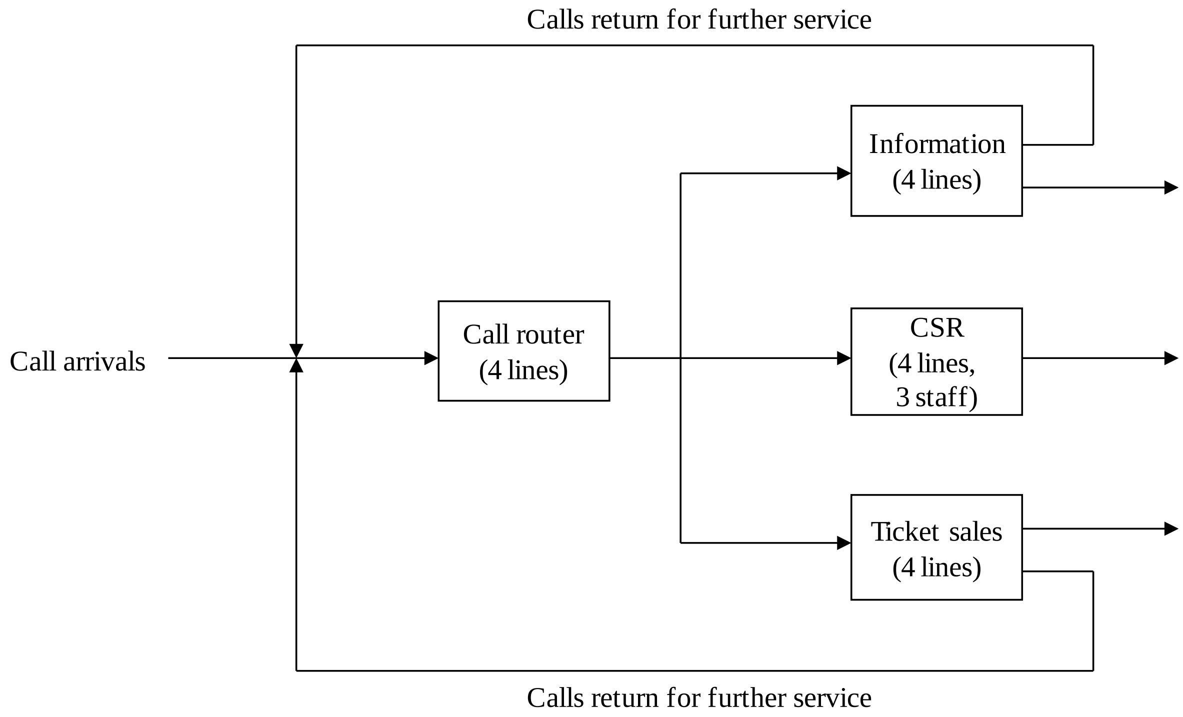


Figure A2.2: *Wardeon Cinema New Telephone System*

Wardeon's aim is to resource the system sufficiently so that less than five percent of calls are lost (balk or renege), even on busy days. They also want the total waiting time; that is the sum of waiting times in all queues, to be less than two minutes for at least 80 percent of calls and the mean to be less than one minute. There are a number of concerns, however, with the system that is being proposed (apart from the name of the supplier!), in particular:

- Whether there are sufficient information lines.
- Whether there are sufficient ticket sales lines.

Wardeon are quite happy with the current plan to have four call router lines because of the short time it takes calls to be handled by this part of the system. They would also like to know how many CSR staff to have available at different times of the day.

In the future Wardeon are planning to expand the Nottingham cinema complex. As a result, it is expected that the demand on the telephone system will continue to grow. They would, therefore, like to know the maximum capacity of the telephone system so they are able to identify the demand level at which further expansion will be needed.

The management board are meeting in three weeks and need to make a decision as to whether to go ahead with the Dodgey proposal, or not. The intention is to have the results of the simulation ready for that meeting. It should be noted that a number of the managers at Wardeon are sceptical concerning the need for a simulation. They have never used it before and will need convincing that the tool is useful"

Robinson (2014)

Test Case 3: King’s Road Nursery problem

“King’s Road is a pleasant residential street in the commuter town of Warentry in Central England. In Warentry there are many working families with young children and as a result there is a growing need for nursery places. Kiddycare, specialists in preschool nurseries, have identified this need along with a possible site for a new nursery in King’s Road. The residents of King’s Road are concerned about the impact the nursery will have on their lives and have decided to challenge Kiddycare’s planning application.

There are a number of grounds for the planning challenge, but the one of particular interest here is the impact of the additional traffic and parking requirements on life in the street. There are 63 houses in King’s Road and the adjoining King’s Meadow. Figure 1 shows a schematic of the two roads. King’s Road is also sandwiched between two busy trunk roads (Highland Road and Lowland Road). Throughout the day traffic uses King’s Road to cut through from one trunk road to another. This does cause a build-up of traffic, especially during busy commuter times; just the time that traffic will be arriving to drop-off and collect children from the proposed nursery. The problem is expected to be most acute in the morning rush, between 7.30am and 9.00am, as the children from the nursery will generally be collected in the middle of the afternoon before the evening rush begins.

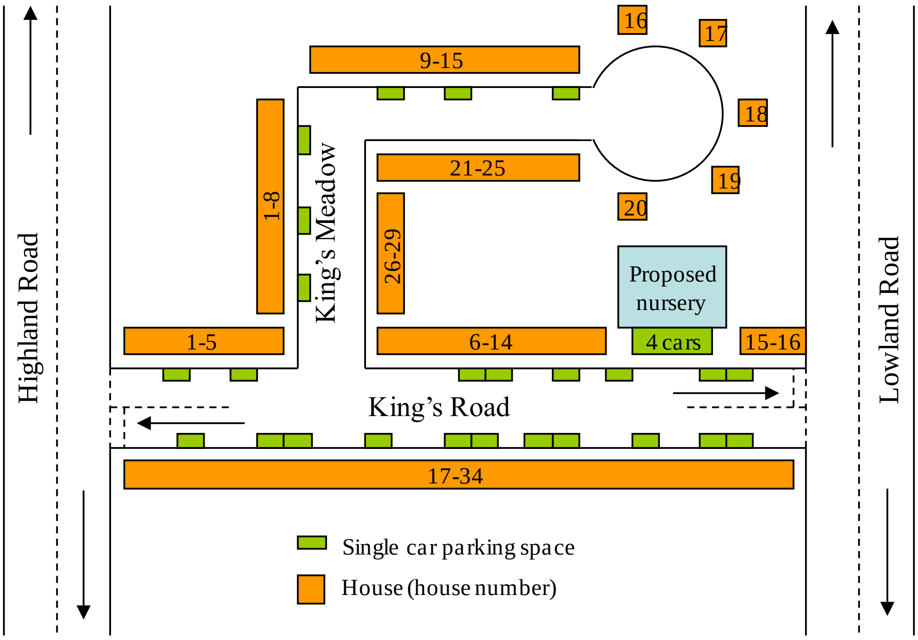


Figure A2.3 King’s Road and the Proposed Nursery

The location of the houses in King's Road and King's Meadow is shown in figure A2.3 as well as the proposed nursery. Most of the houses have off-road parking for at least one car. There are then a limited number of parking spaces on the street (also shown in figure A2.3). Since the residents own anything from one to up to three or four cars, most of these spaces are taken-up overnight and only vacated as the residents make their way to work in the morning. Meanwhile, the nursery plans to provide four off-road car park spaces, but these will only be sufficient for the nursery staff. It is expected that parents dropping their children at the nursery will spend anything from 1 to 10 minutes parked in the road, depending on whether they simply drop their children at the nursery or if they stop to talk with the staff or other parents.

The residents have collected data on car ownership, the times that they leave for work, and the time it takes to turn out into Highland and Lowland Road. At present the exact size of the nursery is uncertain, but the planning application suggests it will be for between 30 and 60 children.

There are two specific concerns for the residents of King's Road (and King's Meadow):

1. Will there be sufficient parking spaces for the residents and the nursery parents and staff?
2. What will be the effect on the traffic flow through the street, especially the queues that develop to turn out of the street onto the Highland and Lowland Roads?

The residents are due to present their initial case to the planning authorities in 7 days time. For this meeting they would like to have some data on the extent of the problem in relation to the two concerns above. They are hoping that you will be able to develop and use a simulation to help generate some data in support of their case"

(www.scs.org/magazines/201007/index_file/ConceptualModelingCorner.htm), accessed Feb 2014)

Appendix 3: Reference Solutions for SADT CM test cases

Test Case 1: Panorama Televisions

Entity Table

Entity	**Label**
TV	E1
Pallet	E2
Completed TV	E3

Table A3.1: Entity table for Panorama TV case

Activity Table

Activity	Control	Resource	Output Entity
TV arrives into system Sc(OP9)	Production Schedule (C1)	Forklift truck (T1)	E1
Boxes loaded to pallet (OP10)	n/a	Shift operator (R1)	E1/E2
LCD assembly (OP20)	n/a	Maintenance operator (M1)	E1/E2
Circuit Board Assembly (OP30)	n/a	Maintenance operator(M1)	E1/E2
Electrical Assembly (OP40)	n/a	Group of shift operators (R2-5)	E1/E2
Test area one (T1)	n/a	Maintenance operator (M1)	E1/E2 E1/E2
Rework area one (RT1)	n/a	Shift operator (R6)	E1/E2
Back Assembly (OP50)	n/a	Maintenance operator (M1)	E2/E3
Finished TV Unloaded from line (OP60)	n/a	Shift operator (R7)	E3 E2
Test Area Two (T2)	n/a	Maintenance operator (M1)	E3 E3
Rework area two (RT2)	n/a	Shift operator (R8)	E3
Packaging (OP70)	n/a	Shift operator (R9)	E3
Finished TV Transported to Warehouse (OP80)	n/a	Forklift truck (T2)	E3
Product exit form system SK(OP90)	n/a	n/a	n/a

Table A3.2: Activity table for Panorama TV case

SADT CM Branches

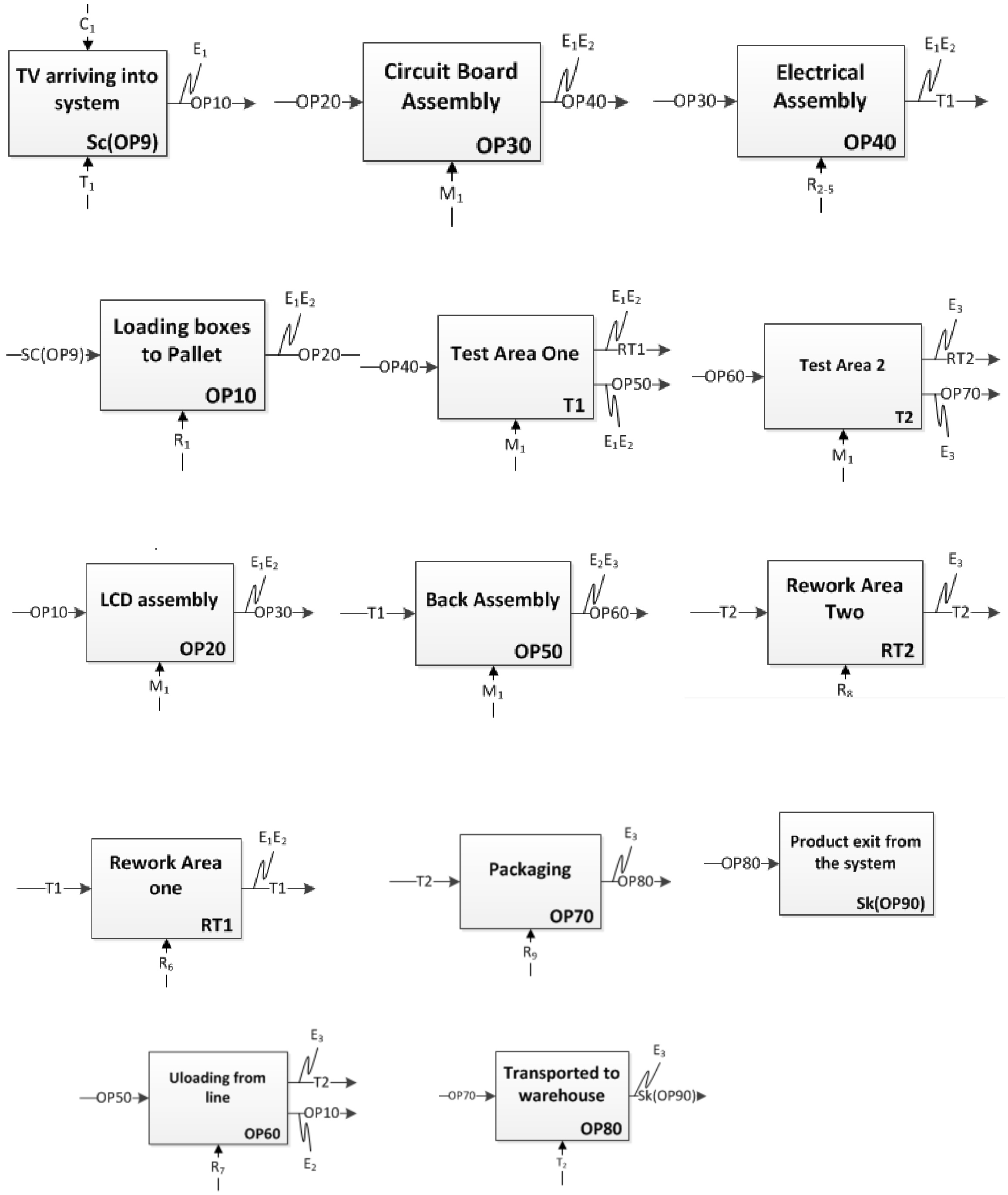


Figure A3.1: SADT CM branches for Panorama TV Case

Complete System Description

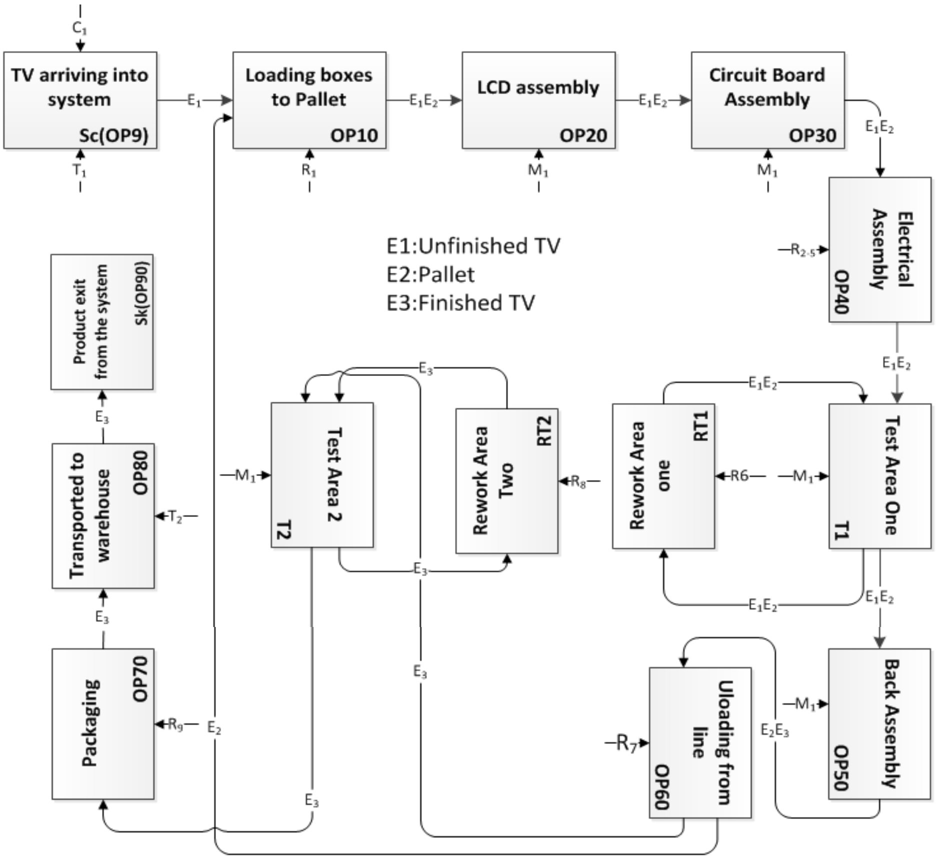


Figure A3.2: Complete SD from SADT CM branches for Panorama TV Case

Modelling objective

The modelling objective for Panorama TV case is identified as follows:

“To determine whether 500 units of throughput per day can be achieved”

Experimental Factor

The two experimental factor identified in Panorama TV case are as follows:

The number of pellets

The size of the buffers (conveyers) between the operations

Objective Activity (OA)

The OA in the Panorama TV case is identified as OP60. The reason for identifying OP60 as OA is because it is linked most closely to the number of finished TVs produced in a day. Also it is the last activity which determines the throughput of finished TVs as before this activity TVs can fail the testing procedure and could be returned.

Abstracted layout of Conceptual Model

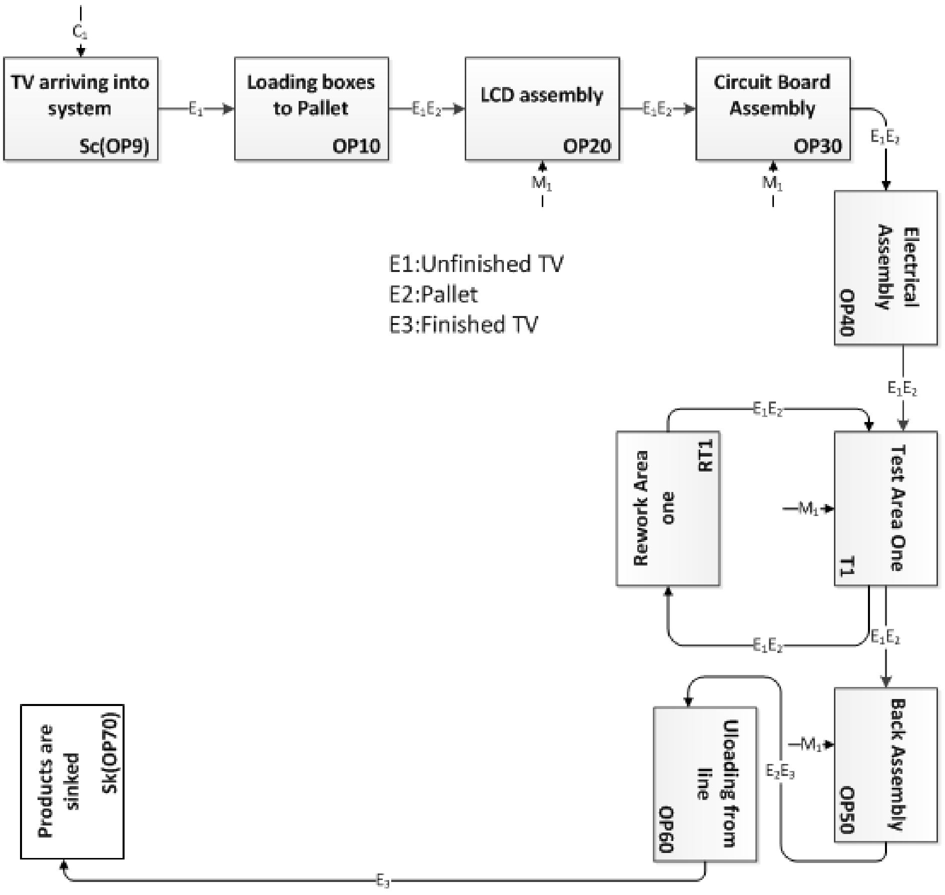


Figure A3.3: Abstracted layout of conceptual model for Panorama TV case

Scope Table

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
TV arrives into system Sc(OP9)	Production Schedule (C1)	Forklift truck (T1)	E1	YES	YES	Initiates arrival of entities in the system	E1- Yes	Main entity initializing the flow of product	T1- No	Assuming materials are always avaialble	C1-Yes	Determines production schedule	No
Boxes loaded to pallet (OP10)	n/a	Shift operator (R1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R1-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
LCD assembly (OP20)	n/a	Maintenance operator (M1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Circuit Board Assembly (OP30)	n/a	Maintenance operator(M1)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Electrical Assembly (OP40)	n/a	Group of shift operators (R2-5)	E1E2	YES	YES	Effects flow of entities in OA	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R2-5-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Test area one (T1)	n/a	Maintenance operator (M1)	E1E2 E1E2	YES	YES	contributes in the throughput of the final tested TVs	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Rework area one (RT1)	n/a	Shift operator (R6)	E1E2	YES	YES	contributes in the throughput of the final tested TVs	E1E2- Yes	E2 are experimental factors. E1E2 represnts TV in production	R6-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Back Assembly (OP50)	n/a	Maintenance operator (M1)	E2E3	YES	YES	directly contrbutes to entities in OA	E2E3- Yes	E2 are experimental factors. E2E3 represnts TV in production	M1-Yes	Shared Resource	n/a	n/a	YES (single)- Experimental factor
Finished TV Unloaded from line (OP60)	n/a	Shift operator (R7)	E3 E2	YES	YES	OA	E3- Yes E2 - NO	E3 finished Tvs (determines thorughput) E2 returned conveyers not modelled	R7-No	Dedicated Resource	n/a	n/a	YES (single)- Experimental factor
Test Area Two (T2)	n/a	Maintenance operator (M1)	E3 E3	NO	No	Does not effect OA	E3-No	Path not modelled	M1-No	path not modelled	n/a	n/a	No (path not modelled)
Rework area two (RT2)	n/a	Shift operator (R8)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	R8-No	Dedicated Resource	n/a	n/a	No (path not modelled)
Packaging (OP70)	n/a	Shift operator (R9)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	R9-No	Dedicated Resource	n/a	n/a	No (path not modelled)
Finished TV Transported to Warehouse (OP80)	n/a	Forklift truck (T2)	E3	NO	No	Does not effect OA	E3-No	Path not modelled	T2-No	Does not effect throughput of finshed TVs	n/a	n/a	No (path not modelled)
Product exit form system SK(OP90)	n/a	n/a	n/a	NO	No	Does not effect OA	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Table A3.3: Scope table for Panorama TV case

Test Case 2: Wardeon cinema problem

Entity Table

Entity	Labels
Information line calls	E1
CSR line calls	E2
Sales line Calls	E3

Table A3.4: Entity Table for Wardeon Cinema problem

Activity Table

Activity	Control	Resource	Output Entity
Call Arrivals (Sc(A1))	Customer's choice (C1)	N/A	E1/E2/E3
Call Router(A2)	N/A	N/A	E1 E2 E3
Information Lines (A3)	N/A	Three lines (R1)	E1
CSR lines (A4)	N/A	Three staff (R2) Three lines (R3)	E2
Ticket sales lines (A5)	N/A	Three lines (R4)	E3
Calls Exit (Sk(A6))	N/A	N/A	N/A

Table A3.5: Activity Table for Wardeon Cinema problem

SADT CM branches

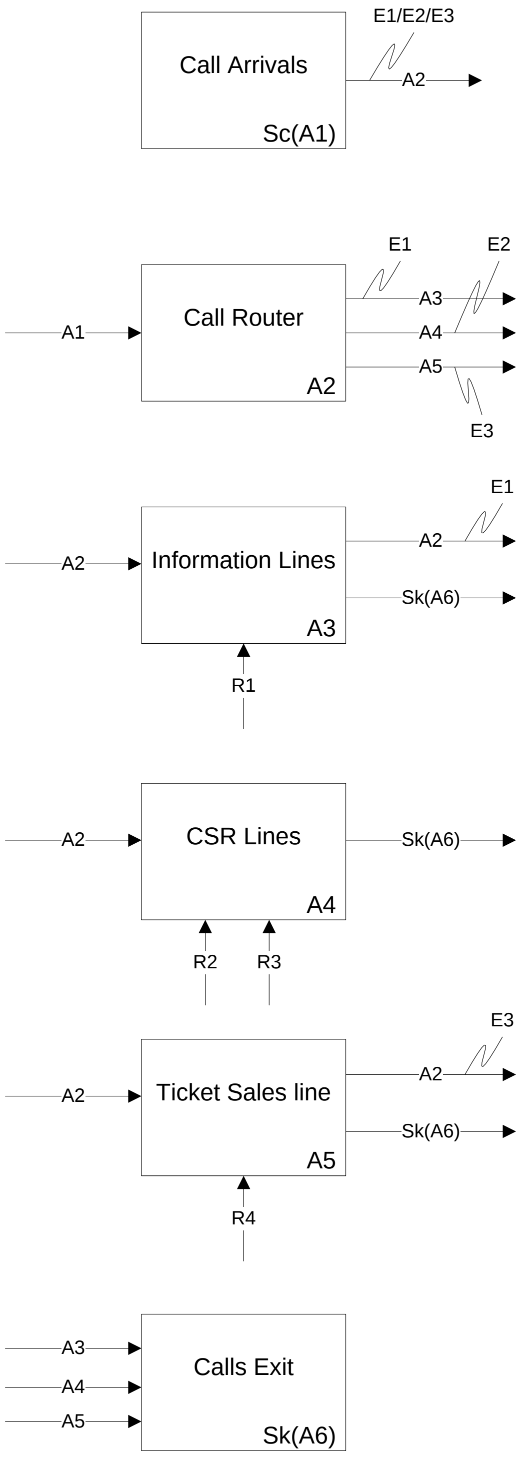


Figure A3.4: SADT CM branches for Wardeon Cinema problem

Complete System Description

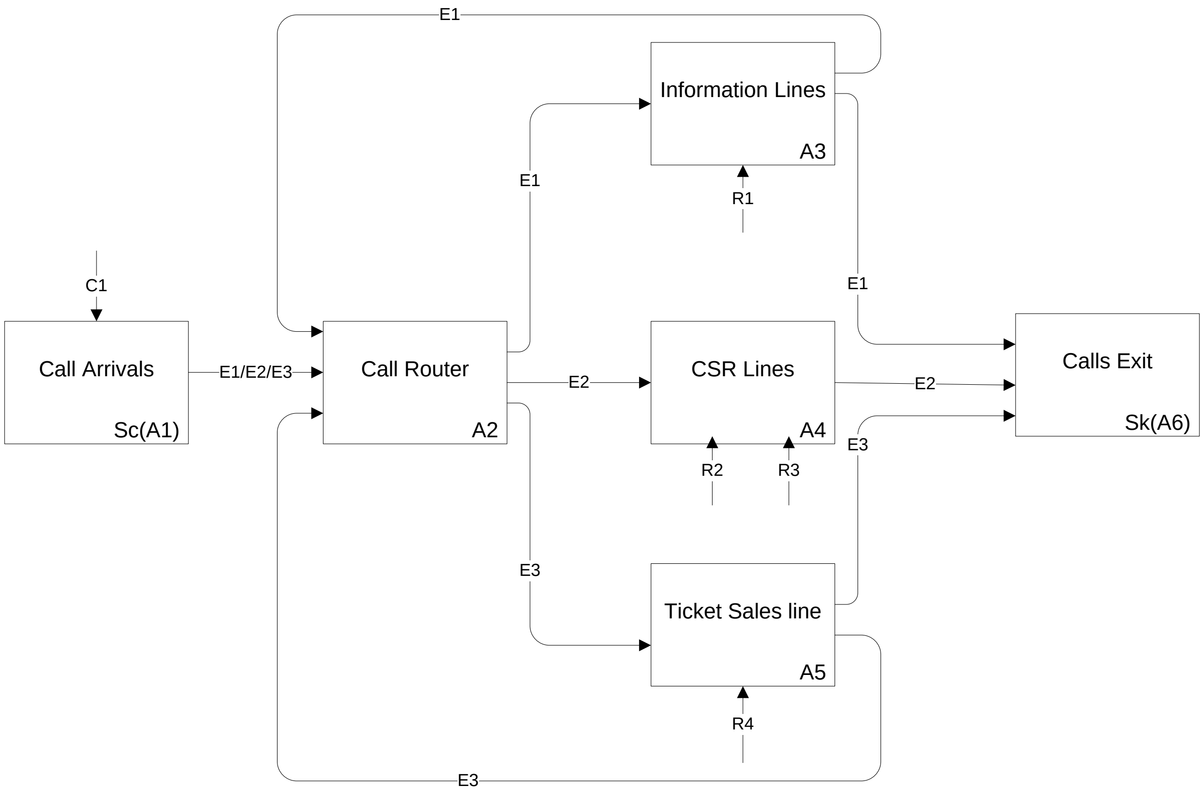


Figure A3.5: Complete system description for Wardeon Cinema problem

Modelling Objective, experimental factors and Objective Activity (OA)

Modelling Objective: To determine the capacity of information lines for achieving the required target

Experimental factor: "R1" as this represents the resource capacity of the information lines

Objective Activity: "A3" (information lines), as this is the main activity whose capacity needs to be determined

Scope table

The scope table is shown in Table A3.6

Abstracted layout of conceptual model

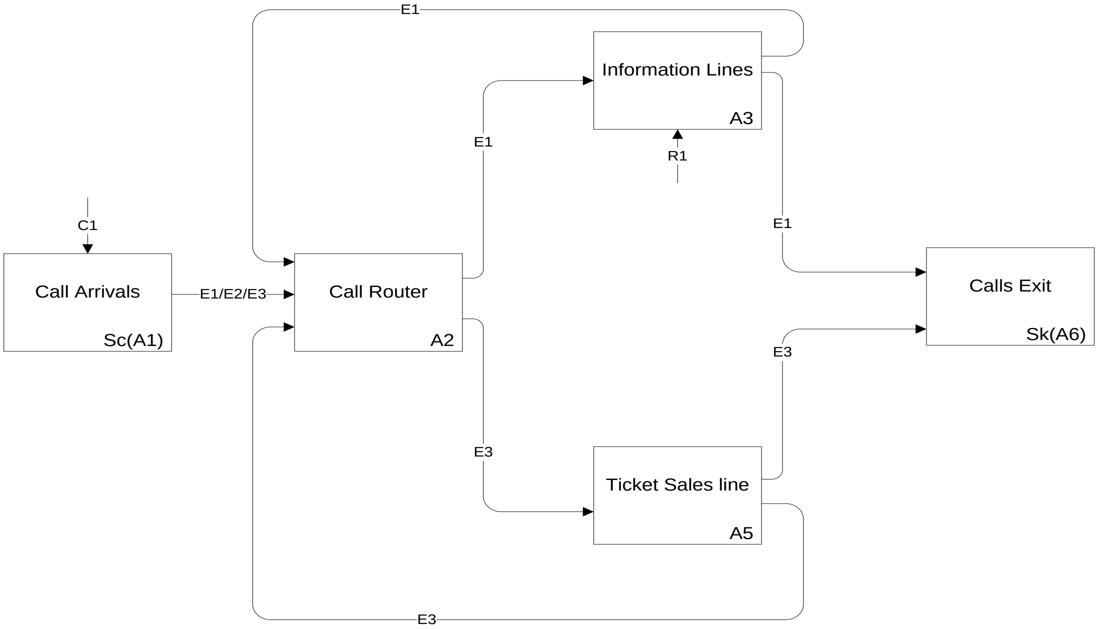


Figure A3.6: Abstracted layout of the conceptual model

Scope Table

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
Call Arrivals (Sc(A1))	Customer's choice (C1)	N/A	E1/E2/E3	Yes	Yes	Entry point of calls	Yes	Determines entities into OA	Yes	Detemrines flwo of entity into OA	N/A	N/A	No
Call Router(A2)	N/A	N/A	E1 E2 E3	Yes	Yes	Routes entities into OA	Yes but E2	E3 enters into A4 which is not modelled	N/A	N/A	N/A	N/A	Yes (single)
Information Lines (A3)	N/A	Three lines (R1)	E1	Yes	Yes	Objective Activity (OA)	Yes	Determines entities into OA	Yes	Experimental Factor	N/A	N/A	Yes (single)
CSR lines (A4)	N/A	Three staff (R2) Three lines (R3)	E2 E2	No	No	Does not affect OA	No	A4 not modelled	No	A4 not modelled	N/A	N/A	No
Ticket sales lines (A5)	N/A	Three lines (R4)	E3	Yes	Yes	Calls routed back may have an effect on OA	Yes	Call returned path is modelled	No	Dedicated operators	N/A	N/A	Yes (single)
Calls Exit (Sk(A6))	N/A	N/A	N/A	Yes	Yes	Determines exit point for the served calls	No	Exit point	N/A	N/A	N/A	N/A	No

Table A3.6: Scope table for Wardeon Cinema problem

Test Case 3: King’s Road Nursery

Entity Table

Entity	Labels
Resident cars	E1
Parent cars	E2
General pass-through cars	E3

Table A3.7: Entity Table for KRN problem

Activity Table

Activity	Control	Resource
Entering Highland Road (Sc(A1))	N/A	N/A
Entering Lowland Road (Sc(A2))	N/A	N/A
Driving on Highland road (A3)	N/A	N/A
Turning from Highland Road to Kings Road (A4)	N/A	N/A
Turning from Kings Road to Highland Road (A5)	N/A	N/A
Driving on Lowland Road (A6)	N/A	N/A
Turning from Lowland Road to Kings Road (A7)	N/A	N/A
Turning from Kings Road to Lowland Road (A8)	N/A	N/A
Driving on Kings Road (A9)	N/A	N/A
Turning from Kings Road to Kings Meadow (A10)	N/A	N/A
Turning from Kings meadow to Kings Road (A11)	N/A	N/A
Parked on Kings Meadow (A12)	N/A	N/A
Parked on Kings Road (A13)	N/A	N/A
Parked at nursery (A14)	N/A	N/A
Driving on Kings Meadow (A15)	N/A	N/A
Exiting Highland Road (Sk(A16))	N/A	N/A
Exiting Lowland Road (Sk(A17))	N/A	N/A

Table A3.8: Activity Table for KTN problem

SADT CM Branches

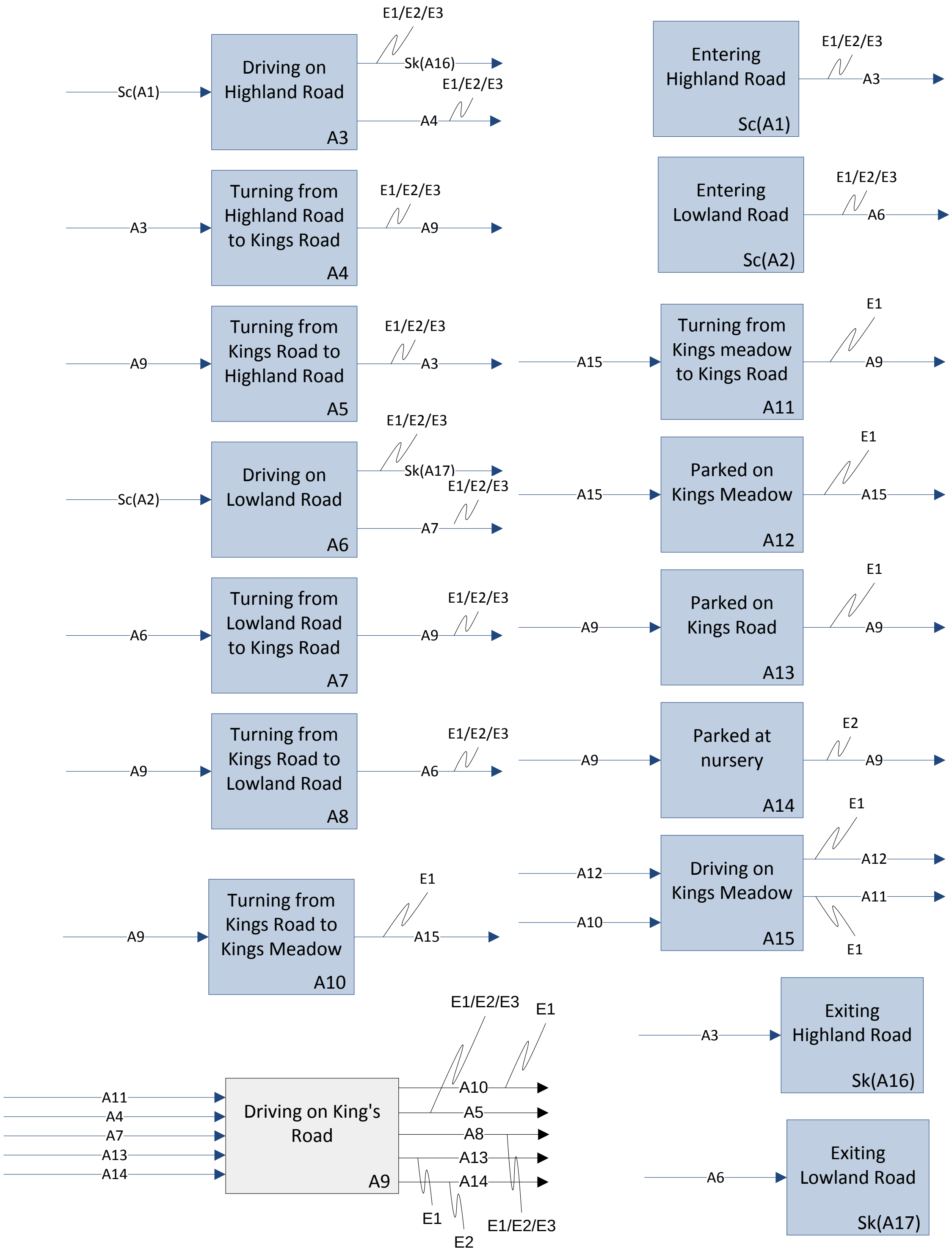


Figure A3.7 SADT CM Branches for KRN Problem

Complete System Description

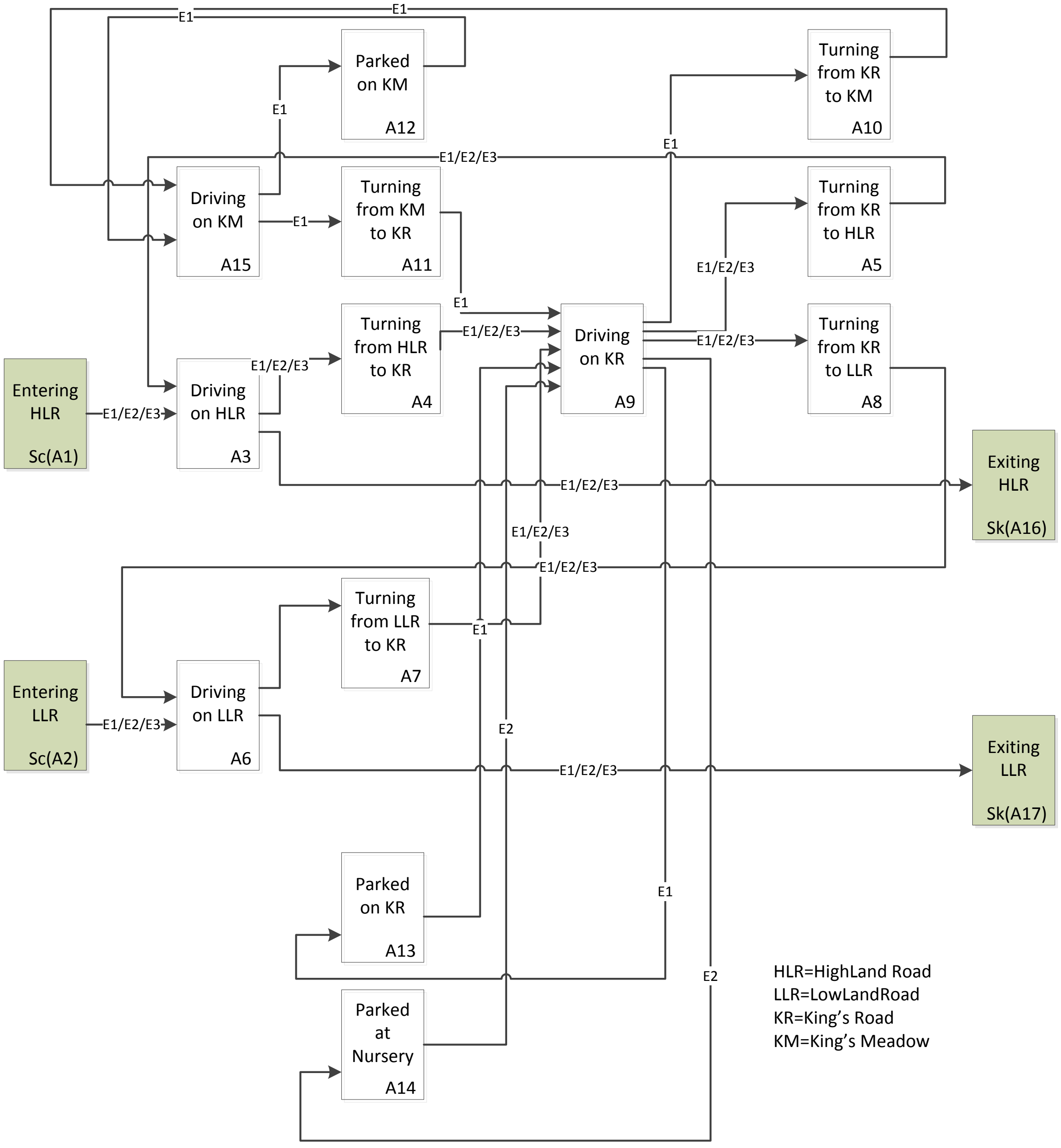


Figure A3.8: Complete system Description for KRN Problem

Modelling Objective, experimental factors and objective activity

Modelling objective: To study the effect of traffic flow on King's Road, in particular by determining the waiting time for the queue developing to exit onto Highland Road.
Experimental factor: Nursery parking capacity
Objective activity: Cars turning to exit onto Highland Road (A5)

Scope table

The complete scope table form participant C is shown in table A3.9

Abstracted Layout of Conceptual Model

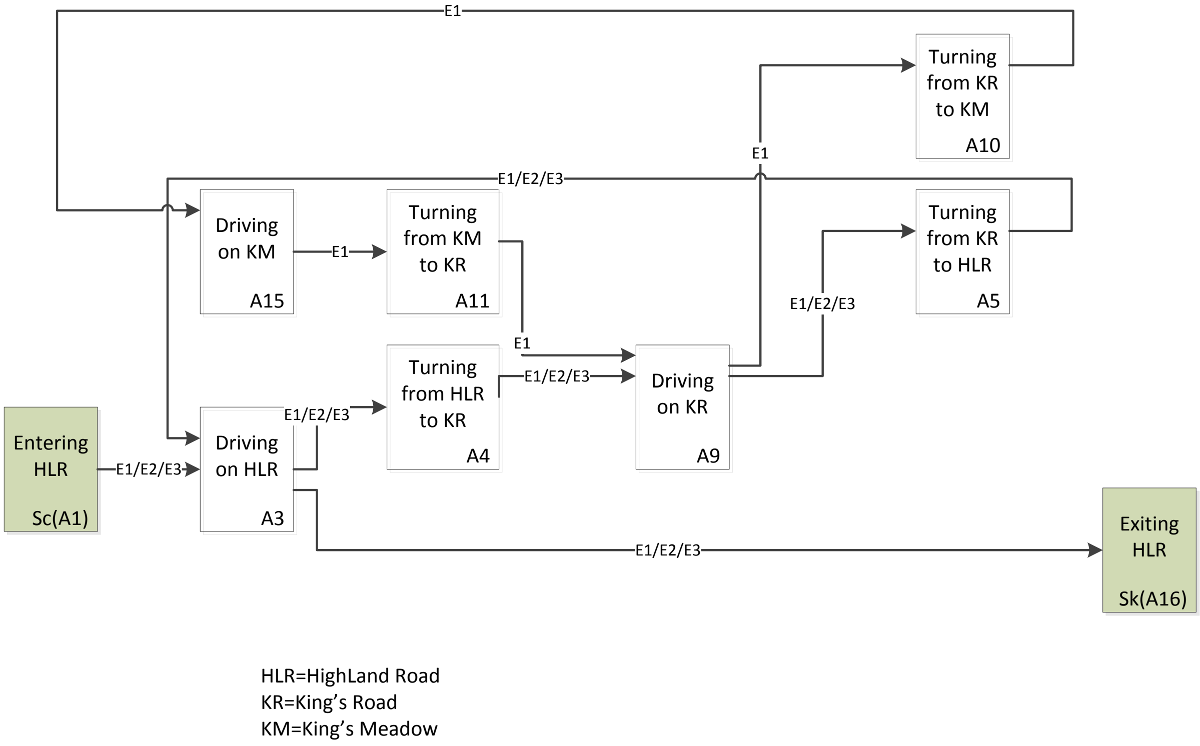


Figure A3.9: Abstracted Layout of Conceptual Model for KRN Problem

Activity	Control	Resource	Output Entity	CA impacts OA ?	Decision to model	Justification	Entity modelling	Justification	Resource modelling	Justification	Control Modelling	Justification	Queue Required
Entering Highland Road (Sc(A1))	N/A	N/A	E1/E2/E3	Yes	Yes	Entry Point	Yes	Entry point for entities	N/A	N/A	N/A	N/A	No (Entry Point)
Entering Lowland Road (Sc(A2))	N/A	N/A	E1/E2/E3	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Driving on Highland road (A3)	N/A	N/A	E1/E2/E3	Yes	Yes	Detemrines traffic on HLR	Yes	Determines entities into OA	N/A	N/A	N/A	N/A	Yes
Turning from Highland Road to Kings Road (A4)	N/A	N/A	E1/E2/E3	Yes	Yes	Detemrines traffic on KR	Yes	Determines entities into OA	N/A	N/A	N/A	N/A	Yes
Turning from Kings Road to Highland Road (A5)	N/A	N/A	E1/E2/E3	Yes	Yes	Objective Activity (OA)	Yes	Impacts entities flow from OA	N/A	N/A	N/A	N/A	Yes
Driving on Lowland Road (A6)	N/A	N/A	E1/E2/E3	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Turning from Lowland Road to Kings Road (A7)	N/A	N/A	E1/E2/E3	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Turning from Kings Road to Lowland Road (A8)	N/A	N/A	E1/E2/E3	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Driving on Kings Road (A9)	N/A	N/A	E1/E2/E3	Yes	Yes	Main driving road for exit lanes	Yes	impacts flow of entities into OA	N/A	N/A	N/A	N/A	Yes
Turning from Kings Road to Kings Meadow (A10)	N/A	N/A	E1	Yes	Yes	Impacts traffic onto KR	Yes	May effect entities into OA	N/A	N/A	N/A	N/A	Yes
Turning from Kings meadow to Kings Road (A11)	N/A	N/A	E1	Yes	Yes	Impacts traffic onto KR	Yes	May effect entities into OA	N/A	N/A	N/A	N/A	Yes
Parked on Kings Meadow (A12)	N/A	N/A	E1	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Parked on Kings Road (A13)	N/A	N/A	E1	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Parked at nursery (A14)	N/A	N/A	E2	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Driving on Kings Meadow (A15)	N/A	N/A	E1	Yes	Yes	Impact traffic onto KR	Yes	May effect entities into OA	N/A	N/A	N/A	N/A	Yes
Exiting Highland Road (Sk(A16))	N/A	N/A	E1/E2/E3	Yes	Yes	Exit point for OA	Yes	Delays in exit can effect OA entities	N/A	N/A	N/A	N/A	No (Exit Point)
Exiting Lowland Road (Sk(A17))	N/A	N/A	E1/E2/E3	NO	No	Does not effect the OA	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table A3.9: Scope Table for KRN problem

Appendix 4: Glossary of terms

SRE	Software Requirements Engineering
SDLC	Software Development Life Cycle
UOD	Universe Of Discourse
SE	Software Engineering
IS	Information Systems
DB	DataBases
CS	Computer Science
SADT	Structured Analysis and Design Technique
SADT CM	Structured Analysis and Design Technique Conceptual Modelling
M&S	Modelling and Simulation
ER	Entity_Relationship
M&S CM	Modelling and Simulation Conceptual Modelling
HDC	Happyfaces Day Care
SD	System Description
CM	Conceptual Modelling