Graphical Documentation to Aid Simulation Studies of Manufacturing

Ronan Campbell, BEng

Graphical Documentation to Aid Simulation Studies of Manufacturing

Ronan Campbell, BEng

A thesis submitted in fulfilment of the requirement for the degree of

Masters of Engineering

Supervisor:

Dr. Paul Young

School of Mechanical and Manufacturing Engineering,

Dublin City University, Ireland

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of MEng is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: I.D. Number: 53387127

Date:

Acknowledgements

I would like to thank all the people that have helped and guided me throughout the course of my research.

I would like to thank, wholeheartedly, my supervisor, Dr. Paul Young. His guidance and knowledge was an invaluable resource to me throughout my postgraduate studies. I would also like to thank Dr. John Geraghty who lent his assistance and experience during my research. I want to thank Mr. Keith Hickey for the technical and IT support he provided at times when my laptops were grinding to a halt. I'd like to extend huge thanks to Ms. Suzanne Dockery who was always there to lend a hand and point me in the right direction. Furthermore all the members of staff in the Mechanical and Manufacturing Faculty that have aided and thought me during my time in DCU, I would like to thank for helping me get to the point I am at today.

I would like to thank all involved in the KSETS project conducted between Dublin City University and University of Limerick. All the project members were extremely helpful when I sought their input and what I learned greatly impacted the direction my own research took.

I am more than grateful to Enterprise Ireland who funded me and gave me the opportunity to further my education through research.

I would like to thank my fellow students and researches from the Mechanical and Manufacturing Faculty in Dublin City University, namely Ramzi Salman, Neil Byrne and Emmet Galvin for all the time spent in learning from one another.

Last but not least I want to extend the most sincere thanks to my family and my friends, especially my wonderful girlfriend Isabel who has supported me greatly during this task. Together they have helped me monumentally not only over the past years but in all aspects of my life so I could complete my education to date.

Thank you.

Abstract

Computer processing power has developed to the stage where simulation has become an extremely popular and applicable way of representing real world systems for investigation. For the most part simulation studies as a whole can be long complex projects. Through-out the simulation industry there is a common consensus from available literature that certain steps should be followed to create a credible, successful simulation. While this is well known it appears that these guidelines are more “Do what I say, not as I do,” within the simulation community. In the experience of the author of this project simulation teams and modellers approach their own simulation studies in their own different ways, yet when the simulation study reaches its conclusion a credible simulation still has to be presented to a client or relevant party. For example a simulation modeller may often spend the most time and in turn resources on building the simulation model, yet this model will be next to useless without any documentation relating to the validation of said model. With good documentation being applied through-out a study, to each specific step, it only serves to make the succeeding steps easier to implement. This project highlights and uses the Systems Modelling Language (SysML) as a tool and method to develop diagrams to aid either the team or the modeller. These diagrams can be used as references when adhering to the steps of creating a credible, successful simulation study as well as a graphical support when presenting the entire simulation study to the client or relevant parties involved. This thesis also covers an independent assessment of the generated SYSML diagrams. Importantly when developing a method such as using SYSML as a graphical aid for simulation studies it must be reviewed by interested parties so that the areas that work well can be highlighted as well any areas which lack or need developing.

Table of Contents

List of Figures

List of Tables

1 Introduction

The word manufacture is derived from the Latin, manus and factus, the literal translation meaning "Made by hand" [1]. Manufacturing has existed since the first human invented materials and processes to create materials. Since the industrial revolution approximately 200 years ago the manufacturing industry has increased immensely, especially with the introduction of engines and machine tools. These paved the way for factory systems to be created. The industrial revolution saw a huge change in the economy from agriculture and handicraft to an economy driven by both industry and manufacturing. Manufacturing industries account for about 20% of Gross National Product in the United States [1]. Throughout human civilization manufacturing played a pivotal role whether it was creating vaccines to combat illness or a nation's ability to manufacture better tools/weapons in times of war.

From the shoes worn, to cars driven to mobile phones to make calls on, technology's effect on the modern world can be seen everywhere. Kevin Kelly spoke at length in a 2005 presentation on "How Technology evolves" and technology's impact worldwide. He theorized that technology is driven by the same factors that drive evolution in the six kingdoms of life (Fungi, Animalia, Protista, Eubacteria, Plantae, Archaebacteria), such as ubiquity, complexity, diversity and socialisation. He even goes as far to say that technology may be considered as a 7th kingdom [2]. Technology drives differences, diversity, options, choices, opportunities, possibilities and freedoms in today's society. Each and every piece of technology used in daily life is manufactured. Concurrently manufacturing and technology are inherently linked. Technologies are manufactured. By the aid of research and technology, manufacturing systems become more capable. Technologies further evolve and successively manufacturing abilities expand.

Economically manufacturing is the means to add value to raw/starting materials and have a processed piece exit the manufacturing system worth more. Technically the manufacturing process is an amalgamation of machinery, tooling, power and labour, the output of which is the processed product as well as scrap and waste [1]. These different definitions are depicted in Figure 1-1.

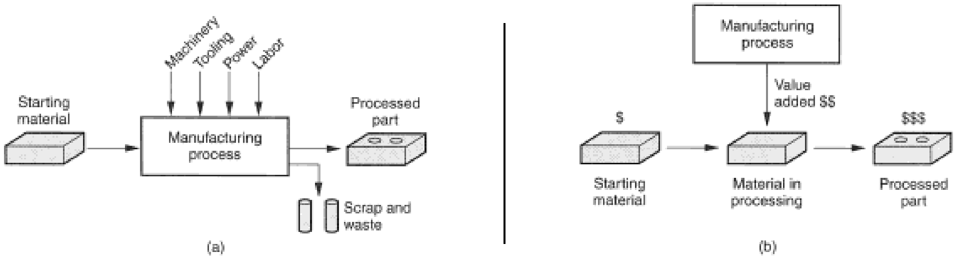


Figure 1-1 : Manufacturing Definition: (a) as a Technical process (b) as an Economic process [1]

It is in a manufacturer's best interest to improve their capacity, efficiency and quality of output. Improving these factors increases the wealth generated for the business unit. It is also in a manufacturer's best interest to operate their manufacturing systems to their utmost. Especially in the latter half of the 20^{th} century and to date, manufacturing facilities have tended towards 24 hours a day operation. Downtime in a facility that operates round the clock can lead to immense cost for the company. In addition to the fine line between operation and cost, competition drives a manufacturing company to use its technology and machinery in order to maximise productivity. Sequentially maximum productivity begets a greater capital gain.

The application of computer based simulation and modelling is a solution to how a manufacturing facility can improve their capacity, efficiency and quality. Simulation based experimentation on a model, which represents the real world system, saves the need for downtime on the real system. Model based simulation has become a pivotal tool in the manufacturing industry whereby a facility can enhance their processes and not suffer from traditional drawbacks of actual experimentation and alteration of those real world systems [3].

As simulation has become an incredibly useful asset in the manufacturing industry over the past 30 years simulation studies are becoming more common place. Model Based Systems Engineering (MBSE) can help manage complexity, while at the same time improve design quality and cycle time, improve communications and facilitate knowledge capture and design evolution [4].

The motivation behind this research came about from a manufacturing research group that concentrated on simulation. Across the different projects in progress during this research each had their own aims. Each member of the group had their own methods and means to complete their simulation studies respectively as well. Across the group there was also diversity in the number of programs that were being used to create and implement the simulation models. The research presented in this thesis evolved as a tool that could be used as a common grounding for all relevant members in the research group and as a means of communicating between them (and their respective projects). The research developed further into addressing communication (documentation) within simulation studies in all industries.

This thesis highlights that, even though simulation has vast applications within many industries, it has no specific standard for documenting the study. The thesis shows how and where documentation can play an important role. Several simulation studies were investigated and in each case how documentation was applied. Each case was evaluated and compared with the others to show where documentation tended to be strong and where it was weak. The thesis also highlights how SysML (Systems Modelling Language) can be used as an aid in documenting a simulation study and an example of SysML being applied is presented and discussed. This research delves into the development of a simulation study and how MBSE and requirements engineering can aid computer based simulation and modelling. There is no standard or specification currently on how a simulation study should be conducted or delivered. This research aims to address this and put shape to the cycle of a simulation study.

1.1 Project Aims and Overview

In 1977 Highland stated the prime purpose of his paper was to *"nudge those engaged in modelling and simulation into action in the development of some system of model documentation"*. At that time a comprehensive system of model documentation didn't exist [5]. 33 years on at the time of compiling this literature review, there are tools that can be used in conjunction with simulation models to aid documentation, however there is currently no standard of how to document or how to apply documentation to a study.

This Project aims to accomplish the following:

- Conduct a literature review of Simulation and documentation in the simulation industry
- Examine several simulation studies and assess the documentation methodology that was used to complete that simulation study.
- Use SysML to aid the documentation of a simulation study.
- Assess SysML as a tool for aiding simulation.

- Look at SysML as a tool to help accelerate productivity in simulation.
- Tend towards a methodology that can be employed to aid a simulation study and help in reducing the time taken for certain stages of a simulation project.
- Highlight the areas that "Good Documentation Practice" can be easily applied to with little or no extra effort.
- Consider how SysML and good documentation practice can help with reusability in the simulation industry.

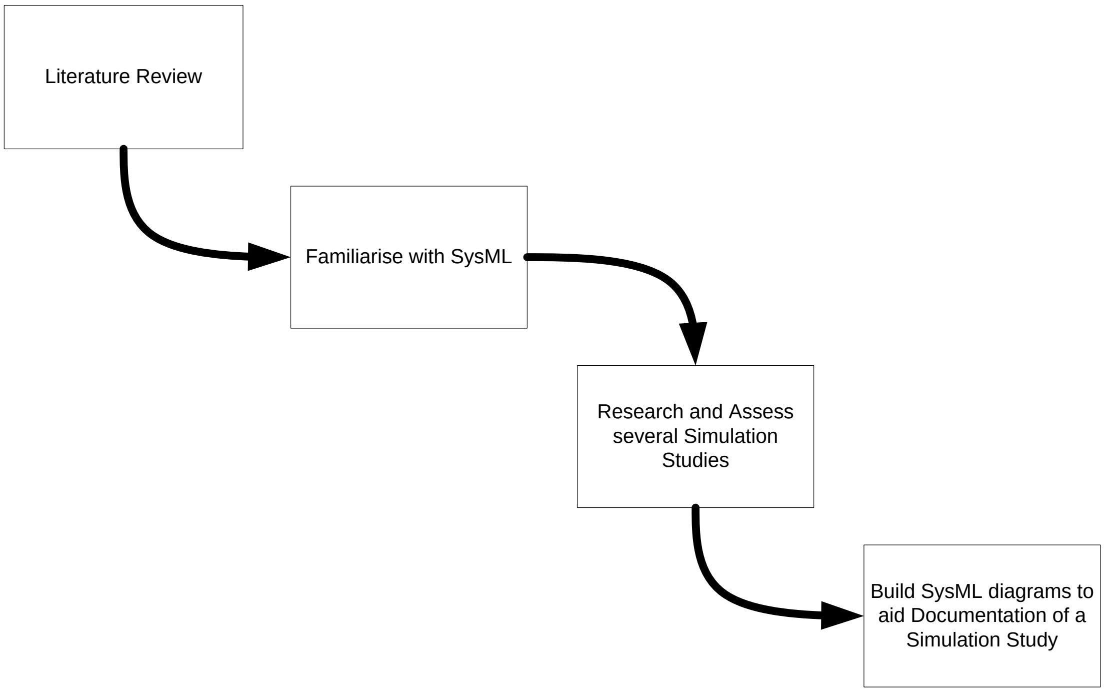


Figure 1-2: Flow of Project Approach

Figure 1-2 depicts the approach taken to this project. The research began with a Literature Review (Chapter 2) of the simulation industry and specifically the involved sub areas of manufacturing systems and the documentation of models. After several specifications/notations were investigated it was decided the research would involve SysML as a means to aid simulation. SysML contained a larger choice of diagrams to cover a wider area in aiding a simulation study than the relevant diagrams from other notations. No standard existed (at the time of writing) to explicitly state how a simulation study should be conducted. The research will look at the System Modelling Language as a tool to enhance the different areas that form a successful simulation study (Chapter 2). This was followed by familiarisation with SysML. Learning and familiarising oneself with SysML can be a somewhat lengthy process and required a lot of time, patience and discipline. The next major area was seeing how simulation studies are approached in the modern day by researching several case studies (Chapter 3). By assessing these case studies it was possible to gain a cross section on how simulation studies are conducted by several different people/teams. In addressing the same issues and areas of each case study it was possible to form an opinion on comparable subjects as well as seeing the contrasting pieces from each. From that the SysML diagrams were then created with the aim to highlight areas where they can aid a study. The created diagrams were then presented to 5 members of differing technical backgrounds (Chapter 4). With their feedback it was then possible to gain an understanding of how the simulation industry might react to SysML as a method to graphically aid the simulation industry/simulation studies. Unfortunately the created diagrams would need to be distributed on a much wider scale to gain a proper indication of how the entire simulation industry would perceive them. Importantly though it was seen that the individuals were able to understand the graphical aids with little or no prior knowledge of SysML or the included information. This has been addressed in section 4.2 and the following discussion chapter (Chapter 5).

2 Literature Review and SysML Introduction

2.1 Introduction

Chapter 2 introduces modelling, simulation and how to successfully complete and conduct a simulation study. A non-exhaustive list of some of the tools and methods available within the manufacturing industry to aid in simulation studies will be examined. The benefits of such tools and methods will be highlighted and considered. Documenting an entire simulation study and its importance will be assessed during this literature review. Reusability in relation to simulation will be examined as good documentation can aid in the reusability of different aspects in simulation study. Further into this chapter (section 2.10) an overview of the systems modelling language SysML and an application of it to the simulation industry will be provided by showing how to create SysML diagrams will also be looked at during this section after the literature review proper.

When considering engineering papers and published resources for this thesis, the author became familiar with simulation, modelling and documentation. To illustrate the quantity of papers available the Table 2-1 was compiled from the major keywords searched from several of the available databases at the time of writing. It tabulates the huge number of hits that occurred when using these particular keywords. For the purposes of this project and thesis the search terms had to be refined as this table was only an illustration of how wide the industry is. The papers considered were those that could be used to build a picture of the current ideas and methodologies employed in the various areas of simulation. Together all the literature sourced an understanding of the simulation industry and how MBSE could be used to guide a simulation study from start to finish as there is no current standard of specification stating how to conduct a simulation study.

Table 2-1: Simulation Keyword Related Papers Available

Database:	ACM Digital Library	Engineering Village	IEEE	JSTOR	Science Direct
Keywords:	Available Papers				
Simulation	602,043	2,400,079	323,834	4,466	771,370
Simulation Modelling / Simulation and Modelling	6,360	1,145,705	8,498 /128,648	740	356,495
Simulation Documentation /Simulation and Documentation	3,543	13,756	9 / 191	268	22,707

Simulation is a huge area of research. However there was a considerable drop off in the number of papers when researching simulation modelling / simulation and modelling and a further drop off relating to simulation documentation / simulation and documentation. The drop off in the keyword search between simulation and simulation modelling / simulation and modelling may be worth noting. In the manufacturing industry the words “simulation” and “modelling” are inherently related; this relationship is covered in section 2.4. Through the research conducted by this author it is seen that simulation, modelling and simulation modelling are often used interchangeably on the same topic between authors of different works.

In addition to talks, presentations and hard copy books, the search terms were refined to highlight good applicable work that directly relates to the project at hand. Table 2-2 below shows the number of papers (and other sources) reviewed over the course of the author’s two year research, the spread of time which they were researched and finally the number of useful sources directly used for this thesis.

Table 2-2: Citation Breakdown

	Number	Time
Abstracts Read	200+	
Works Fully Investigated	74	1977-present
Cited Works	49	

The following review of information begins at the base of simulation and the knowledge required to understand simulation and modelling, to create a good understanding of the industry with the aim of this thesis to have an impact upon it.

2.2 Systems

A system is a set of interrelated elements. These can be referred to as either elements or components [6] [7]. Each element is defined by attributes and generally the activities of these elements occur over time. Activities in a system cause changes to the system’s state. The performance of a system depends on each entity contained within that system and effects the systems output.

A system's structure can be viewed through the different configurations as in Figure 2-1.

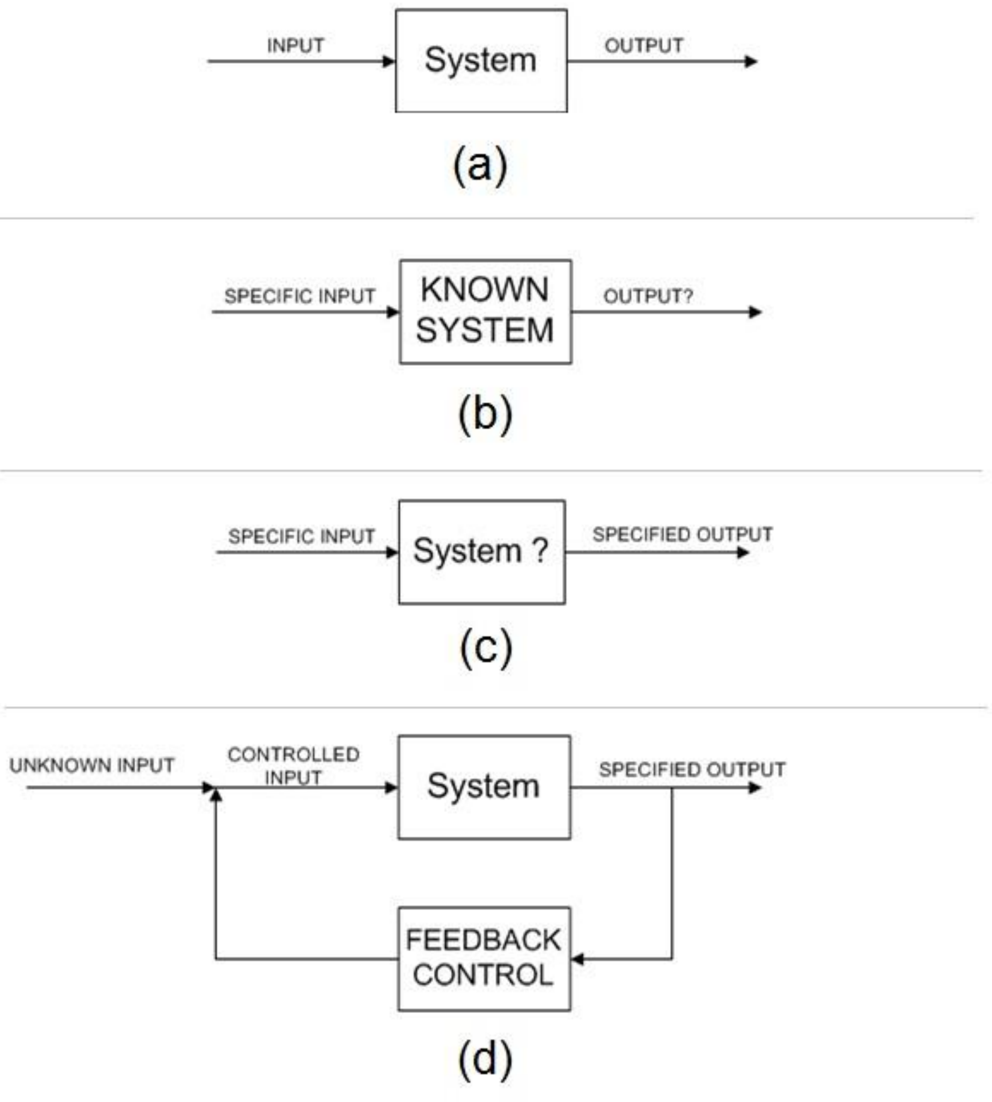


Figure 2-1: System Structure [7]

(a) Simple System Structure (b) Unknown System Output

(c) Unknown System structure (d) Unknown System Input

Figure 2-1 (a) shows the simplest of system structures where there is an input, the system itself (where the system's entities reside), and an output caused by the combination of inputs and internal workings of the system. Figure 2-1 (b) (c) and (d) depict the three main situations that occur when considering systems.

The first situation, Figure 2-1 (b), is when the output is unknown. In this case knowing the input and the specifics of the system the outputs can be predicted. The second situation, Figure 2-1 (c) occurs where the inputs and outputs are known while it is the system that is unknown. Sometimes this can be the case if a system has to be designed rather than investigated to yield specific outputs, through using specific inputs. A third situation happens when an input would need to be calculated to show that a specific output could be produced. For such a system setup Figure 2-1 (d) illustrates how the input into the known system must be controlled with the information fed back from the specified output.

Systems can be made up of one or more subsystems. Each of those subsystems can consist of one or more subsystems and each of those subsystems can be comprised of one or more and so on indefinitely. The parameters and variables that define the system(s) and or subsystem(s) are defined by the relationships between entities and other entities, and between entities and the system. Variables can be independent where they may be open to manipulation or dependent upon other factors, controllable or uncontrollable and continuous or discrete. Systems can be classified under a number of identities; natural or artificial, stable or unstable, dynamic or static, deterministic or stochastic, adaptive, non adaptive and linear or non-linear [7] [8].

Systems may act differently depending on their state [4]. The description of the elements at a certain point in time defines that system at that time, i.e. the state of that system. The variables related to each element are called state variables. In a binary system each element will have two states corresponding to 1 and 0. In a decimal system the elements will have ten possible states. A rise in the number of states will invariably lead to an increase in system complexity [7].

When taking systems into consideration each must have a clearly defined boundary. All systems operate within an environment. Changes that occur in the environment are outside the control of the system but may still affect the inner workings therefore altering the relevant outputs [7] [4].

Manufacturing facilities are systems. They have inputs, processes that alter the inputs and from the system come outputs. To investigate a manufacturing system on any level one must first describe the manufacturing system. In this research the simulation studies that will be assessed are all based on real world manufacturing systems.

2.3 Modelling

“A model is a simplified representation of a system (or process or theory) intended to enhance our ability to understand, predict, and possibly control the behaviour of the system” [7]. To explain the behaviour of a system one must understand that system. Whether a model is mental, physical or symbolic, the quality and quantity of the information in the model helps one to determine the outcome of an action on the real world system without having to alter the system in the real world. Maria [8] defines a model as “a representation of the construction and working of some system of interest”. Modelling expresses the associations that connect the inputs and outputs of the system of interest. Good modelling must be a trade off between realism and simplicity according to Rubinstein [6]. The simpler a model, the easier it is to understand, however a model must be an appropriate representation of the real world system upon which it is based. Adding excessive detail to a model may make a solution more difficult to obtain. What is required of a model is that a high-quality association between model predictions and the performance of the real world system exists. A model should only be considered as a representation of reality. It cannot have all the same characteristics and attributes of the real world system no matter how complex it is [6].

Symbolic models are more economically viable than building physical models [7]. This research is interested in Symbolic models, specifically mathematical models as the research is aiming towards aiding simulation and modelling in the manufacturing industry. Figure 2-2 followed by Figure 2-3 shows the various choices available when considering symbolic models and mathematical models respectively.

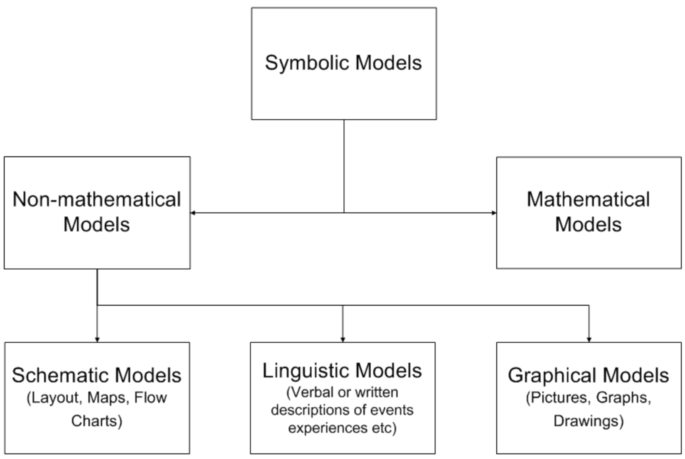


Figure 2-2 : Symbolic Model Taxonomy [7]

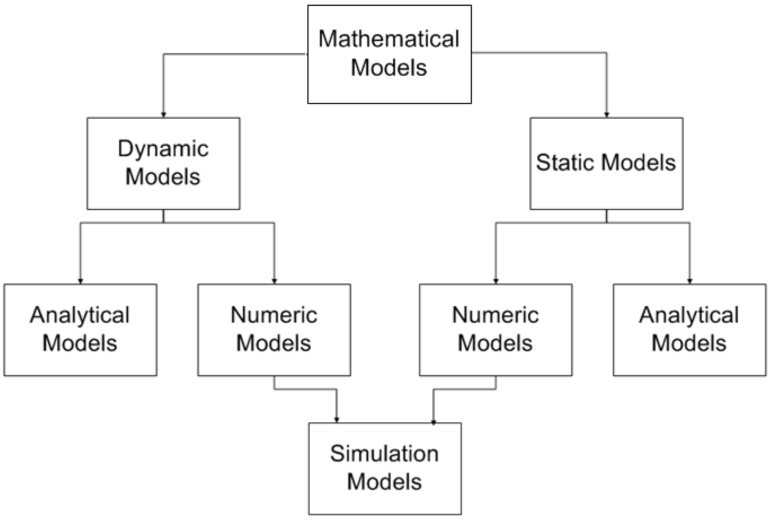


Figure 2-3 : Mathematical Model Taxonomy [7]

A mathematical model is defined as "a set of mathematical and logical relations between various system elements" [7]. "Mathematical models classifications include deterministic, (input and output variables are fixed values) or stochastic (at least one of the input or output variables is probabilistic); static (time is not taken into account) or dynamic (time-varying interactions among variables is taken into account)," [8], as well as it being a continuous, discrete or a hybrid model. Furthermore the model could be linear or non linear [7]. Choosing the appropriate type of mathematical model is determined by the characteristics of the real world system. A simulation model may be any of these and are explained in greater detail in section 2.4.

There are numerous different mathematical models that can be used to describe systems as seen in Figure 2-3 Manufacturing companies can use mathematical models to describe their systems and specifically use simulation as an aid to investigating their manufacturing process [7].

2.4 Simulation

Simulation is defined by Banks [3] as "the imitation of operation of a real-world process or system over time". Expanding on that definition further Maria classes simulation as the "operation of the model of the system" and additionally defines it as "a tool to evaluate the performance of a system, existing or proposed, under different configurations of interest and over long periods of real time" [8]. Similarly Shannon [9] sees simulation as "the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system”. By representing a system mathematically, simulation allows numerous questions be posed and answered without ever having to alter the real world system [3]. Importantly simulation can also be applied where a real world system doesn’t yet exist to likewise answer important questions. Figure 2-4 shows the interaction between a simulation study and the real world.

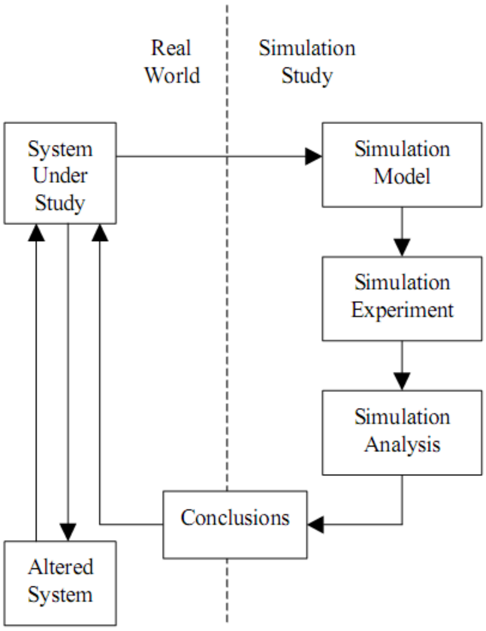


Figure 2-4 : Simulation Study [8]

2.4.1 Simulation Models

Simulation Modelling is a method of modelling a system mathematically so that the system in question can be altered or interrogated to calculate information that the user is interested in. According to Rubinstein [6] simulation models can be classified in several different ways as follows:

<u>Static and Dynamic Models</u>

Static models don’t represent a system through a passage of time nor do they evolve over time. Dynamic Models represent a system that changes and evolves with time.

<u>Deterministic and Stochastic Models</u>

A simulation model which only includes non random elements is known as a deterministic simulation model. A deterministic simulation model contains relationships (mathematical and logical) which are fixed and subject to no uncertainty. Any model with one or more random elements is known a stochastic simulation model.

Continuous and Discrete Simulation Models

Whether you use a discrete or a continuous simulation depends upon the objectives of the study. Discrete simulation consists of observations being gathered at points in time when changes occur in the system. Continuous simulation requires that observations are continuously gathered at all times during the simulation [10].

This research pertains to the manufacturing industry and particularly discrete event simulation above other simulation modelling, however due to the nature and purpose of the research, its relevance to simulation in other industries and areas should not be ignored.

2.4.2 Discrete Event Simulation Modelling

Discrete event simulation software packages will generally contain the same components which include the following [11] :

Entities

Entities cause changes in the state of the simulation in question. Each entity is defined by attributes which are needed to understand the function or purpose of that entity. Attributes could be process time or mean time before failure for tools on a manufacturing line for example. Attributes aren't necessarily unique to the entity and may be shared among other entities. Also anything non-physical in a system that causes changes are also entities, such as flow of information. These entities will also have attributes.

Activities and Events

Activities are the processes and logic present in a simulation. Events are the conditions that occur at some point in time which cause a change in the state of a system. Events are created when activities and entities interact with one another. The three major activities are (i) Delays, (ii) Queues and (iii) Logic.

Delays are when an entity is delayed for a definite amount of time. The time can be constant or randomly generated depending on the situation. Queues are when an entity is held up for an unspecified amount of time. Queues are generally a waiting line whereby an entity is waiting for a resource to be freed up. Finally Logic activities are those which allow the entity to effect the state of the system. For example should a batch enter a manufacturing line where one tool is down for maintenance this would increase the overall time taken to completion.

Resources

Resources are anything in the simulation that has a capacity. Examples can be workers, tools, machines etc. Entities use resources. However resources can also be unproductive. A batch may be at the end of its process waiting for collection to be moved to a warehouse. The batch (entity) must wait on the transportation (resource) so the simulation can continue.

Global Variables

A global variable is one that is available to the model at any stage. Anything of interest in the simulation can be tracked by these global variables. By altering the global variables an analyst can see the impact they have upon the simulation of the system.

Random Number Generator

A random number generator is a software routine which it creates a number between 0 and 1. Its relevance in simulation is for sampling random distributions. An example would be to produce the mean time before failure for a machine. Anything that requires a random value as an input is created by a random number generator.

The Calendar

The calendar is a list of events that are scheduled to occur in the future of the simulation. Any event that is due to happen will be present in the simulation calendar.

System State Variables and States

All systems have a common state variable i.e. the present time in the simulation. This present time is updated when each scheduled event that is completed. Systems can contain numerous state variables that when altered alter the state of the system. Examples of states of a system may be on or off, or more specifically for manufacturing “Manufacturing in progress”, “Down for scheduled maintenance” or “Down for unscheduled maintenance”.

Statistic Collectors

Statistic collectors are used to collect information on global variables, performance statistics of entities or certain states. The 3 main types of statistic collectors are (i) counters (ii) time-persistent and (iii) Tallies. Counters simply count. Time-persistent give the time weighted values of variables contained within the simulation e.g. the consumption of a resource. Tallies are observations that occur separate from the time between each observation.

2.4.3 When to use simulation

Simulation should be regarded as the next best thing to observing a real world system or process in operation. By using simulation one can monitor the performance of a real world system without having to alter that system or wait a large amount of time [10].

Carson [12] believes that simulation can be most useful in a number of situations that may present themselves in industry. Where there is no simple analytical model or easy calculations that can be implemented to analyze the situation, simulation should be employed. When the real world system cannot be properly defined and characterized simulation may be used. Similarly if the complexity of the real world system is too great it will be difficult or impossible to predict what effect any changes would have; Simulation can facilitate such a prediction. Maria [8] agrees that Simulation should be used when it is difficult or impossible to observe certain real world processes. She also sees its use when mathematical models can be created but analytical solutions are impossible or too complicated.

Carson [12] goes on to point out that simulation can be useful even where a real world system doesn't yet exist, as well as where major changes are being made or new and different demands are being placed on an existing system. Large investment may be put in these situations. If there is little or no knowledge/experience the investment could be at risk. Maria [8] also points out that simulation should be applied when the cost of observing the real world system would be far too expensive or if it is impossible. It should also be applied when it would be extremely expensive to validate the mathematical model describing the real system.

Carson [12] also sees simulation of use as a collaborative tool where a team or set of people can agree on a particular set of assumptions and test those assumptions on the model rather than the real world system. Real world systems can be large and sprawling encompassing many departments and personnel. Simulation can aid workers/managers/technicians/engineers when the involved people need to visualise the overall system and how their work and they're peers work affect it.

2.4.4 Simulation and Industry

Simulation can be applied to construction engineering, project management, logistics and distribution, transportation networks, health care, the military, as well as financial and business sectors. This is not an exhaustive list either. These examples show the immense scope there is for application of simulation. Whereas manufacturing companies were in the leading positions for simulation and its implementation historically, it is now commonly incorporated into all areas of industry. Song et al. [13] described a frame work for real-time simulation of heavy construction operations, when considering short term scheduling in their 2008 paper. With the ever enlarging city of Beijing, China, researchers [14] have turned to simulation to help understand the organization of public transport hubs and the direction that needs to be taken when gauging the current issues and problems. Another industry which has undertaken simulation as a tool for assessing their needs is Healthcare. De Angelis et al. [15] deal with what is considered a design and management problem in complex health care systems, using a combination of simulation and optimization. Miwa and Takakuwa [16] on the other hand use simulation to track the population, specifically customer flows in retail stores, so that a proposed procedure could be suggested to management for cashier scheduling, and in store merchandizing. Another example of simulation applied to what may be considered an abstract area by engineers, is Bernard and Lemieux's [17] 2008 paper concerning "Fast Simulation of Equity-Linked Life Assurance with a Surrender Option". Through a comparatively small amount of simulation they gain a quite accurate approximation of the surrender benefit should a holder give up their policy before maturity.

2.4.5 Simulation: Advantages and Disadvantages

Banks [3] discusses at length some of the advantages using simulation. Below are some of his points linked to the manufacturing industry. It is possible to test and change a system without ever having to make those changes in the real world system. Changing a real world system can be costly and time consuming. New policies and operating procedures can be surveyed once the simulation model has been validated, while doing the same on the real world system would cause interruption and incur cost. Simulation allows one to investigate a system by speeding up or slowing time. Large facilities can be diagnosed. Problems in such organisations may not be able to be pin pointed because of complexities. Specific constraints or problems such as bottlenecks can be identified through simulation analysis. A simulation model can help provide an understanding of a system and in turn be used as a tool to educate or train employees or relevant people to the system in question. Banks [3] highlights that "The typical cost of a simulation is substantially less than 1% of the total amount being expended for the implementation of a design or redesign". It makes sense that investment in simulation by companies can lead to a huge return.

Banks [3] lists some disadvantages to using simulation. He points out that model building takes time to learn. It's only over time that a modeller develops and their skills and experience can allow them to build successful models. Often a simulation study can be an intricate affair and its results could be difficult to interpret. If the correct resources aren't allocated to the study it can work out relatively expensive. Banks explains that coupled with simulation studies being complicated by nature, the simulation and analysis may not be up to the task. It is also possible for simulation to be applied when there is no need. Simulation should not be used in lieu of an analytical solution if such a solution is possible.

2.5 Documentation in the Simulation Industry

Documentation is an important element in software development and especially simulation and simulation studies. Documentation helps tie the simulation study together to create a quality package for a client or other involved parties in the study. It is not only the results that define the outcome of a simulation project; "Simulation project quality is assessed by a number of factors one of which is the quality of the documentation" [18].

Silbey defines documentation as "communication". To document a simulation is to describe how a simulation works, and then relay that information to a user [19]. Triebig and Klugl [20] explain that for simulation documentation, meta-level information should be included, unlike in documentation of software which mainly contains information on how the program is working. For example in simulation documentation it is necessary to assess the validity and objective of a simulation models development. As simulation documentation is used as a method of communication by a variety of people, modellers, programmers, operators and managers [19], and it is an interdisciplinary effort, often with many people involved, it should be understandable and useful to people from different areas with different backgrounds [20].

Oscarsson and Moris cite Lehman (1977) [21] and develop upon the three audiences that documentation should be aimed towards:

a) The Programmer or original simulation developer.
b) The new engineer/simulation developer who going to either study the model or use it.
c) Others who want to understand the main frame but not necessarily all details in the model.

This stresses the need that simulation model documentation has to be developed for all different audiences, as well as different 'levels' of documentation for each tier of abstraction.

The author of this work has used Gass [22] as a guideline and analogous with his reasons to document a computer based model, the reasons to document a simulation model are as follows:

- To enable simulation analysts, programmers, including personnel such as managers, engineers and technicians other than the originators use the simulation model.
- To assist the user in understanding what has been done and why.
- To record technical information that enables system and simulation model changes to implemented quickly and efficiently.
- To facilitate auditing and verification of the model as well as validation and evaluation of the model.
- To provide information to management so they may determine if and what requirements have been met.
- To improve "Organizational Memory" and in turn reduce the effects of personal turnover.
- To provide information about, assumptions made, simulation model maintenance, training required, changes made, experiments conducted, and simulation model results.
- To enable users and potential users determine whether the simulation model will meet their needs.

Where there are many tools to help create system models, as discussed in Section 2.7, there are a lack of tools available to help with documenting an overall simulation study. Simulation projects often can be split up using the “40-20-40” rule [23], whereby 40% of the project time is devoted to requirements gathering; 20% to model translation; 40% to experimentation (including validation and verification). Documentation can be costly but a lack of documentation of a simulation study may be more costly and inconvenient [21]. Good documentation of a simulation model is an essential prerequisite for quality assessment. It also serves as a vehicle for maintenance, reuse, or reproduction of the model and its results [20].

The process/system model is only one part contained within the whole simulation study [23]. Documentation is a pivotal part of a simulation as the final stages in any study are to document and present the simulation results [12][24][25]. Documentation plays a key role in recording the data and information required to initiate a simulation. Clema points out that poor or inadequate documentation is a source for error in simulation [26]. Verification and Validation in the simulation study is also aided greatly by good documentation [27], whether it is validation of the assumptions model or the verification/validation of the modelled created. Currently there is no standard practice for documentation in relation to simulation. McNally and Heavey [28] see the “lack of standards for documenting and building models” as a problem which leads to it being very difficult for anyone other than the model developer to maintain the simulation model.

Documenting the assumptions that are made about a system is an important step in “How to Build Valid and Credible Simulation Models”, as well as documenting any algorithms and collections of information/data [24]. Carson [12] suggests the creation of an assumptions document, along with a project plan as an early aim in the initial stages of the simulation project. He goes on to emphasize the information detailed in this shouldn’t be difficult to understand by the relevant parties as its purpose is to communicate the related information to the project team. Carson [12] continues, after stating that an assumptions document is essential, that a reviewed/signed off assumptions document is critical to a simulation study.

Similarly, Sadowski [29] believes that documentation plays an important role and should be used to record options that were unexplored or quickly discarded during the simulation study. The project team may use the documentation to know what the decisions were and why they were decided upon.

Good documentation should be found in simulation were models which are reused a number of times over a period of years. Law [30] highlights that it is imperative to implement good documentation if simulation models are to be reused. Since simulation can involve complex logic Elizandro and Taha [10] think the system in consideration itself must be well documented so to facilitate the design and development of the model, but they see an important part of the documentation process is to allow the use of comments in the model so that users can more easily see the logic of the modeller. In the same panel (Arthur, J. D., et al. [30]) previously mentioned, it is noted that a few “random” comments in simulation programs are not sufficient to be counted as proper documentation. A detailed document should exist describing the assumptions made about the model as well as the data contained within.

Documentation for a simulation study should be generated quickly as not to put a more time consuming burden on the project. As development and deployment of simulation can be hindered by many issues such as inefficient data collection, lengthy model documentation and poorly planned experimentation [31]. Accompanying documentation, for any area, of simulation should assist the project team rather than hinder them.

Carson points out that the writing and seeking approval of the specifications document early in a simulation project lay the ground work for verification and validation [32]. Documentation with respect to verification and validation is also discussed in Sargent’s 2008 paper [27]. The documentation required for both verification and validation can convince users of the accuracy and suitability of the model and furthermore its results. Sargent believes it is important to include a validation document in the overall model document and sees this as a part of a procedure which modellers should follow. Stahl [33] on the other hand deems “documenting the program” and integral part of the whole simulation process and through proper documentation of the program the modeller can convey specific information, which may otherwise be misunderstood by the project team. Good documentation can lead to easier verification and validation of the model.

Overall there is strong evidence that documentation is seen as a fundamental part of simulation by those involved in the industry. It is believed that documentation is a vital component of a simulation study in the recording of various stages of the study and as a means to communicate between involved parties.

2.6 How to create a successful simulation study

It is well known that simulation studies are an iterative process. A flowchart depicting this from [34] is shown in Figure 2-5. This section investigates the different steps as a precursor to examining the documentation requirements.

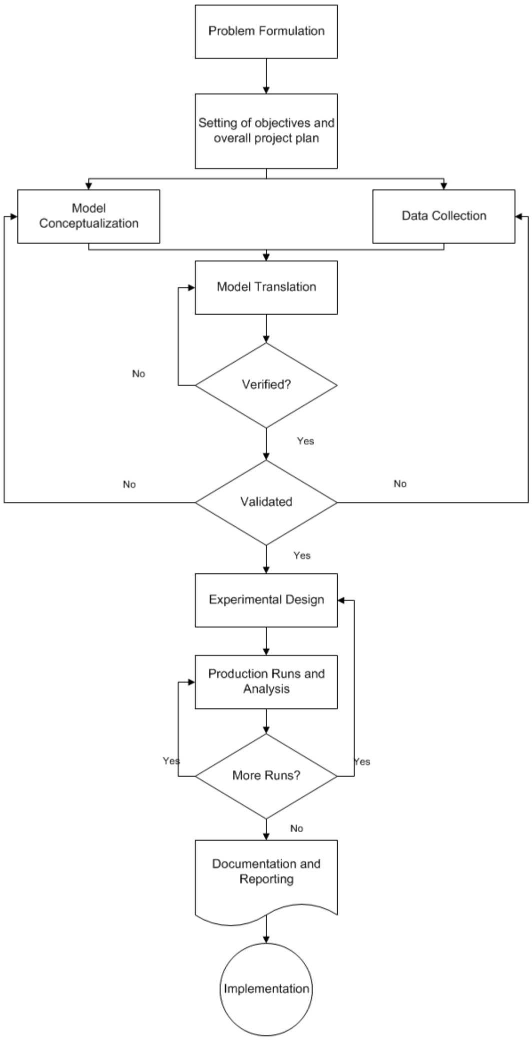


Figure 2-5 : Steps and the different elements in a Simulation Study [34]

2.6.1 Identify and Formulate the Problem

"Since the beginning days of simulation, the conventional wisdom has been that a successful simulation begins with a clear statement of the purpose of the model. This point is hammered home in nearly every introductory textbook on simulation methodology. One begins with a statement of the purpose and develops a model to meet that purpose. This statement of purpose includes the specific questions that need to be answered (e.g., predict daily production capacity) and the accuracy required (e.g., within 5%). The stated purpose then drives the level of detail (and the amount of work) that is put into the model". [35]

Each simulation begins with a problem. The problem that the decision maker needs resolved must be addressed at the very beginning of the study. If the problem statement has been supplied by a client, the project team must take extreme care that the problem is understood correctly. Vice versa, if the problem statement is prepared by the simulation analyst, it is paramount that the client understands and agrees with the conceptualization [3]. The project manager, along with the simulation analysts must gather with experts on the subject matter to lay out the entire project. The aims and goals of the study must be decided upon as well as what the specific questions that the study will need to answer. As well as discussing the scope of the model the project team must also decide on the system configuration that should be modelled. Another important topic that has to be clearly defined is the time and resources required, such as personnel, software, hardware and costing to complete the simulation. The requirements for the entire study should be decided upon at this stage. A project plan should be generated incorporating all the above, as a reference to refer to throughout the project cycle. The end user of the simulation also needs to be identified, e.g. management or line supervisor etc, and all problems should be formulated as best as possible at this stage [8].

2.6.2 Collect Information / Data from the Real World system

Accurate information of sufficient quality and quantity is paramount to a successful simulation study. A schedule of data requirements should be submitted to the client shortening the turnaround time of this phase of a simulation study. Often this can be a difficult task as data may be recorded in an order other than in which it was observed, it may lack accuracy or be grouped with other unneeded information [3]. Gathering data as well as the operating procedures with regard to the real world system must occur early in the life cycle of the project. Whether the data needed is static or dynamic should be clear from the first phase of the study, when formulating the problem (See Section 2.4.1). The project team should also collect the data needed to specify the specific model parameters and probability distributions at this particular stage.

2.6.3 Construct an Assumptions Model

The assumptions made in the simulation relating to the model must be spelled out, and collated along with the relevant algorithms and data summaries in a written assumptions document. Whether the model is deemed stochastic or deterministic

(Section 2.4.1) should be determined prior to the construction of an assumptions model.

Law [24] states that the level of model detail should depend on the following criteria:

1. Project Objectives.
2. Performance Measures of interest.
3. Data Availability.
4. Credibility Concerns.
5. Computer Constraints.
6. Opinions of Subject Matter Experts.
7. Constraints generated by time and money.

Banks [3] also recommends modelling begin simply and expand until a model of sufficient detail has been developed.

2.6.4 Validation of the Assumptions Model

The project team should thoroughly go through the assumptions document and address and errors or abnormalities, prior to building the model. Any omissions that are highlighted should be added to the Assumptions model [24].

2.6.5 Program the model

After the initial stages of the study the project team and the simulation modeller must decide upon whether the simulation model is discrete event or continuous, as discussed in section 2.4.1. It is up to the model programmer to build the simulation model based on the information from the preceding steps. The model should be verified at this stage. Verification should be a continuous process and documented systematically throughout the programming phase. Verification techniques can include varying input parameters over their acceptable range and checking output and substituting constants for random variables and checking the results and animations [8].

2.6.6 Is the Programmed Model Valid?

Comparing the model output to the real world system output will give results validation. The experts on the real world system should scrutinise the model results to check if they are consistent with how they'd perceive the system should function.

At this point it is important to increase confidence in the model. If there are any issues with the validity from management, system experts or the project team, they need to be addressed promptly and rectified [24][36].

2.6.7 Design, Conduct, and Analyze Experiments

After the construction and validation of the model the team must decide upon the configurations that need to be tested, run length, number of runs and the manner of initialization. Analysis of these experiments can determine if further experimentation is needed. The project team must decide upon the appropriate experimental design and establish experimental conditions for the simulation runs [24][36].

2.6.8 Interpret Output and Results; Present study.

When the simulation runs have been completed the final step to the study is to interpret the output and results and present the simulation study to the relevant audience. The assumptions document, a detailed description of the model and all results found throughout the study should be included by the project team in the final document. How the model was built, verified and validated should also be presented to the client. Whatever experiments are conducted should be detailed and noted by the team for the final document presentation [24][36].

2.7 Currently available Tools and Methods to aid Simulation and Modelling

Simulation modelling often becomes a heavy programming task, whereby the workings of a system can be lost to anybody not intimately involved with the programming task [23]. Ryan and Heavey continue outlining that simulation modelling is a poor communication tool in itself and doesn't support team-work well. Since the requirements gathering and conceptual model support is important during the beginning of a simulation study one could potentially use some of the numerous process modelling tools currently available as an aid. Kettinger et al. [37] compiled an abundant list of tools available. Ryan and Heavey [23] have assessed several of these tools and some of their findings and conclusions have been highlighted in the following section.

Petri nets

A petri net is a mathematical modelling language, based around simple objects, simple relations and simple rules. It is possible to represent a real world system through petri nets however such a representation is likely to be large and complex making it difficult as a communication tool or for a non-expert to understand completely.

Discrete event system specification (DEVS)

DEVS is a means of specifying a system. It is a way of formally representing a Discrete Event System. Representing a system using this mathematical method is difficult to understand without a greater knowledge of the entire DEVS formalism. As a result, using it as a communication tool between developers/experts and non-experts is a difficult task.

State Charts

State Charts are diagrams that show the flow of control or alternatively the different states a system passes through as a result of discrete events in a system. While State Charts excel in representing a dynamic system, they cannot represent resource interaction or the activities that cause the change of states in a system. State Charts lack the ability to fully communicate all the said interactions in a visual manner that can occur within a large complex discrete even system.

Activity Cycle Diagrams (ACD)

The method of the Activity Cycle Diagram is a way of representing the interactions between a system's entities. ACDs have the ability to model information, but struggle to capture complex logic and when representing larger complex systems the diagrams can become cumbersome.

Event Driven Process Chains (EPCs)

EPCs are sequences of functions and events within a discrete event system. The EPCs capture, represent and sequence the activities that occur during the progression of a process within a discrete event system. Event Driven Process Chains can accurately represent the previously mentioned activities but are unable to model different states or control in a discrete event system. As EPCs lack the capability of represent more aspects of a system they fail to function as a helpful aid for knowledge capture or as a collaborative tool especially in the initial requirement stages of simulation study.

IDEF 0

Integration DEFinition was developed by the US Air Force's ICAM (integrated Computer Aided Manufacturing) program. IDEF 0 was specifically developed so that the functional characteristics of a system could be modelled [38]. IDEF 0 describes activities in a hierarchical model whereby each process may be continually broken down until an appropriate level of detail is reached. IDEF 0 describes the activities as a combination of processes/activities (which may be broken down further) inputs controls and mechanisms. Where IDEF 0 is extremely good at what it does it is however unable to represent important pieces of a system, namely workflow and the flow of control, both of which are necessary during the initial stages of a simulation study.

Integrated Enterprise Modelling (IEM)

The IEM method uses an object-orientated modelling approach. The three generic objects that it is based around are; Product, Resource and Order. As IEM is restricted to only these objects it cannot represent detailed interactions that often occur in discrete event systems. This leads to it being of little use as a decent communication tool between modellers/developers and non expert personnel.

UML State Charts and Activity Diagrams

The Unified Modelling Language consists of a variety of diagrams, each with a different function. Often UML Statecharts (similar to the aforementioned state charts) and UML Activity Diagrams are used together to represent the execution of a process. UML Activity Diagrams consists of activities, decision points, synchronisation bars and transitions. Together these can be used to model or assist in the modelling of systems. Similarly to other methods UML Statecharts and Activity diagrams can't visually account for all the detailed interactions or use of resources that occur within a complex discrete event system.

SYSML

The Systems Modelling Language has been derived from UML as a general-purpose graphical modelling language and will be discussed in further detail in the following chapter.

2.8 Reusability

Reusability in relation to Simulation and Simulation Modelling is an important field where much research has been conducted. Robinson sums up why the area is interesting to modellers. "The idea of modellers saving time and money by reusing their own, or other people's, models and model components is appealing" [39]. If a modeller can save time and money, the same can be said for potential clients that the entire study is being conducted for. This section (2.7) also contains some references to research on composable models, that is models in the simulation industry that are created from pre-prepared sections arranged to suit the current needs of the simulation study. Reusability and composable models have been grouped together here to highlight some of the issues that this thesis intends to deal with by using Graphical Aids to assist in the simulation industry.

2.8.1 What is Reusability?

Reese [40] explains software reuse as the isolation, selection, maintenance and utilisation of existing software artefacts when one is developing new systems. For simulation modelling reuse there are several areas covered.

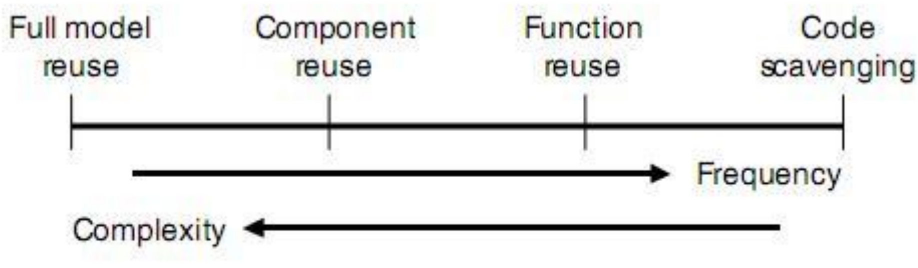


Figure 2-6: Reusability Spectrum [39] [41]

Figure 2-6 presents a spectrum for different types of software reuse gauging how frequently and how complex each area is. This spectrum shows that reuse is more frequent on the right hand side tending towards less frequent on the left. On the other hand complexity is greatest on the left hand side and least complex on the right. This figure is by no means linear but highlights that code scavenging is relatively easy in comparison to reusing an entire model which would be very complex [39][41].

Code scavenging is simply taking a small piece of code and implementing it into one's own software to obtain an equivalent purpose as the original. Functions that are reused have a defined functionality. An example of this would be a modeller using the already in place functions of a particular language. Reuse of a component becomes more difficult. Component reuse should offer all the benefits of the previous reuse options with added value. A component is a very specific item that should be used in a defined setting, hence the reason it is more complex to apply. Pidd [41] refers to Full Model Reuse as “the holy grail in some parts of the simulation world”. Reusing a full model would be extremely difficult and complex unless the simulation studies were exceptionally similar. Even then the knowledge and training require to use somebody else’s model could prove far too great an inconvenience, justifying starting the new simulation study from scratch.

Extending the areas that Pidd [41] discussed the following are areas of simulation modelling that can be considered for reuse, arranged in order of ascending complexity.

1. The modeller reusing small pieces of code.
2. Reusing knowledge from modelling in a previous study.
3. Design reuse of a particular component.
4. Specific components being reused.
5. Design reuse of the entire model.
6. Reuse of the entire simulation model.

Reuse will often need some degree of adaptation prior to the developer adding it to their own model. For the model or component to meet the new requirements the developer may have to add or remove events from the original simulation. Similarly they may have to change how the event is handled to fit into the new simulation model or change how the events are generated overall [42]. Adaptation is an important step in reusability with regards to the simulation industry.

2.8.2 Approaching Reusability

Pidd [41] insists that a properly developed strategy is extremely important to gain any benefits no matter the degree of reusability one is trying to apply to simulation. Whether one is reusing a small portion or an entire component from a previous model a strategy will help bridge the transition. Pidd goes on to point out that a strategy must include features to support abstraction, selection, specialisation and integration. High level concise descriptions of reusable elements are necessary so a developer can gain an insight into their exact function and behaviour. The selection feature should include a method of searching or alternatively a directory of model elements. By this the developer could locate, compare and select reusable artefacts easier. If a reusable artefact is a specialised one, it will require a feature to support that specialisation. For a specialised artefact it is likely that it will need some sort of modification before being inserted into another model and the specialisation feature will help aid that. Finally, the feature of integration support should provide a means of connecting the reusable artefacts i.e. integrating them into another model. By presenting a framework for the integration to happen, integration reusable artefacts will be consistent.

If one were to develop a model or a component with ability for potential reuse, then an architecture must be developed with adherence to: (1) a set of rules, (2) an interface specification (3) documentation standards and (4) a precise definition of the services any implementation must provide. The major problem is to adhere to such architecture will accrue a larger cost for the original developer but it will be the reusing developer that will gain all the benefits [35] [39] [41].

2.8.3 Reusability - Validation and Credibility

Pidd [41] believes the question of validity “looms very large” when considering model reuse. Proper quality validation costs both time and money. Pidd believes that the question of whether a model is valid and in turn fit for its purpose is more important than model fidelity. The complete validation of a model is what developers should aim towards but to attain such a level of validation would be next to impossible. This is the reason that Pidd believe that models should only be used (and reused) for the purposes they were originally developed for. Where a developer is reusing a model it is critical that they apply a new process of validating it, especially if it is being used for a purpose that it wasn’t originally intended for. What may have been deemed valid in one situation or environment may not hold true in another. Even if the model or component is being used in a situation where the fundamental assumptions are the same it is important to have it deemed credible in this new situation. Again the cost of validation and making the model/component credible may deter developers from reuse.

2.8.4 Cost of Reuse

A basic model for the cost of reuse was developed by Pidd [41] and highlighted again in the 2004 collaborative paper [39].

$$Kn = (C + A(N - 1))/N$$

C = cost to develop the software for its first use

A = Cost to adapt for reuse each time it is reused

N = Number of times that the software is reused

Kn = average cost / use

This formula shows that it is within a developer's interest to build a model with the potential for reuse if the model will be reused many times. The more a model (or component) is reused the less cost that is incurred. Another factor towards potential cost saving could be the lower maintenance cost of a simulation model [43]. Kasputis and Ng continued noting that the modular nature of composable simulations and the well-defined interfaces and descriptors of the modules would have a certain level of documentation, leading to the model maintenance tasks being better understood. With the maintenance tasks better understood, the maintenance would be in turn a cheaper process.

Davis et al. [35] highlight a promising idea for the future of simulation whereby prebuilt models and components exist. A developer can plug together the required components to form a model require for a certain study. The models and components would exist as a library and the developer could use them directly. Each model would only have to be built once, have its operation verified and then made available to be applied to any number of different applications. Banks [44] also sees the development of such library based simulation tools as a means to bridge the gap between "wide acceptance" and "wide use". Several products exist that allow the building of such libraries through the use of object orientation or template building [44].

Carnahan, Reynold and Brogan [42] highlight some of the characteristics that are prevalent in simulation but are a rare occurrence in other software. They are:

• Dependence on simplifying assumptions;

• Importance of insight versus precision;

• Use of stochastic sampling;

• Event generation;

• Time management.

They believe these characteristics can be exploited to speed up the process of simulation adaptation which will lead to reduced cost for simulation reuse.

For the simulation industry to reach a point where reusability is easily achievable a standard will have to be met which is applicable to all areas of the industry. Good documentation is one part of a standardized practice that will help achieve this.

2.9 Problem Identification – Aiding the Simulation Industry

This section highlights the problems that can occur during a simulation study yet could be addressed and ideally solved with good documentation practice. There is neither a standard regarding how a simulation study should be conducted nor any specification stating how the documentation of a simulation should be delivered. This thesis aims to investigate how "Good Documentation Practice" can aid a simulation study and the people involved as well as assessing how helpful it is to specific areas within the study. Furthermore SysML (The System Modelling Language) will be used as a tool to document and promote "Good Documentation Practice" in simulation.

2.9.1 Problems Facing Simulation

Simulation models of manufacturing systems can be highly complex and time consuming to develop. Oscarsson and Moris [21] comment on this and point out that it is not enough to only develop an accurate simulation model, but the model must be understood and furthermore updated and used by others. No standardised documentation system exists that could lead to a standardised way of explaining how and why a model was created. There are many notations available throughout the modern engineering industry to aid such matters. Some examples include UML, IDEF0 and Jackson Structured Programming (JSP). UML is used in software development, while IDEF0 and JSP are used in business process and procedural programming respectively. Oscarsson and Moris point out that "simulation models rarely are documented by such recognised and accepted notations" [21]. This lack of a uniform or accepted standard way of documenting leads to poor or scarce documentation of the both the logic and the model [21].

In reality it is often the case that the simulation modeller will begin building the simulation model only to have to revisit it time and time again to alter or revamp it to suit the needs of the client for which it is being built [45]. A client could also ask for certain elements of a study to be changed and in turn it is necessary to change aspects of the simulation model as the study evolves. The problem regarding this area is, without proper documentation in place, the study can evolve unnecessarily [36], potentially lengthening the modelling stage in addition to the verification and validation stages of the simulation, in turn increasing cost for the client. Having proper documentation in place defining what the study is and what the goals and deliverables are can also save the simulation analyst(s) headaches in so far as their client may want the simulation model to represent everything, but that may not be possible depending on the time and resources allocated to the study. For this reason it is important for the project requirements to be clearly defined at the beginning of the study so that they may be easily referred to for its duration. If a study does evolves, a document containing the goals, deliverables, scope, project outline etc could be used to discuss the study evolution with the client and whether it is necessary and/or feasible to implement.

Without good accurate information and data the simulation output will lack credibility. Since this step is paramount to a successful study, good document practice will ensure no key data or information will be mislaid. The information gathered can come in a variety of forms, so having a good structure for information and data capturing in place will allow easy translation of this information into the model. Without good documentation practice important data/information could be lost in the reams of information collected at the beginning of a study [36].

Regarding the modeller who begins building the model without the proper preparation, they too will find themselves returning to alter their model. Having an appropriate assumptions model documented will give the simulation modeller an easier job in translating this into the actual simulation model. This together with good documentation of the collected information and data will aid them in creating the simulation model aligning the simulation model and study with the well known mantra and practice of *Right First Time.*

Davis [35] sees a major problem to be the translation of requirements for simulation into components applicable to the job. He continues to note that a lot of researchers find this an extremely difficult problem to deal with and for the most part is ignored, instead of placing heavier emphasis on it.

Correct validation of the assumptions model can avoid substantial reprogramming later in the simulation model creation [24] [36]. When this validation step is skipped or skimmed past the later steps in a simulation study suffer, hence why it is important to have good documentation practice in place at this stage. If the assumptions regarding the study are correct the project can continue unabated. Good documentation practice of a simulation study up to and including the assumptions model will lead to a quick an easy validation of the assumptions model, so that the modeller may move easily on to building the simulation model. However if any points are raised by the audience validating the assumptions model, it is these points that need to be highlighted and documented so that they will not be overlooked further along the simulation study cycle.

If it is only the simulation analyst/modeller that has intimate knowledge of the workings of the simulation model much of the information contained within the model itself could be lost to the client or anybody else examining the model. With good documentation applied to the actual building of the model stage, the information within could be used to explain, train people or even reuse the model without the simulation analyst/modeller being present [36].

Creating a model that has greater fidelity than is needed for the purpose can lead to undesirable problems such as increased cost. Other problems relating to this are difficulty in verification, more expensive to alter and such models can take longer to run because of the increased fidelity [35].

Verification and validation are extremely important steps in a simulation study. The verification process can take up a large amount of time and incur large costs. A modeller/development team should utilize good documentation throughout their project up until this point, so they can save themselves time with easy reference to previous information. However this is not always the case in simulation studies. During the verification process the modeller is guaranteeing the actions occurring in the simulation model are the same as those happening in the system in question. Documentation from the information and data gathering phase of the project will assist greatly when comparing the simulation model to the real life situation [36].

Model validation is not a cheap process. If quite high confidence is required the cost can increase substantially [27]. If a simulation study has been well documented from the beginning of the project validation become easier. Again similar to verification this is not always the case. Where model developers can return to original assumptions models and information obtained from initial meetings on the subject matter, they can confirm that the created simulation is indeed an acceptable representation of the real system.

It is reasonable that the user(s) of a simulation model might not be versed in how the simulation model actually works, but only educated in using the model. Good information capture and documentation throughout the creation of the model will again assist the user in deciding the validity of the model in question. Similarly if a third party is involved or a scoring model is being used well documented information can only reduce the time taken for a simulation model to be validated and in turn reduce the cost. When good documentation practice isn't used this leads to an increase in the time taken for verification and validation [36].

A further problem is if a model is deemed invalid. The reasons behind such a conclusion must be addressed promptly. Where a project leader, manager, developer etc can return to the information gathering and model programming processes easily and quickly, it give them a better chance to assess what the problem may be.

Proper planning with regard to how one will experiment with the simulation model(s) can save monumental amounts of time. Similar to verification and validation, good documentation can aid the "experimental planning" stage of the simulation study. Noting what the modeller wants to achieve and outlining specifically what one wants the completed model to achieve will keep the said modeller's aims from drifting and exclude avoidable experimentation which can easily happen during a simulation, adding to time and cost [36].

The final step in any simulation study is the presenting of the results of the study to whomever it concerns. More often than not a document is drawn up encompassing all aspects of the study, including areas already discussed in this paper such as Verification, Validation, what tests and experiments were conducted by the simulation model and importantly what were the results and outputs. Compiling such a report will be made all the more easier if each of the preceding stages has been carefully documented using good documentation practice. When presenting the results it is of immense importance to be able to show where and how the results were obtained. Since the model has been previously verified and information and data contained within validated, the results are accurate portrayals of the real system, based on the assumptions made at the beginning of the study [27]. A major problem occurs in simulation studies whereby the project team/simulation analyst only begin compiling the information needed for a final presentation document at the end of the study.

Davis et al [35] believe that a great problem beset upon many novice users is that they start building a simulation model with "no clear purpose in mind or no understanding of the questions they want to answer". The parts of the system that the modeller understands to a better degree will be stronger supported and modelled in more detail leading to other elements of the model to be a bit neglected.

2.10 SysML – The Systems Modelling Language – An Introduction

Section 2.10 introduces SysML and the specification that governs it. Each of the 9 separate diagrams that are part of SysML will also be introduced. To explain each examples will also be given. Furthermore the pros and cons of SysML are highlighted in this chapter to explain where this graphical language excels and where its downfalls maybe. Finally an introduction to Artisan Studio Uno is presented. This is the programs used to generate the SysML diagrams presented in Chapter 4.

2.10.1 What is SysML?

SysML is an extension of the Unified Modelling Language (UML). The name SysML is attained from Systems Modelling Language. In January 2001 with the decision of the International Council on Systems Engineering's (INCOSE) to adapt and customize UML for the purpose of systems engineering the SysML initiative was born. With a collaborative effort The International Council on Systems Engineering (INCOSE) and Object Management Group (OMG) jointly chartered the OMG Systems Engineering Domain Special Interested Group (SE DSIG) in July 2001. The SE DSIG went on to develop the requirements for SysML which were in turn issued by the OMG in March 2003 [46]. SysML has been designed as a general purpose graphical modelling language and supports analysis, specification, design, validation and verification of complex systems [4]. These systems can include a multitude of elements ranging from data, information, software, hardware, operations, human resources and environmental characteristics. SysML was developed to be an open source specification project and the specification is publicly available for download and distribution, with an open source licence existing for its use. It reuses certain elements of UML, builds on others while introducing its own new features. Figure 2-7 depicts the crossover between UML and SysML in the form of a venn diagram. SysML reuses a large quantity of UML.

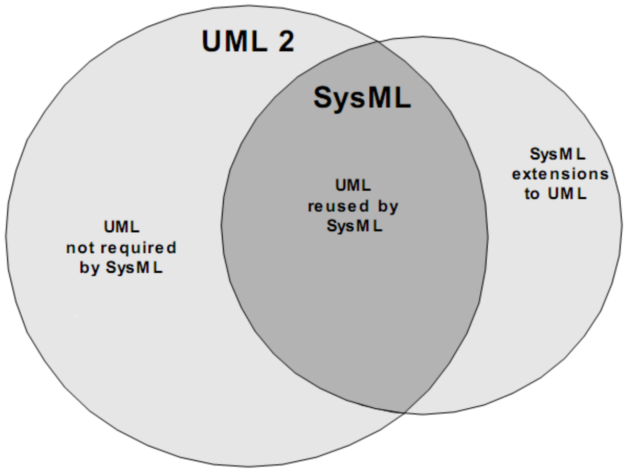


Figure 2-7: Crossover Relationship between UML and SysML[46]

The conventional approach to systems engineering is from the top down. As SysML was developed from UML it reinforces the view of designing / analysing systems at their highest conceptual level and working down into the finer details [4]. This “should make the traditional approach to systems engineering less prone to errors” and allow the implementation of systems engineering a more efficient process [47] SysML reuses a certain amount of UML2 represented in the venn diagram Figure 2-7.

2.10.2 SysML Pros and Cons

The Systems Modelling Language has wide ranging applications from the aerospace industry to medical device manufacturing and everywhere in between. Any system can be represented through this language thanks to the spread of diagrams available. As a modelling language SysML is both expressive and flexible allowing systems to be modelled by it [4]. The positives of having several diagrams is that while one specific diagram may not be applicable to the area of system that needs representation, another of the SysML can cover it. Mar [48] highlights four views, Functional View, Requirements View, Answer View and the Test View, that are essential for systems modelling. SysML covers these “views” and as this is the case the language is complete i.e. meets the minimum requirements for a systems modelling language [49].

SysML was designed purposely with the goal of representing real world systems. It builds upon the ground work laid out by the UML, its predecessor, but where UML was software orientated, SysML’s is orientated to represent real world systems, big or small [4].

Where there are other ways to represent real world systems SysML supports the analysis, design, and verification/validation [49]. The virtue of traceability is contained within the System Modelling Language which can aid the aforementioned processes that SysML supports.

The Systems Modelling language is an open source language. The SysML specification is readily available. In turn it may be implemented by anyone who wants to use is to represent systems in their industry [42].

SysML isn't the most intuitive modelling language when considering there are nine different diagrams. Where it may appear to the untrained eye that the diagrams are a collection of flow charts and boxed diagrams, they are more elaborate than this, with each separate diagram containing nuances and elements different to one another. SysML requires a certain amount of training to be proficient in it. It has steep learning curve [49]. This may be looked at on two different levels:

- the amount of training one needs to understand the diagrams;
- The amount of training needed to create the diagrams.

The reason for this is that a graphical language and graphics are easier to follow than the compiling of them. Furthermore different SysML diagrams are easier to understand than others. For example, the requirements diagram is inherently easier (requiring less training) to understand and construct than the SysML State Diagram.

2.10.3 Concept

The following are some of the more important ideas that have been translated from UML to the SysML specification.

Types

"Warehouse" is a type, specifically a type of building. A "type" is a means to categorize items into similar groups. Items in the same groupings will have shared characteristics, for example "warehouse" characteristics will include, size, floor space, number of loading/unloading bays, storage facilities. Individually each "warehouse" will have specific characteristics.

Generalization

Generalization allows new unique types to be created from existing types, whereby higher level types define common characteristics and the lower levels type inherit those same properties and characteristics. Figure 2-8 shows a hierarchical relationship between a "Building" two examples of its subtypes and the further subtypes of a "warehouse" including relevant characteristics.

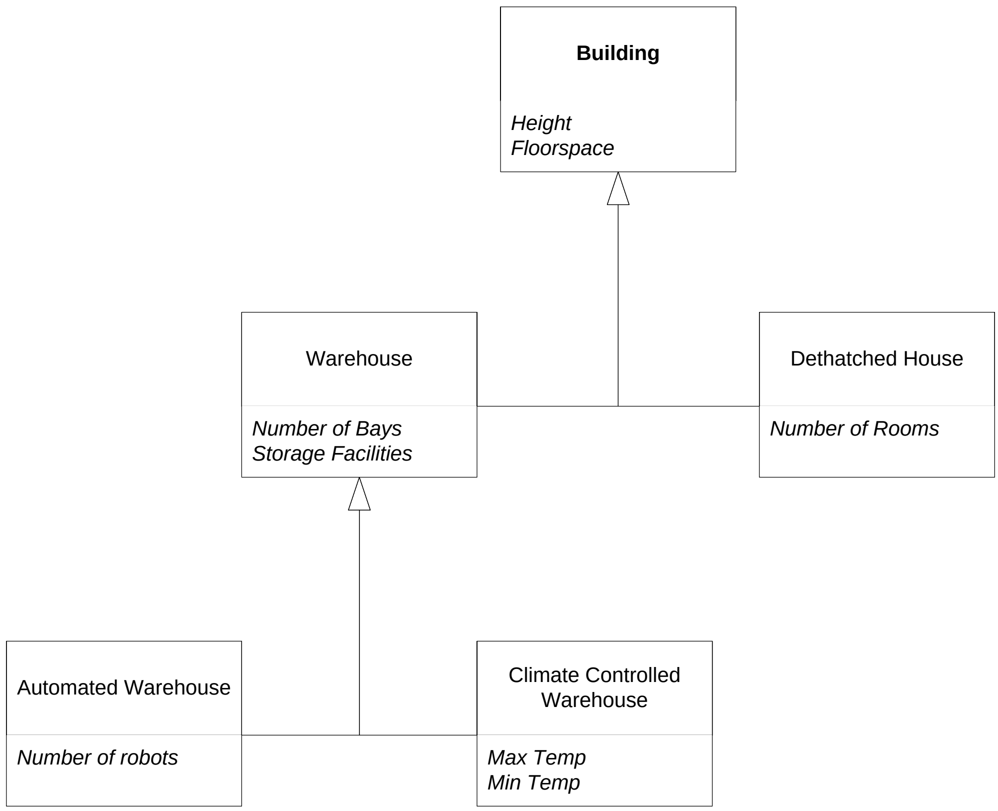


Figure 2-8 : Generalizations (Hierarchical Relationships)

Composition

A composition again shows a hierarchical relationship, however it depicts the parts contained within of the higher “type”. The relationships are defined by a solid black diamond and can label part names as well as types. This can be applied to either structural or behavioural models. See Figure 2-9 below.

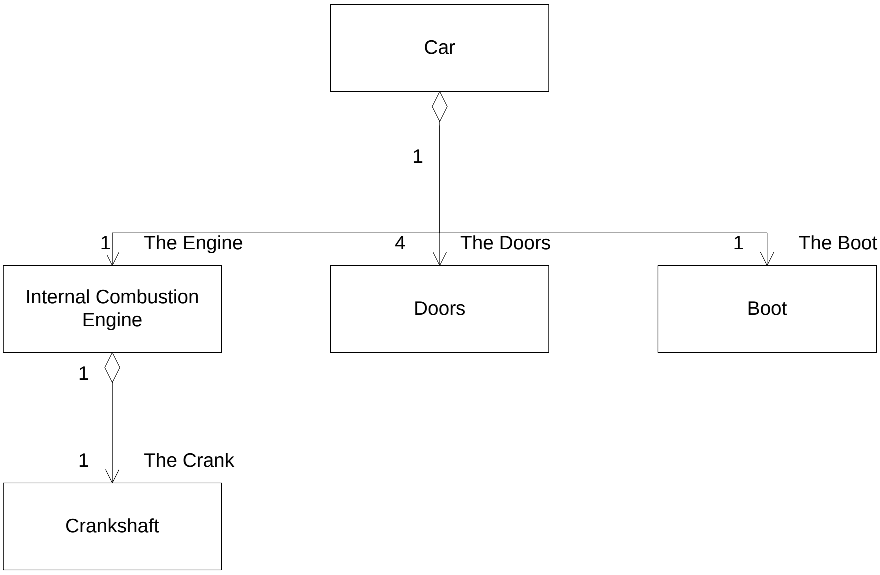


Figure 2-9 : Composition (Hierarchical Relationships)

Stereotype

A stereotype identifies specific properties to whatever model part to which it is applied. Its structure is in the form <<stereotypename>>. Stereotypes are user defined notation e.g. <<warehouse>> or <<system in fault>>.

2.11 SysML Diagrams

SysML represents complex systems through nine separate diagrams, each with its own specific function. The diagrams can be separated into behavioural and structural sub categories, whereby the behavioural diagrams represent what is happening in the system and the structural diagrams represent specifically what the system is. The SysML requirements diagram doesn't fall under either of these groupings and is considered separately. Following is an overview of each of the diagrams currently present in SysML. Table 2-3 groups the relevant diagrams to their respective categories where Figure 2-10 shows how each individual diagram feeds into the overall SysML diagram as well as any other relationships between diagrams.

Table 2-3: List of SysML diagrams

Behavioural		Structural		Other	
i	Use Case Diagram	v	Block Definition Diagram	ix	Requirements Diagram
ii	Activity Diagram	vi	Internal Block Diagram		
iii	State Machine Diagram	vii	Package Diagram		
iv	Sequence Diagram	viii	Parametric Diagram		

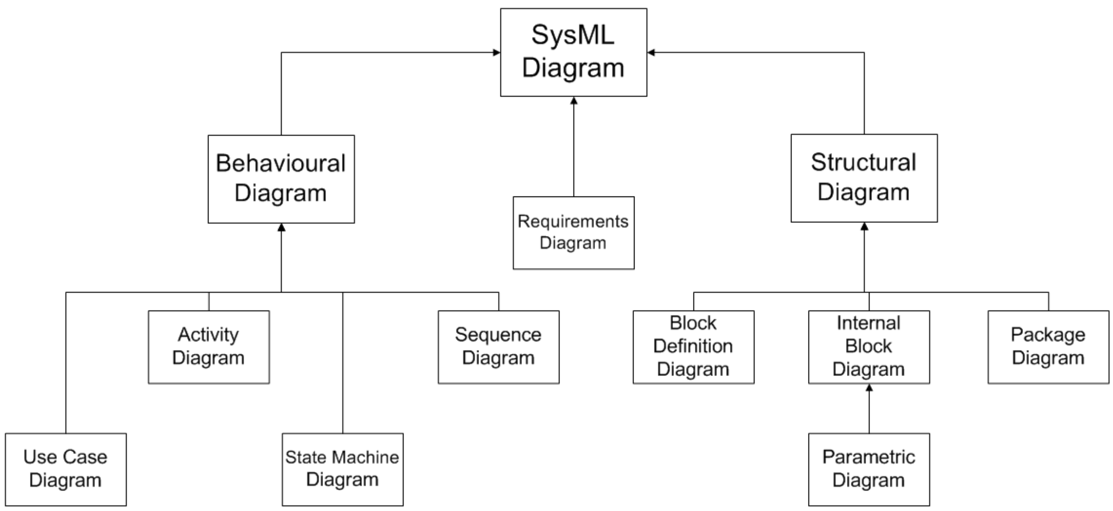


Figure 2-10 : SysML Diagram Taxonomy

Use Case Diagram

How a system completes its tasks through users' interaction and the interaction of outside entities (actors) can be described through Use Case Diagrams. Use case diagrams represent system information through both graphical and textual information. The use case diagram can be implemented in conjunction with the requirements diagram(s), capturing the requirements of the system, in so far as the system's uses. Use Cases outlines the services a system provides for the people (actors) involved [4].

Activity Diagram

In SysML, the activity diagrams represent the flow based behaviour in the system. Activities represent a controlled sequence of actions. An activity can be expressed as a protocol whereby an input enters and through an action (or number of actions) and output exits. In turn actions can be described as the way in which activities implement their purpose [4].

State Machine Diagram

A state machine is a potentially reusable definition of some state-dependent behaviour. In SysML state machine diagrams are applied when the system moves through a cycle whereby during each state the system acts differently. The state machine represents a system's reactions when internal and external events occur. Every state machine must contain at least one region. A region describes the state-related behaviour of the state machine. Each of these regions is described in terms of states and each region may on have one active state at any one time. For example a manufacturing line may have a start up phase, normal working condition phase, maintenance phase, unexpected downtime phase, shut down phase etc. During each phase a block representing part of the system may act differently. A state machine diagram is generally a child diagram of a parent block.

Sequence Diagram

Sequence diagrams represent the interaction of exchanges/messages between entities. When an owning block/entity implements its use the interaction can be seen as a sequence of events. In the sequence diagram, a block corresponds to a lifeline. The exchanges or messages that pass between lifelines can stand for exchanged signals or an applied operation in the system. Signals come in two forms; (i) where a signal travels from one life line to another the block executes its function (asynchronous) or (ii) where the first lifeline waits for a response signal (synchronous). Whereby an Internal block diagram illustrates a static picture of entity interaction, a sequence diagram describes it dynamically.

Block Definition Diagram

Blocks are the primary components when modelling a system through SysML. Blocks can represent a multitude of elements present in a system. Compartments can be added to a blocks representing the blocks different features. Blocks have three properties which are structural features. These are “Part”, “Reference” and “Value” properties. These properties stand for a role or use in the block. Blocks can also contain features such as “ports”, “operations” and “constraints” which are functions executed by the block. The block definition diagram defines the structural and behavioural attributes of the blocks as well as the interactions between the blocks. The block diagrams have the ability to model the interfaces, components and flow of a system, whether the system being modelled is a physical or logical one.

Internal Block Diagram

Internal block diagrams are quite similar to Block Definition Diagrams. Every frame of an internal block diagram represents a block. The internal workings of a block are depicted inside this frame, capturing the internal structure, including properties and the connections present between them. Internal block diagrams can also show flow between parts. Flow of data, environmental, physical, energy amongst others, can be represented.

Package Diagram

The purpose of the package diagram is to organise and structure the SysML model or a section of that model. Each package contains further packages of model elements. Correct organization leads to easily reusing model elements and finding one’s way through a SysML model. Package diagrams show the package hierarchy contained within.

Parametric Diagram

Each frame of a parametric diagram represents a block or constraint block. The parametric diagrams constrain the blocks by defining the constraining equation that the block(s) must adhere to. They are used to show how the constraint parameters and the value properties (belonging to parent block) interact. Parametric diagrams capture equations that govern the system being modelled.

Requirement Diagram

All different requirements can all be captured on the SysML requirements diagram, allowing a substitute for conventional text-based requirements and instructions. A requirement dictates a condition or ability that needs to be satisfied. The hierarchy of system requirements are represented on the diagram as well as different requirement interaction with other model elements. By grouping requirements and breaking them into their constituent parts through the SysML diagram, easy navigation and application to the model is achieved. The requirements diagram allows traceability of requirements to be easily observed between how they have been derived and the source of their derivation.

2.12 Creating SysML Diagrams for aiding a Simulation Study

Originally the SysML diagrams for aiding Simulation were created using the commercially available program Microsoft® Office Visio® 2007. Using a SysML template produced specifically for the Systems Modelling Language each relevant diagram was created individually. Each of the 9 SysML diagrams are represented within the template as seen in Figure 2-11.

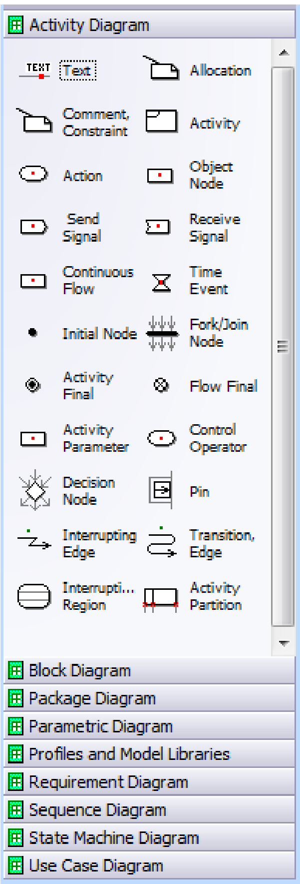


Figure 2-11: Visio – SysML Activity Diagram Template

The 9 different menus contain the relevant elements that can be placed into each diagram. The elements are dragged onto the diagram pane. Here they may be resized, rotated, moved and renamed among other options at the behest of the user. Figure 2-12 depicts several of the “Use Case” elements as an example.

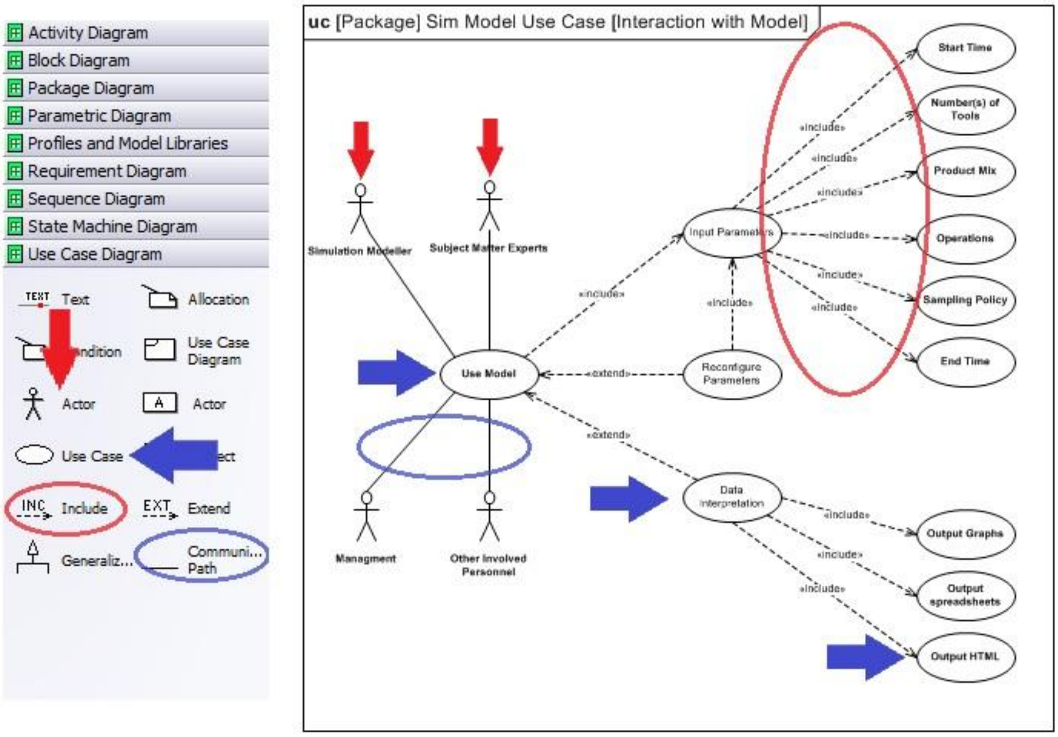


Figure 2-12: Visio - SysML Diagram Generation

Of particular interest are the connectors circled in blue and red respectively. These are used to connect diagram elements. It is up to the user how and where these connectors are placed. As is the case with other diagrams, when using Visio to generate SysML Diagrams it is possible to place a connector in the wrong situation. Similarly it is possible to place elements from other SysML diagrams into diagrams which would not adhere to the SysML specification. An example would be placing a “port” element on a “Use Case” element and using an “extend” connector to join it to an entire package diagram. Because all the content is user generated on the diagram there is scope for altering diagrams to suit the creating needs of that user. Without being held to the SysML specification strictly the diagrams could encounter validity problems of whether they are truly SysML diagrams or not.

The major benefit to using Visio is that the program is quite intuitive and doesn’t need any more than basic knowledge of using office suites, which the majority of computer users have been exposed to. The simplistic use of dragging items leads to easily created diagrams. However it is up to the user to have an adequate knowledge of SysML to create valid SysML diagrams when using a program that gives you the same freedoms as Visio.

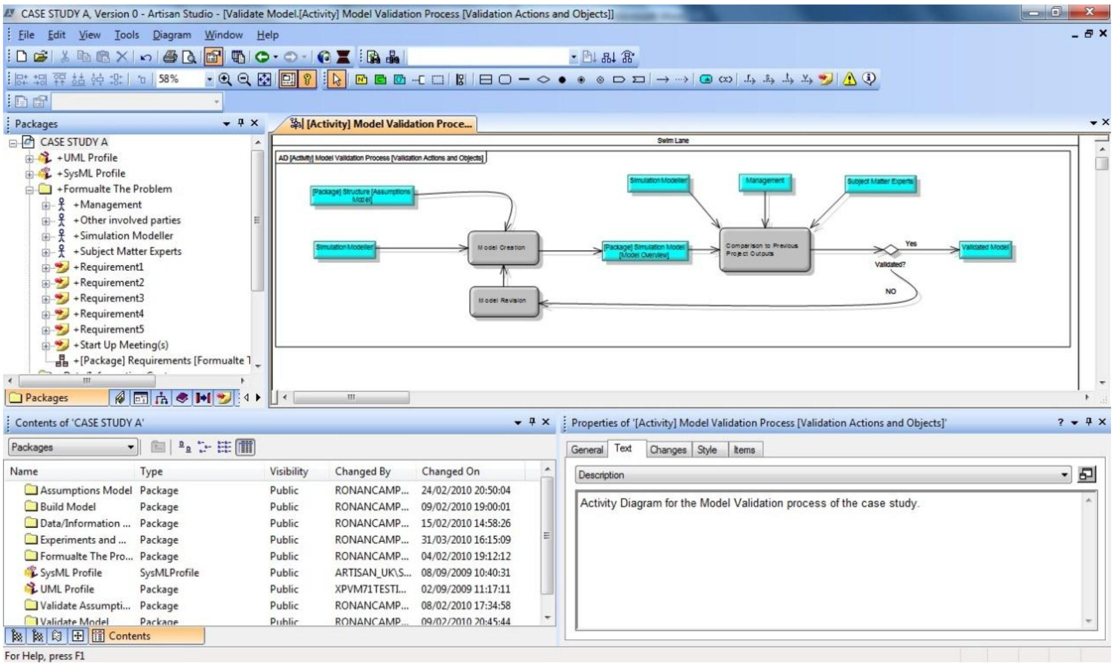


Figure 2-13: Artisan Uno Overview

Artisan Studio Uno™ is a free software and system modelling tool. The program has the ability to model using both the OMG SysML and UML standards. The program was created by Artisan® Software Tools as a standalone single user version of their standards based tool suite Artisan Studio®. Figure 2-13 shows a layout of the main studio Uno window. The window is split into four main areas each depicting different aspects of the created model, each which will be discussed in turn. Other information can be displayed but for this particular model the following explanations cover the important areas of the Case Study A SysML model which was created to show how SysML could be applied to a simulation study.

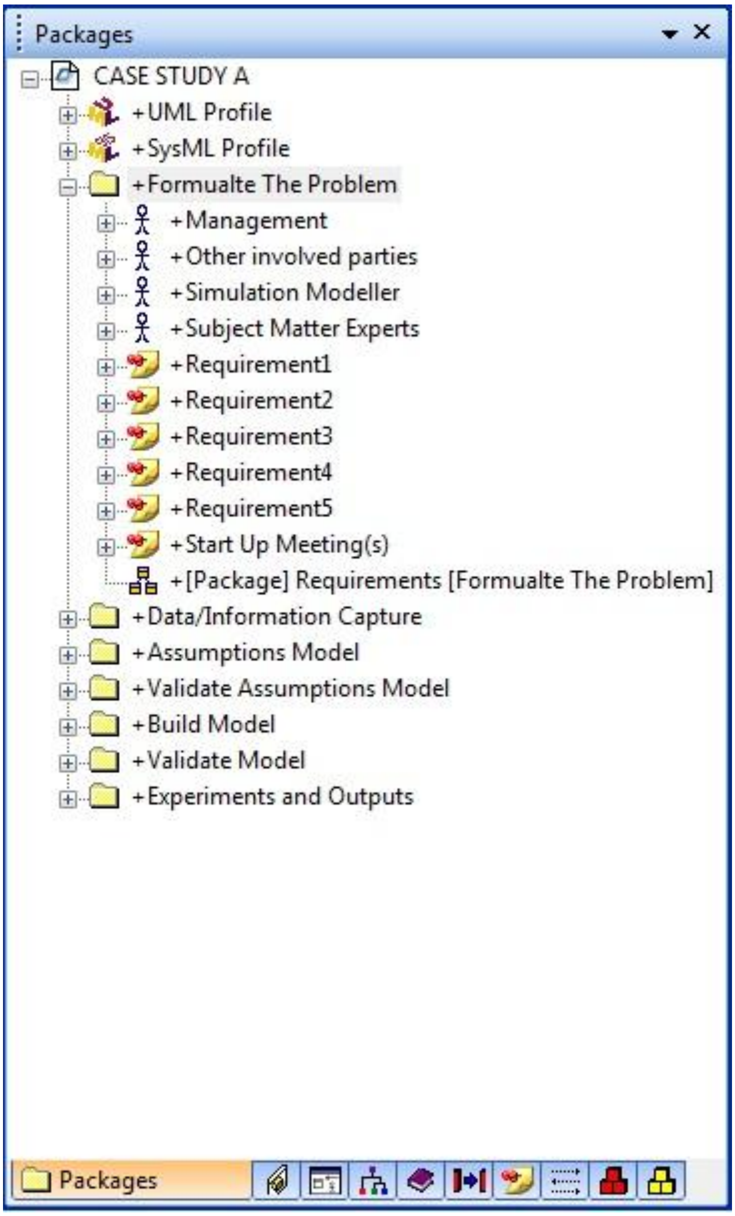


Figure 2-14: Artisan Studio – Explorer Pane

Studio Uno is set up with an explorer pane for navigating through the model. Figure 2-14 shows the hierarchical layout of the packages in the Case Study A model. The explorer pane allows the user to view the parts, diagrams, relationships, dictionary, ports and flows, requirements, allocations, blocks and constraint blocks using the same explorer, by selecting the tabs at the bottom of Figure 2-14. Where it is this work's intention of using SysML in a way that makes it easy to create diagrams to aid simulation, models created using the SysML specification can be quite large, in terms of number of diagrams and elements. The explorer pane allows the user to navigate the model easily. In comparison to Visio, one would have to keep track of the above themselves making it considerably more difficult to create large SysML (or UML) diagrams in that program.

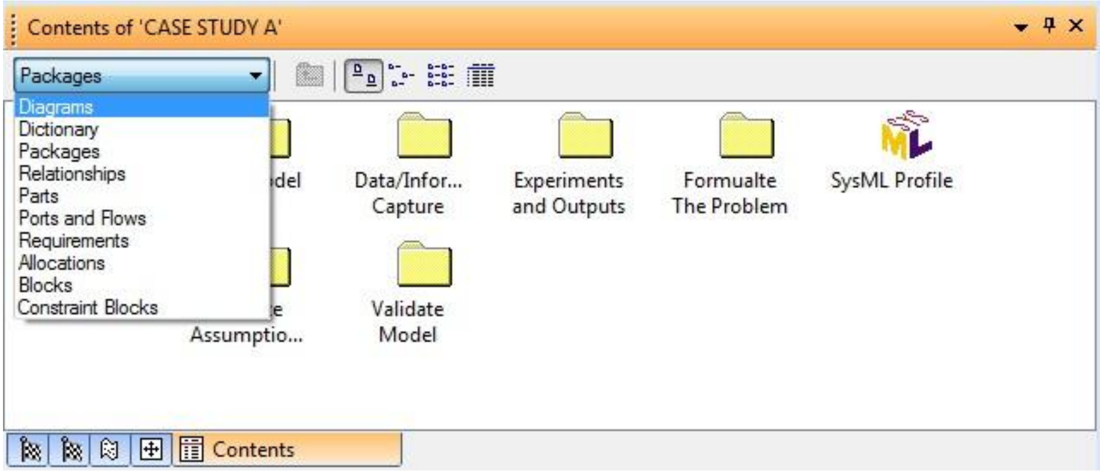


Figure 2-15: Artisan Studio Contents Window

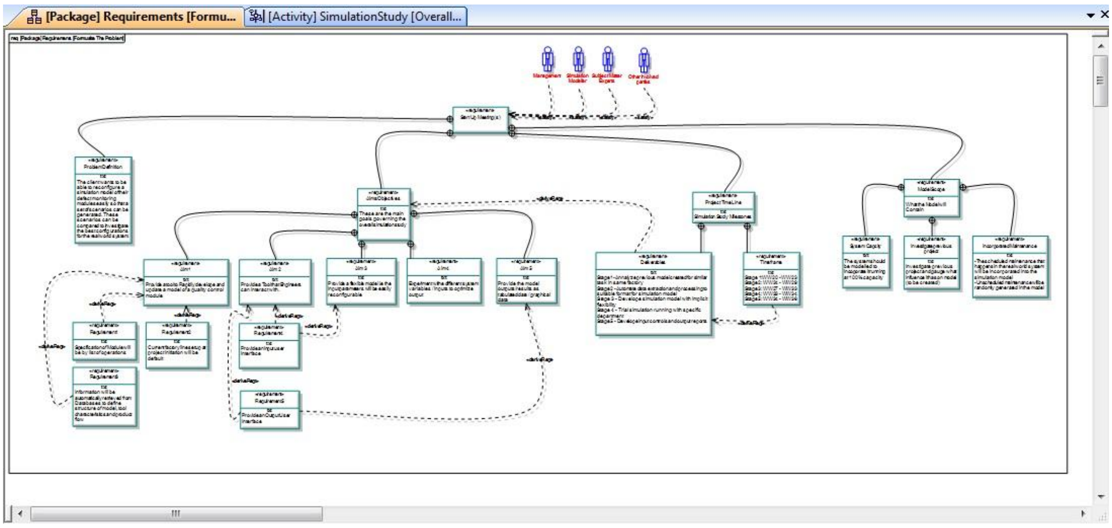


Figure 2-16: Artisan Studio - Diagram Viewer

The contents window allows the user to easily move between different sections of the model (Figure 2-15) but it is the Diagram pane is where the majority of the system modelling happens. It allows for multiple diagrams to be open simultaneously. When one places an element on the Diagram pane it automatically adds it to the explorer pane in the relevant area. This element is now available to be placed into other diagrams within the same model. In Figure 2-16 the actors are an example of this. They are visible at the top of the diagram and appear throughout the model in the activity and use case diagrams. These are the same elements being reused in both diagrams. All elements can have notes and text attached to them. By using the same elements on different diagrams any notes/text need only be attached once.

When two elements are joined with a connector on the diagram pane Studio Uno creates a relationship between the two and adds it to the model. One element may have numerous relationships between one or more elements throughout the diagram. Again the explorer pane is an easy way of viewing and keeping track of said relationships.

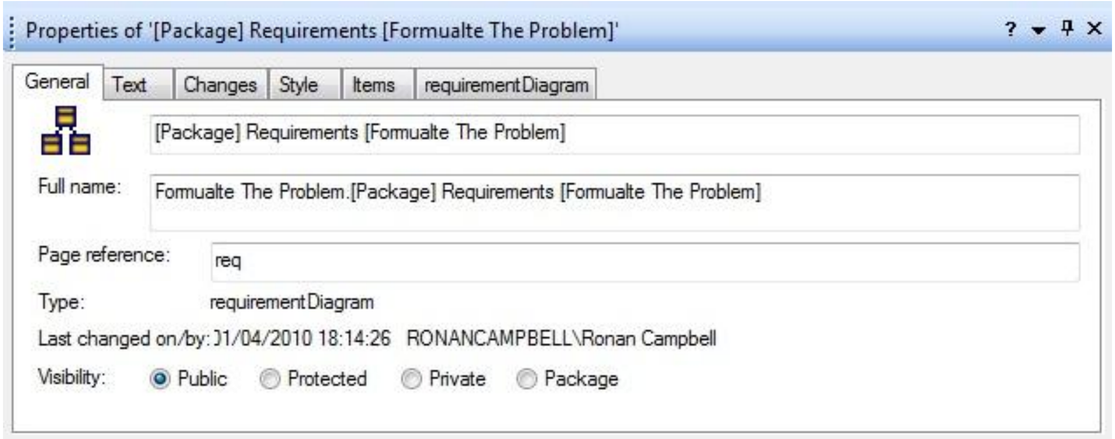


Figure 2-17: Artisan Studio - Properties Window

Every part of a SysML model created in Artisan Studio Uno has properties that can be viewed by means of the properties pane (Figure 2-17). Everything from the Name to whether it is public or protected is governed here. As mentioned the properties pane has scope for recording notes and text related to the model piece, that the user might need to refer to as can be seen in Figure 2-18.

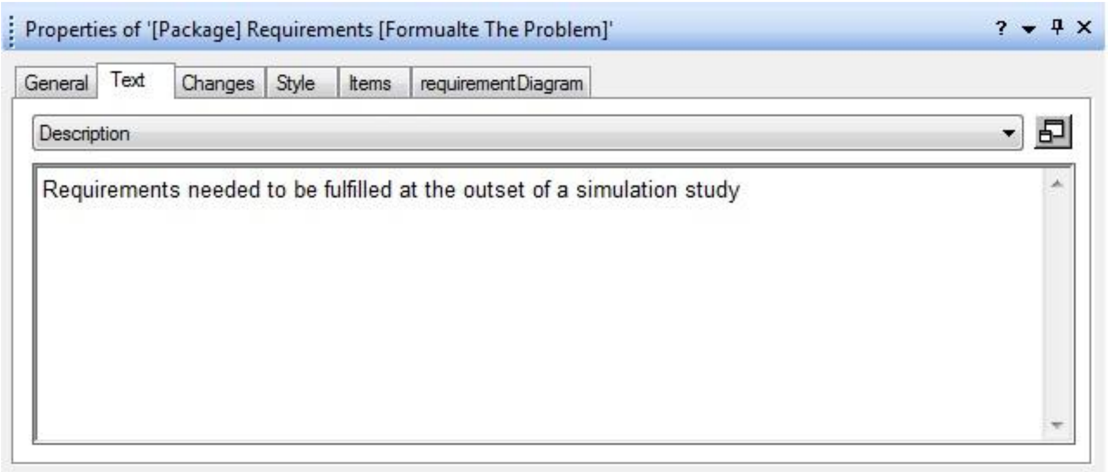


Figure 2-18: Artisan Studio - Properties Pane (Text)

Depending on the degree of detail needed the user can include as much or as little information as they want. If one wanted to record similar information in Visio the diagrams these could potentially become cluttered with text or alternatively would have to record the information somewhere outside the model itself.

Artisan Studio Uno is a purpose built tool for systems modelling. Whereby it is possible to create SysML diagrams in Visio, Uno does a noticeably better job. Even though this is the case Artisan Studio Uno is much more difficult to use for the uninitiated and requires a good amount of training to become proficient at it. The benefits of using it though are clear. It ties all the elements and diagrams together and steers the user to using the SysML specification as exactly as possible.

Artisan Studio Uno allows the user to include a great deal more detail in a SysML model in comparison to its equivalent in Visio. Having a more structured setting in Artisan Studio allows the user to focus on creating the model instead of concentrating on the SysML notation that needs to be implemented outside of a structured environment.

2.13 Conclusion

Simulation is a vast industry. An understanding of this industry is required if one hopes to have an impact on it. This was completed by gaining an understanding of it at its base, systems, models, constituent elements.

The literature review proceeded to investigate how documentation is used and how it affects the simulation industry. The importance of documentation in a simulation study was explored followed by the reason one should document a study.

To aid the simulation industry by individually aiding simulation studies one had to understand what was needed for creating a successful simulation study. The exploration of this subject in the literature review was important as it lays the foundations for the graphical aids created later on during this project.

Reusability was also studied, as good documentation has the potential to breathe better reusability into the simulation industry. Reusability is an area of simulation that there appears to be great interest in.

This thesis aims to aid simulation studies through graphical documentation. The problems involving the simulation industry and documentation were investigated as part of the literature review.

Several tools that have the potential to aid simulation were deliberated upon. SysML was chosen as it is a powerful tool that can be used in graphically representing all different aspects of systems. A further level of knowledge was researched for SysML (see section 2.10 to 2.12). It has 9 separate diagrams each with their own specific function. Although it takes a certain amount of learning/tuition SysML is flexible and does extremely well in supporting the analysis, design, and verification/validation of real world system models.

There are intricate programs available In relation to creating SysML diagrams. Artisan Studio UNO is a more intricate platform than the SysML template available for Microsoft Visio. Modelling in UNO necessitates more learning beforehand but the positives to using a structured program well out-way the negatives. Visio and other diagramming programs may be simpler and straighter forward to use but after investigation when it comes to creating valid credible SysML models specifically designed software is a much better choice.

3 Case Studies

3.1 Introduction

The following chapter presents three cases where simulation studies have taken place or are in the latter stages of development, and the documentation that the simulation analysts have applied. There is no specification on how simulation documentation should be recorded or presented so each case study was broken down into sections representing areas where good documentation could be applied. Each of the case studies was assessed under these sections and scored on a scale of 0 – 5 (Zero being nonexistent, 5 being excellent). This allowed an assessment of a person not intimately involved with any of the case studies take place, and allowed each study to be compared to highlight areas of common good documentation practice as well as areas which lack proper documentation.

3.2 Assessing the Case Study Documentation

To help illustrate and gauge the documentation Radar Diagrams will be used in the three subsequent assessments of the case studies. The documentation of each case study will be separated into different streams for consideration, similar to the areas that one must cover when conducting a simulation study (see section 2.6 for how to conduct a successful simulation study); Overall Methodology, Problem Formulation, Data / Information, Programming the Model, Validation, Experimentation are the areas investigated in each simulation study case study. Each stream will be scored so that each case study is measured on its uses and capacity for useful, relevant documentation.

- Methodology regards the overall methods applied to documentation by the team or simulation analyst throughout the project life cycle.

- Problem Formulation deals with the documentation relating to what is set out at the beginning of the simulation study, and what needs to be achieved by the project, as well as how it was implemented in the simulation study.

- Data / Information capture is the knowledge gained prior to the simulation model being built for the study and how well it has been documented. Furthermore the relevance and traceability of the data and information was gauged in each situation. The assumptions made about the real world system are also streamed under this section in the case study assessment.

- Programming the Model concerns what approach has been taken to the role of documentation during the construction phase of the simulation model. Verification is an important step during this phase and as a result has also been included.
- The validation documentation of both the assumptions model and the programmed model are assessed in this stream. As the two are intertwined the assessment covers both aspects of this important stage across the three case studies.
- Finally documentation that has been generated for the experimentation that the model has/will run is measured, including the documentation on how the outputs of the model and the results have been interpreted.

In each of the aforementioned streams of documentation the structure, clarity, ease of use, ease of understanding and ease of reuse by a third party has been measured. The headings below deal with those common attributes in each stream.

Structure:

The arrangement and organization of the findings in the documentation stream is considered here. Whether the structure is rigid or flexible, is adaptable, defines what information is documented or has the capacity to expand, have all been assessed. Whether the structure is consistent throughout the entire section is also scrutinized.

Clarity:

The clarity of the documentation deals with how readily available the documented material is and how clear it is to the person viewing it, e.g. a paragraph written in 5pt font may contain and convey all the necessary information but would be hard to read and the information may become lost in the text, hence poor clarity. Bullet points may make the information easy to read but if each point contains a one word description the contained information won't be deemed clear.

Ease of Use:

Ease of use is how easily the people involved used and applied their own documentation in their simulation studies. Ease of Use ties in with the both the clarity and structure, that has been previously gauged.

Ease of Understanding:

How comprehensible the documented stream is when being examined by an internal or external party. The ease of understanding assessment examines if the documented knowledge is readily understandable or whether specialist training or specialist skills are needed. It also examines whether the contained information of the stream is esoteric or not.

Ease of Reuse by Third Party:

Whether the documentation can be easily reused by a third party (not already knowledgeable of the project) has been considered. Since one aim of this thesis is to examine the possibility of using good documentation practice to aid further simulation studies in terms of model reusability. When considering reusability one has to consider reusing several pieces or partial pieces of the documentation.

The different areas of each case study will be gauged on the following scale as tabulate in Table 3-1.

Table 3-1: Documentation Evaluation

Rating	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Very Poor
0	Non Existent

To help the reader discern the above keywords from Table 3-1 have been italicized in the following sections to allow an easier parallel be drawn between the radar diagrams and written assessment of the case studies.

Each case study assessment is presented with a 3 x 2 matrix of radar diagrams. Each diagram has an accompanying section giving an insight into the documentation of that area of the respective case study.

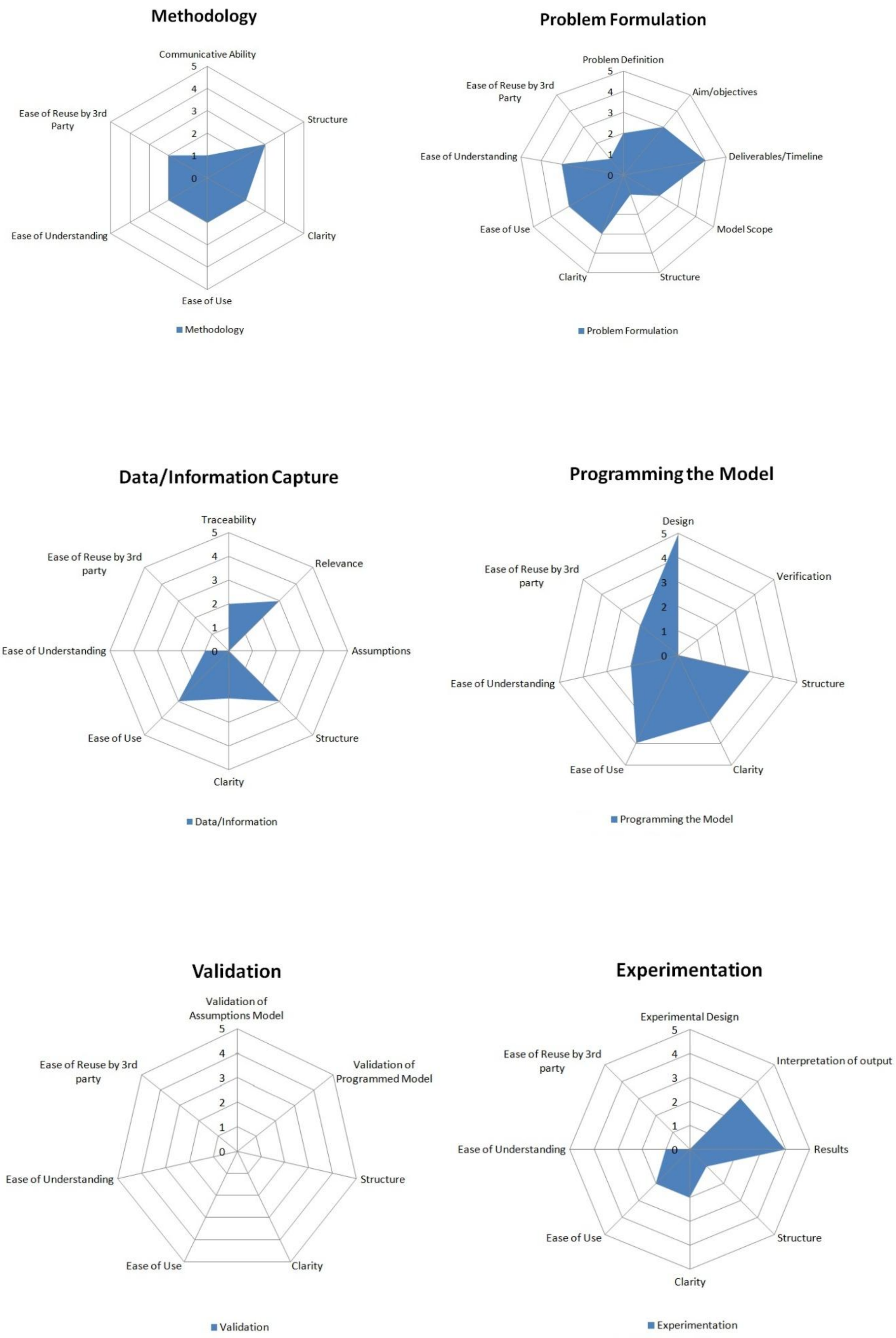


Figure 3-1: Case Study A Assessment

3.3 <u>Case Study A</u>

Case Study A is a simulation study created for a semi-conductor manufacturer. The simulation consists of a flexible/reconfigurable simulation model, simulating a monitoring module on a specific manufacturing line.

The majority of the documentation for the simulation in Case Study A was structured in the same way using a common office software suite. Most files were dated so as to easily reference each when using them. Much of overall documentation was recorded in point format with the intention of it being readily available to access and easy to view. The documentation was constructed to be easily communicated to an audience due to its format and structuring. The assessment of Case Study A can be found in Figure 3-1.

3.3.1 Methodology Assessment

The overall methodology was average in Case Study A. There was a certain amount of structure throughout the documentation; however the Case Study lacked a stringent structure. Most of the information was recorded by the simulation modeller in the form of presentations at regular meetings with the client. How clear and concise the actual information contained within suffered in turn from the lack of structure. The clarity of the said information was difficult to determine as a lot of the information was presented in bullet points and headings. The bullet points and headings were understood when written, i.e. as the modeller spoke and presented them to the client at some stage. However since the simulation study happened over a long period of time it required a certain amount of thought to recall exactly when each presentation covered, in some cases over a year after they were created. Hence the “ease of use” by the involved party was considered poor as the methodology applied to the documentation was impaired by its structure and clarity; how easy it was to use by those involved was difficult. Similarly the “ease of understanding” was hampered in the same way. One would have to be intimate with the simulation study, particularly the notes and meeting presentations to understand the methodology to an appropriate degree to reuse it in any capacity. The methodology applied in Case Study A led to any reuse of said methodology being quite complicated. Overall the way information was presented and communicated was quite poor. The communicative ability score suffered as much information wasn’t actually present in the presentations and documentation but remembered by the simulation modeller.

3.3.2 Problem Formulation Assessment

The problem definition wasn't spelled out particularly well in Case Study A. Everybody involved knew that the simulation study concerned the best arrangement of a quality monitoring station in a manufacturing line but this information wasn't specifically written down in any of the documentation. After several start up meetings between the clients and the simulation team it was decided what the aims and Objectives would be. This information was recorded better than the problem definition itself. The Deliverables / Timeline were spelled out quite well. The Aims and Objectives and Deliverables / Timelines were documented to a greater degree as the client was interested in how long the project would take, and when they would see results from this particular simulation study. Within Case Study A's documentation, the model scope wasn't defined clearly. Again all involved knew the scope of the model, e.g. whether the model would be deterministic or stochastic, or whether it would be a static or dynamic model. The problem that arises from not documenting the scope is that further into the study the client may ask for the model to be altered or the model may evolve unnecessarily during its creation. The same can be said for not documenting the other areas in the Problem Formulation stage. The structure of this stage was *very poor*. The simulation modeller didn't apply any specific configuration to it during the documentation. Although the structure was partially lacking in order, the contained information was for the most part clear, allowing the involved parties to use it to a *fair* degree as per the documentation evaluation (Table 3-1). The information didn't require specific training or knowledge so that it may be understood, for this reason the ease of understanding was *fair*. Even though the overall scoring for the problem formulation was decent how easy a third party would reuse the information was *very poor*. The documentation of this stage wasn't recorded with the plan of having a third party reuse it so it suffered.

3.3.3 Data / Information Capture Assessment

Case Study A required a large amount of information and data to develop into a successful simulation study. Through the radar diagram it is clear that much of the areas of documentation were subpar. It was difficult to see exactly where the information/data came from which led to the traceability score *poor*. The presentations in which the majority of the documentation was recorded contained relevant data/information, but the raw data needing input into the simulation model was difficult to interpret and needed the simulation analyst to sift through to find the relevant data. How readily available the information was to a viewer, both the presentations and spreadsheets of raw data, was lacking in clarity. For those involved in the simulation study the information was easily used, however one needed an in depth knowledge into the area and simulation study to understand the contained information and data.

The assumptions for the project were by not spelt out in the documentation, although the structure for the Data/Information scored *fair*. Even with a *fair* structure in place, the ease of reuse by third party was nonexistent due to difficult nature of understanding the documented simulation study's data/information as well as the aforementioned lack of clarity regarding this section.

3.3.4 Programming the Model Assessment

The documentation already in place for Case Study A contained no information regarding the model verification. Although this was missing, the documentation included by the modeller concerning the other areas of Programming the Model was stronger. This backs up the anecdotal evidence that simulation modellers concentrate mostly on the actual simulation model, than the overall simulation study. The structure and clarity were both fair within the documentation. It was easy for the modeller to document in a clear structured way that which they spent the most time on. The Ease of Use and the documentation relating to the design of the Model innately linked to one another. The simulation modeller's documentation was strong in these areas.

One would be required to have a good understanding of what was occurring at this stage in the simulation study and so the ease of understanding was *poor*. The ease of reuse by third party was also *poor* as without having an in depth knowledge of the project, or the model itself, the documentation of this section would be quite difficult to reuse.

3.3.5 Validation Assessment

The documentation contained no information regarding "Validation". After several conversations with an involved party in the Case Study A's simulation study, it was clear what the process of validation would contain, however this wasn't contained within the actual documentation. This situation will be discussed in section 5.2.5.

3.3.6 Experimentation Assessment

The interpretation of the simulation study's outputs was *fair* and the documentation covering the results of the study were well presented. The experimental design was nonexistent, which is due to the nature of the simulation study. The aim was to create a flexible simulation model that managers, engineers and technicians could experiment on themselves. As a result there were no specific experiments defined in this study. Examples of how the simulation modeller intended the model to be used could have been included within the documentation. That said the structure of this area of documentation wasn't well documented causing it to be very difficult to use and understand without detailed knowledge of how the model works. The clarity at this stage suffered as the simulation modeller hadn't concentrated on Experimentation documentation. Any reuse of Case Study A's Experimentation Documentation by a 3rd party would more hindrance than a help to their own study.

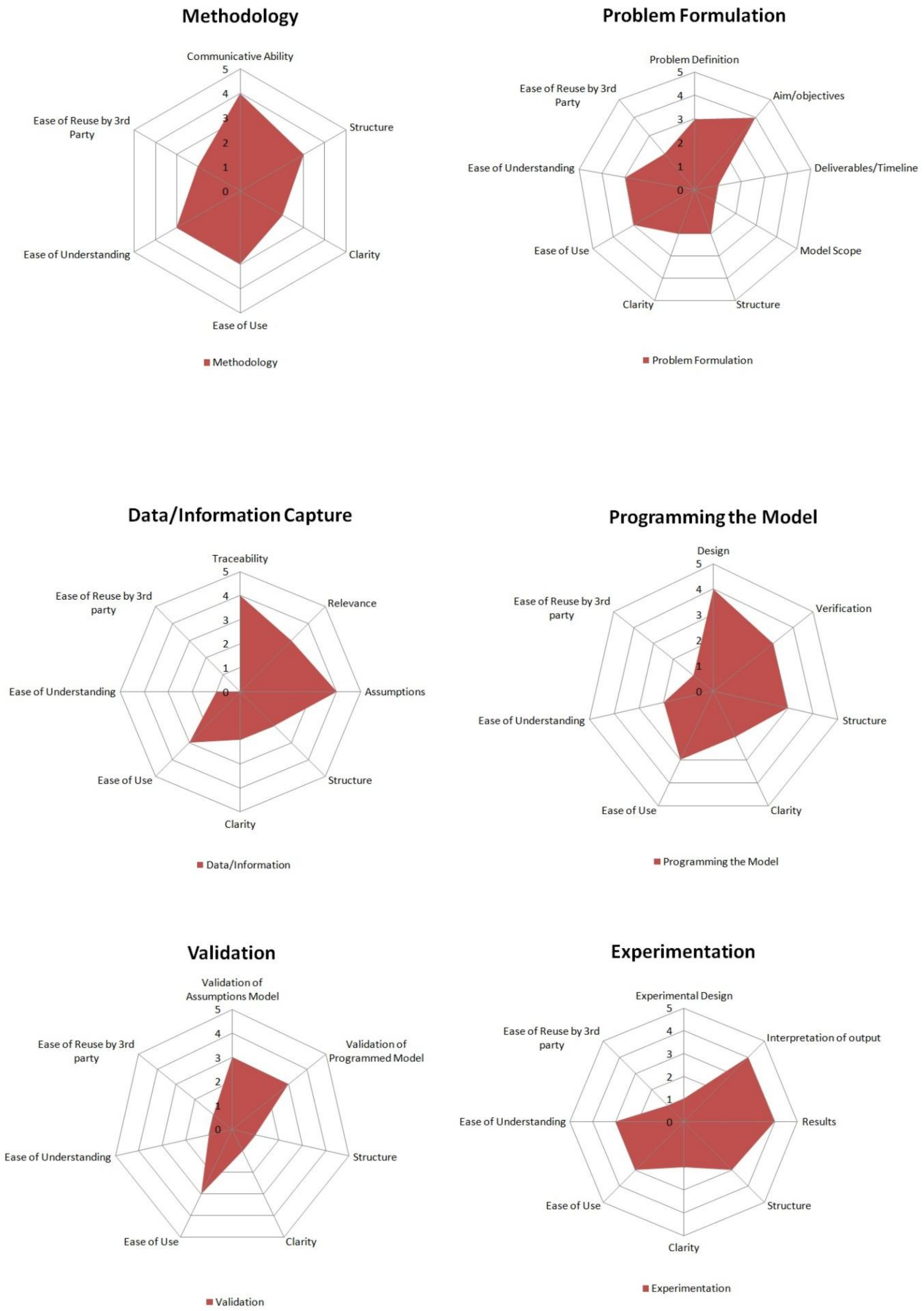


Figure 3-2: Case Study B Assessment

3.4 CASE STUDY B

The simulation study found in Case Study B has also been commissioned for a semi conductor manufacturer. The simulation as a whole is similar to Case Study A's however Case Study B's has concentrated on applying different operating curves to the simulation to glean the best results from the constructed model.

The bulk of Case Study B's documentation was recorded and presented in a report style document, which was not the final document that was to be presented to the involved client at the culmination of the simulation study. Had this been the final document to be presented to the client in question the documentation examined would more than likely be more complete. Figure 3-2 illustrates the assessment of Case Study B under Methodology, Problem formulation, Data/Information Capture, Programming the Model, Validation and Experimentation with each having an accompanying paragraph in this section.

3.4.1 Methodology Assessment

The simulation modeller applied a particular methodology to overall documentation for Case Study B. The structure was *good*, written in the style of a report. However the clarity didn't score as highly as the information was lost within the large paragraphs contain within the documentation. This affected the ease of use as it required the modeller or anybody else needed to sift through the document to obtain the specific information needed. No particular knowledge was needed to understand the methodology applied to Case Study B by the simulation modeller. However for a third party to be able to reuse this particular methodology would involve the simulation study being very similar to the one in Case Study B. Overall the methodology applied by the simulation modeller communicated the information quite well as seen that the communicative ability was scored *good*.

3.4.2 Problem Formulation Assessment

The problem that Case Study B intended to tackle was defined in the documentation to a *fair* degree. As the documentation of Case Study B was written as a report there was a large volume of text written on problem formulation and definition. Again how the aims and objectives were defined was *good* but was contained within a block of text. The simulation modeller covered both the deliverables/timeline and model scope, but each was *very poor(-ly)* defined within the documentation of Case Study B. To extract the information with regards to the entire "Problem Formulation" stage was difficult as much of the information was contained within the body of text, therefore the clarity was *poor*. The ease of use by the simulation modeller, of the contained information was *fair*. They were aware of all the contain information in the document and were able to access it easily to use. As this section had a consistent structure and no detailed training or knowledge was require the ease of understanding was also *fair*. Regarding the ease of reuse unless the simulation study and Case Study B were similar there would be no scope for reuse of this phase.

3.4.3 Data / Information Capture Assessment

The assumptions made about this simulation study were well defined and deemed *good* in the data / information assessment. The traceability of the data and information was also considered *good* in Case Study B's documentation. The simulation modeller spelt out clearly where the data/information came in their report style document. The relevance of the data and information was *fair*. The document contained a lot of information directly related to the simulation study but also much information that wasn't necessarily needed. The way in which the data and information was structured was *poor*. In the format of this documentation the data and information was spread throughout different sections with no specific headings or titles drawing attention towards specific data or information. The clarity of the data and information also suffered for this same reason, which also affected the problem formulation phase i.e. the information being lost in blocks of text. The simulation modeller found it *fair(-ly)* easy to use the data and information they had recorded and documented as it was their own work. The information was esoteric to some extent and would require a very good knowledge of the manufacturing process to understand it fully. All these points combined led to the assertion that a third party would not be able to reuse the Data/Information documentation of Case Study B's simulation study easily.

3.4.4 Programming the Model Assessment

The simulation modeller involved in Case Study B covered the design of the model in a detailed way. The model is generally considered the crux of the entire study and this is echoed by the good documentation practice applied by the modeller in Case Study B. The modeller also detailed how the model was verified to a *fair* extent. The information contained within the "Programming the Model" phase was also structured *fair(-ly)*. The clarity lacked slightly. Again the structure was consistent as in other sections of Case Study B but the clarity of the information on this section was *poor* as it was somewhat lost in all the text. The use of the information within the simulation by the modeller was fair but to understand the documentation fully one needed an excellent understanding of what the information pertained to, hence the reason for the *poor* score for ease of understanding, which also affected the reasoning behind the *very poor* score for ease of reuse by a 3rd party. An outside 3rd party not having the same knowledge or involvement of someone working closely with Case Study B's simulation study would find it difficult to re-apply the documentation to their own work.

3.4.5 Validation Assessment

The documentation regarding both the assumptions model and the programmed model were *fair* in Case Study B. There was little structure surrounding the section on validation and in turn how easily the information was presented to a reader/user was *very poor*. The information in the validation stage of Case Study B was to some extent easy to use, but an experienced understanding was required to reap the full information from this segment of the documentation. The simulation modeller had an in-depth knowledge of the validation process but for a third party to reuse this documentation would be very difficult as it lacked structure and clarity, and was difficult to decipher without a good working knowledge of the specific study.

3.4.6 Experimentation Assessment

The results were well documented in Case Study B. The simulation modeller highlighted results for presentation to the client. The interpretation of outputs from the simulation model were spelt out to a high enough degree that they too could be presented. The overall documentation for this section on experimentation lacked any defined structure or clarity. Even though the results and their interpretation were scored relatively highly the clarity suffered as once again the information was lost in paragraphs of text. For this information to be understood the involved personnel of Case Study B needed no explicitly specific training, other than that needed to be involved in the simulation study in the first place. Hence the reasoning the ease of understanding scored *fair* in the assessment. The ease of reuse of a third party using the documentation specific to this area of the Case Study was very poor due to the information being quite case specific to the actual simulation study.

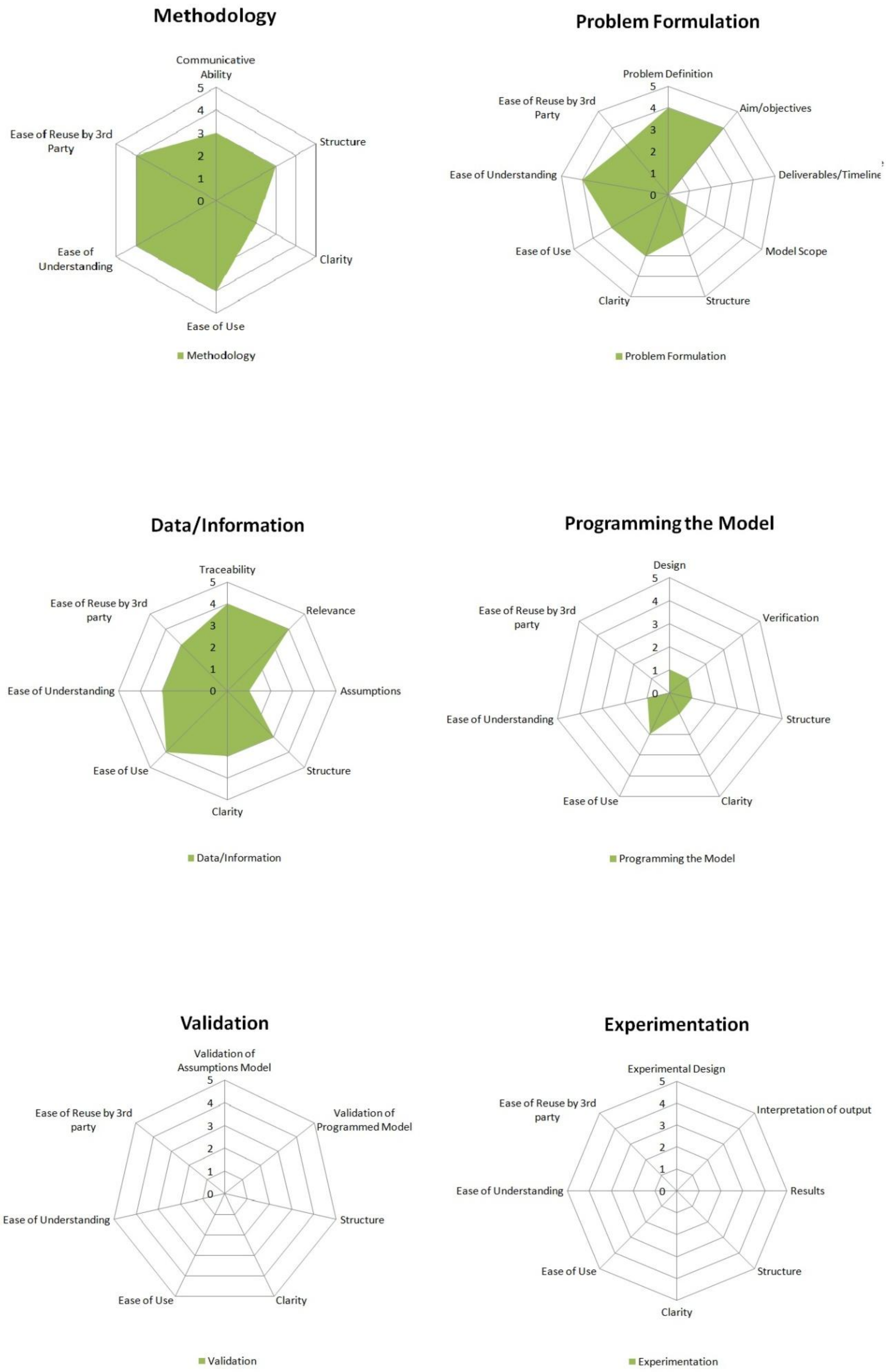


Figure 3-3: Case Study C Assessment

3.5 CASE STUDY C

Case study C is a simulation study yet to be commissioned by a medical device manufacturer, to identify what are the driving forces behind the cycle times on a manufacturing line and what maximum and minimum buffer sizes should be applied to the manufacturing cycles. The documentation supplied was in the form of a short report compiled using a word processor and associated diagrams. Figure 3-3 shows Case Study C's assessment with the relevant radar diagrams having an accompanying explanation following.

3.5.1 Methodology Assessment

The structure of the overall methodology in Case Study C's documentation was *fair*. The documentation was quite concise and to the point so it was easy to implement the same structure throughout. This is the same reason why it communicated the information *fairly* also. The clarity lacked somewhat. The information could have been labelled better to highlight it for the user, but no particularly complicated knowledge was required to understand it. Similarly its compactness would lead to a 3rd party being able to reuse this methodology in a further study. For the same reason the ease of use of the contained information was considered *good*. It was easy for the user to access the information from the short document.

3.5.2 Problem Formulation Assessment

The problem definition was outlined very well in this Case Study as this study was yet to be commissioned. The simulation team had already conducted several "start up" meetings with the client where they discussed the direction that the study would take. The aims and objectives for the simulation study were also outlined equally through these "start up" meetings. How each was documented lacked a definite structure even though both the clarity and ease of use were scored averagely. The structure may have suffered as the documentation wasn't created for any other purpose than to act as notes for the simulation team. Unlike Case study B with regards the clarity assessment and ease of use the information wasn't lost in reams of text. This made it easier to decipher for those involved. There was no documentation relating to the deliverables and timeline of the simulation study and the model scope wasn't spelled out particularly well in Case Study C. This may be because these details were yet to be nailed down between the client and simulation team. Overall no in-depth knowledge was particularly needed to understand the information contained within the "Problem Formulation" section and it seemed possible that a 3rd party could potentially reuse this documentation in another study. Even though the documentation was incomplete, there were clearly defined sections that could potentially be reused.

3.5.3 Data / Information Capture Assessment

For the most part the data and information documentation was good. The structure and clarity of this section was *fair*. The simulation team only included relevant information and no sifting of data was needed to glean that applicable information from the documentation. The data and information recorded was all quite relevant and where the data/information was obtained was easily traceable from the documentation. The simulation team saw the importance of having traceable information early in the study. Using the information/data captured in the documentation was easy for the simulation team thanks to the structure clarity, relevance and traceability. The documented material wasn't complicated. A more detailed knowledge of the entire proposed study and manufacturing facility would more than likely be required, but other than that the ease of understanding was considered *fair*. A third party could reuse this documentation with some ease. However the specific assumptions relating to the simulation study were by no means detailed enough and were scored *very poorly*.

3.5.4 Programming the Model Assessment

The documentation on the subject of "Programming the Model" was *very poor* as a whole. This is most likely due to the fact that the simulation study hadn't been commissioned at the time of writing and will be discussed further in section 5.2.4

3.5.5 Validation Assessment

It is pointed out section 3.5.4 that the documentation of programming the model suffered as the simulation study hadn't been commissioned and similarly there is no documentation on the validation. Documentation could and should still be created for validation prior to the model being built and will be discussed in section 5.2.5.

3.5.6 Experimentation Assessment

The experimentation stage contains no documentation to be assessed at all. Again it suffers the same as the Programming the Model and Validation Assessment in so far as the simulation model hasn't been fully completed and in turn no experiments have been put in place to gain results, outputs etc.

4 Applying SysML to the documentation of a Simulation Study and Assessment

4.1 Introduction: Application of SysML to the Documentation of a Simulation Study

This section deals with the application of the Systems Modelling Language to those areas of a simulation study that can benefit from having accompanying documentation. The following sections each relate to a different "step" as seen in Figure 4-1.

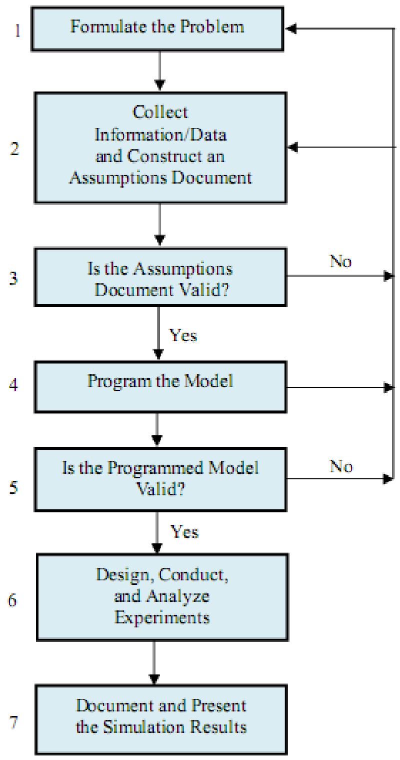


Figure 4-1: Law's Seven Steps for a successful simulation study [24]

Law [24] outlines seven steps to complete a successful simulation study, shown in Figure 4-1. Using Law [24] as a guideline, documentation in the form of graphical diagrams using the systems modelling language SysML will be applied to the simulation study Case Study A as seen in Section 3.3. Case Study A is a simulation study created for a semi-conductor manufacturer. The modeller created a flexible/reconfigurable simulation model where a module was simulated with the intention of investigating the quality of said module.

The approach was taken to create graphical aids to cover all the elements of the final document that would be presented to a client in the concluding stages of a simulation study. This chapter deals with a breakdown and explanation of the SysML diagrams that have been compiled and used to represent the different sections present in the above steps needed to create a successful simulation study. While the majority of the diagrams created are intuitive in themselves, an explanation of each will broaden the knowledge and application of each specific diagram and where it can be applied in other studies. Law's seven steps [24] have been broken down slightly further whereby the data / information capture section and the construct an assumptions model section is divided into two separate sections (i.e. having their own SysML diagram), showing that there is more scope for SysML to aid the different areas of documentation needed.

4.1.1 Formulating The Problem

This requirements diagram, Figure 4-2, represents what has to take place during the initial stages of the simulation study. The diagram itself is hierarchical showing the higher level requirements followed by the lower level and further derived requirements.

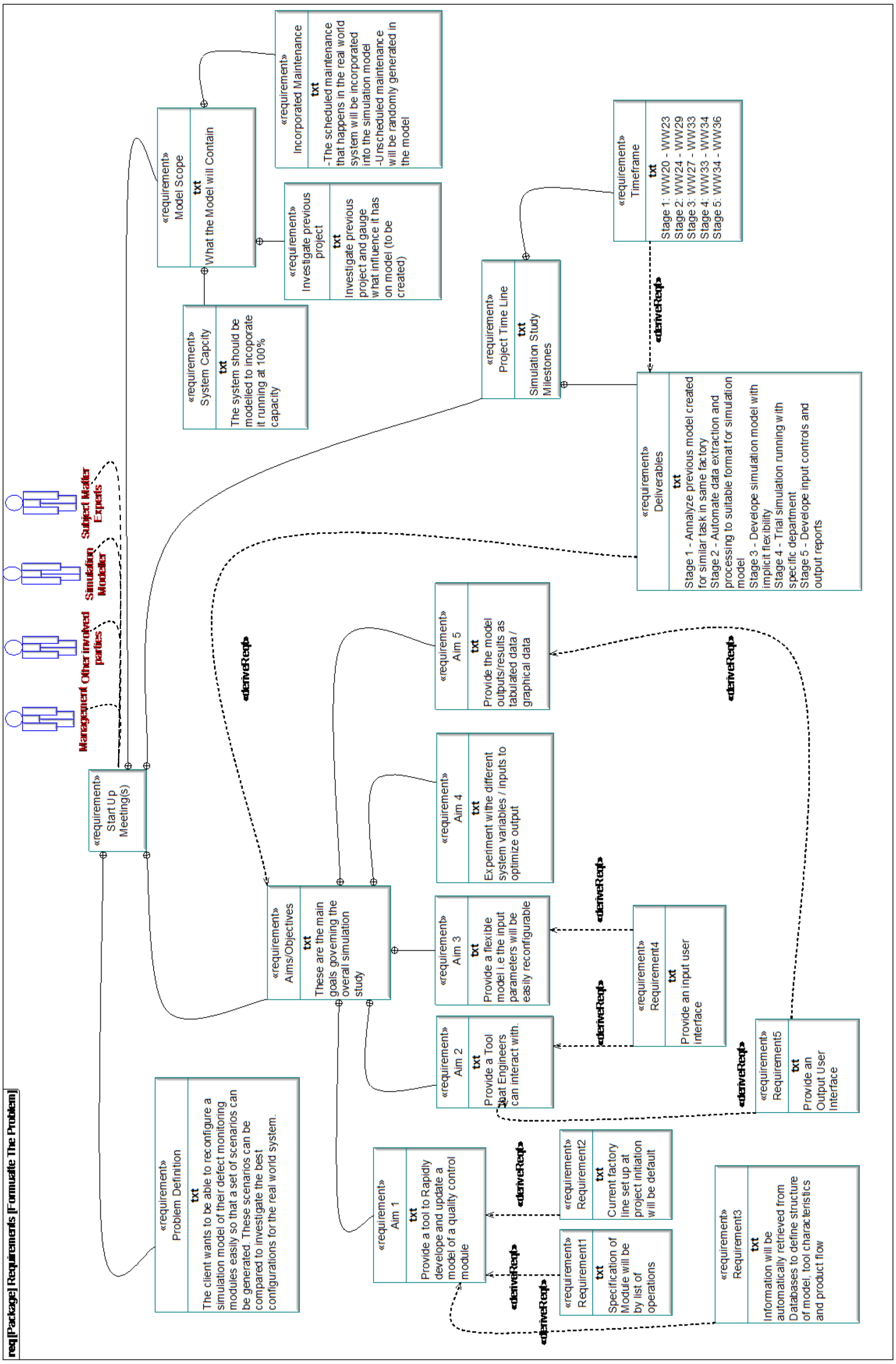


Figure 4-2: Requirements Diagram - Formulating the Problem

The top level requirement in formulating the problem is the meeting that all involved parties need to attend. From that the next level requirements are those of “Problem Definition”, “Aims and Objectives”, “Project Time Line” and “Model Scope”. Through the text description, on all requirements, the involved parties can see what exactly is required for each level as an explanation on the each label. Using Aim 1 (ID: A1) as an example, it is seen that it is a contained requirement of Aims/Objectives (ID: S1.2). The involved people in this simulation study decided upon “providing a tool to rapidly develop and update a model of a quality control module”. From the requirement Aim 1, three further requirements were derived as a result of said Aim 1, namely Sub-Aim 1, 2, and 3. Similarly it can be seen how Sub-Aim 4 (ID: A2.1) was derived as a result of Aim 2 and Aim 3, as well as Sub-Aim 5 (ID: A2.2) being a derivation of Aim 2 and 5.

The requirement labelled Deliverables (ID: D1) is a contained requirement of “Project Time Line”. However it is also derived from the “Aims/Objectives”. This can be explained as the deliverables exist as a requirement because of the parent requirement i.e. “Project Time Line”, but what the deliverables themselves are, is when the aims/objectives are due, hence the derived connection between the two (see connection between ID: S1.2 and ID: D1).

Actors represent the role of personnel, an organisation or an external participant in a system. All four actors in Figure 4-2 represent the personnel involved in the simulation study. They are joined to the highest level requirement through a satisfy connection. The connection says for the “Start Up Meeting” requirement to be fulfilled these four people must be involved. Subsequently any lower level requirements will also be governed by this satisfy connection meaning any child requirements of “Start Up Meeting” (ID: S1) must also be satisfied by all four actors.

4.1.2 Data and Information Capture

The information and data input into the simulation model is recorded in the package diagram Figure 4-3.

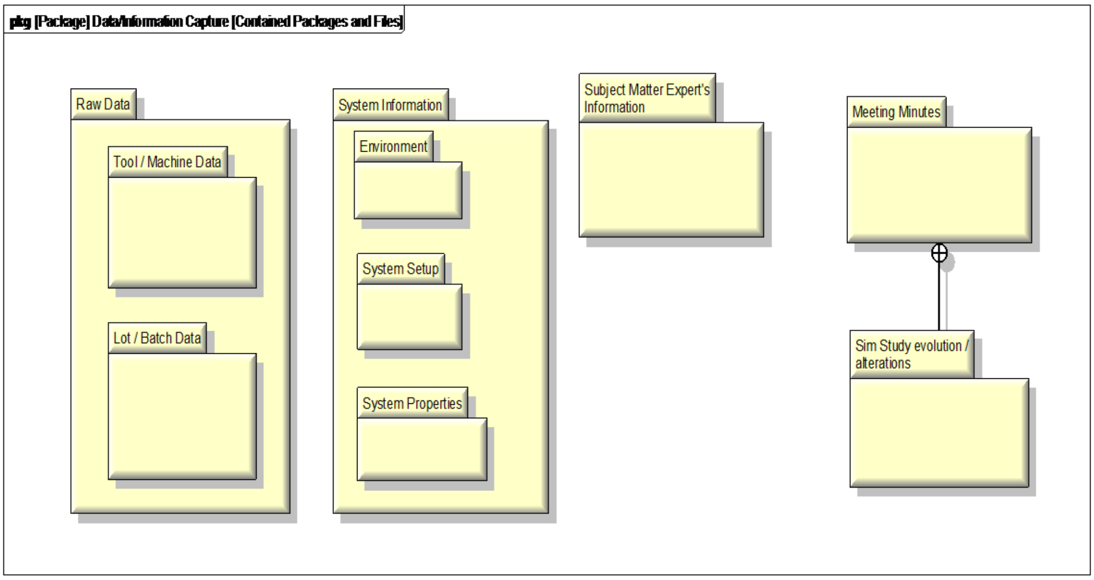


Figure 4-3: Package Diagram - Data/ Information capture

The package diagram is quite straightforward as it depicts the packages representing what information is necessary and within each package is the child packages for separating said data/information. The packages contain an assortment of information and the reasoning behind approaching the data/information capture using this specific SysML diagram will be discussed in section 5.3.

4.1.3 Generation of the Assumptions Model

In generating the assumptions model three SysML diagrams have been created. These document certain areas relevant to the assumptions for the overall simulation study. They are an activity diagram showing the flow of the simulation model, a block diagram representing the different sections of the simulation model and a requirements diagram showing the assumptions made about the diagram. These SysML diagrams represent the step that occurs prior to being able to build the actual simulation model itself.

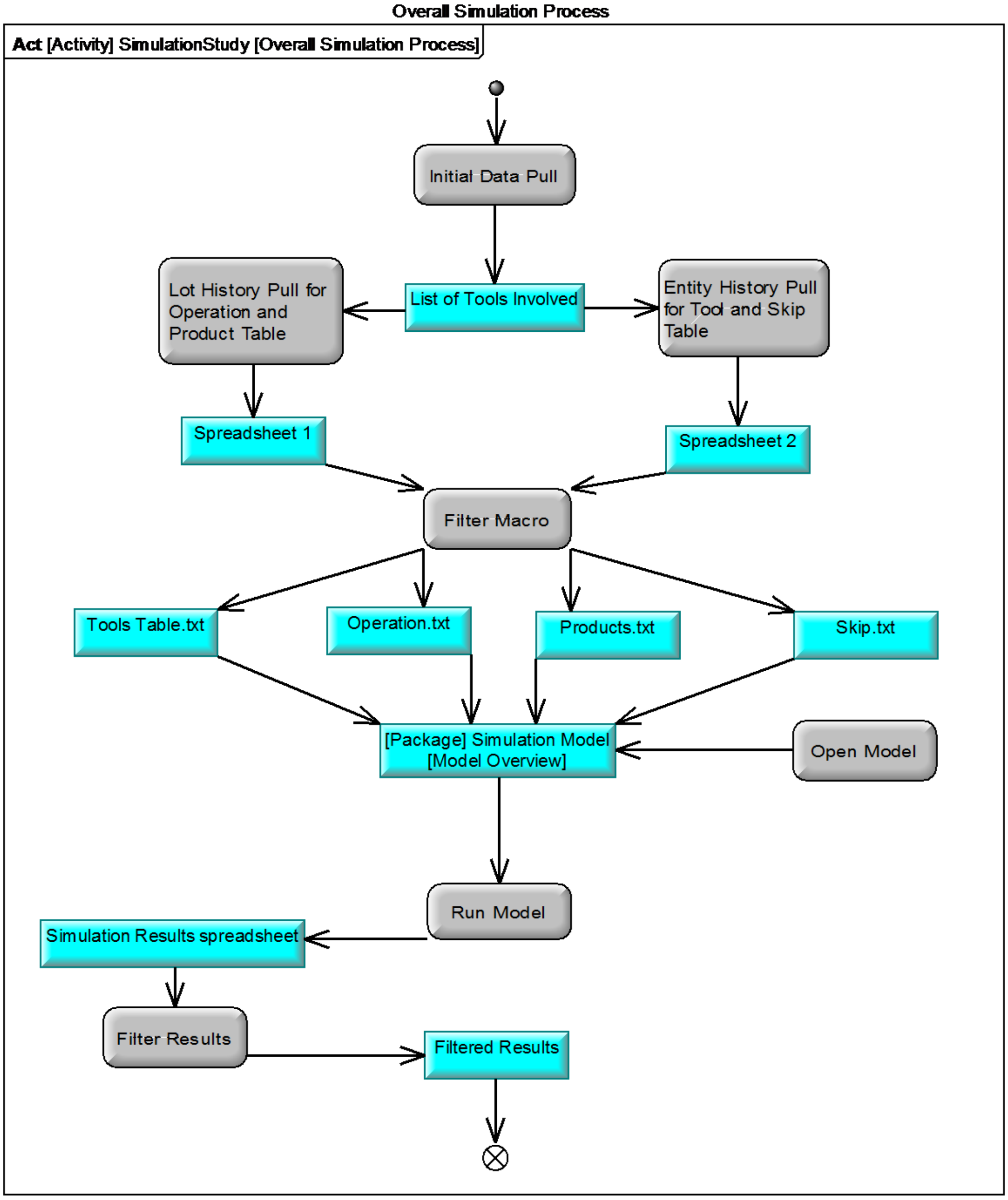


Figure 4-4: Activity Diagram - Simulation Model Flow

The flow through Figure 4-4 shows the entire simulation interaction from the initial action until the end point. Through a number of different actions and objects the overall interaction is mapped. The relevant objects in Figure 4-4 are represented by the SysML elements that are rectangular while the actions are symbolized by the elements with chamfered corners. These SysML elements are coloured blue and gray respectively in this case of Figure 4-4, but these colours are arbitrary. The action and object elements represent the main pieces with the entire diagram representing the flow of the aforementioned interactions in the simulation.

Here the actions are events that occur during the execution of the simulation model. The objects are the elements of the simulation model that are needed/created for the subsequent actions/objects. This diagram may be difficult to understand to a lay person, but would greatly aid the simulation modeller when referring to how the simulation model runs, or even when presenting to the client.

Mostly throughout this particular diagram, the actions result in the generation of objects. However, this is not always the case as can been seen from the four objects "Tool Table.txt", "Operation.txt", Products.txt" and "skip.txt". These feed into the object "Simulation Model [Model Overview]" automatically as part of the operation of the "Simulation Model". The action "Open Model" also occurs prior to the object "Simulation Model [Model Overview]", entering from a different flow. The object "Simulation Model [Model Overview]" is a representation of a more detailed activity diagram which is shown in Figure 4-9.

The information from Figure 4-4 could be represented in a block diagram or alternatively a SysML sequence diagram. In a block diagram the information presented in Figure 4-4 showing what activities occurred in order to affect the next object/activity may be lost. A sequence diagram could show when each activity occurred at what stage of the flow. A sequence diagram could be created to accompany this activity diagram but would have required more in depth knowledge of how the overall simulation process occurred.

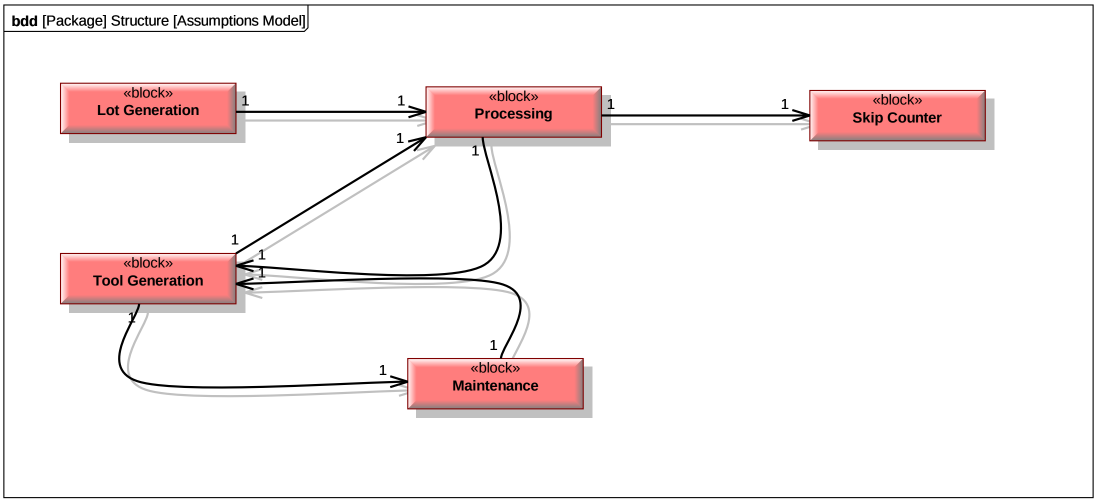


Figure 4-5: Block Diagram - Assumptions Model

This SysML Block Diagram (bdd), Figure 4-5 shows the assumptions model, the structure of which is important for a modeller when it comes to the point of programming the model.

In this particular case the simulation model has been split into 5 different sections that need to glue together in order to create the completed model.

In this case study both Lots and Tools needed to be created to be representative of the real world system, hence the "Lot Generation" and "Tool Generation" Block. It was decided that the maintenance of the tools would need to be represented within the simulation model leading way to the "Maintenance" Block. The "Model Processing" block is where most of the actual processing of the lot and batches will take place while the "Skip Counter" block represents the section where quality control is monitored.

Between each block is an interaction or interactions when there is more than one block connected. These signify the movement through the simulation model. For example in the real world system a *Lot* and *Tool* will be required before any processing takes place. In the model this too is the case. When the lot has been processed (by the tool) the tool returns and may be used again, hence the arrow from the "Processing" Block to the "Tool Generation" Block. The situation regarding the "Maintenance" Block is similar as the tool enters the maintenance loop, as it requires maintenance, is altered and returns ready to be used in the processing section of the simulation.

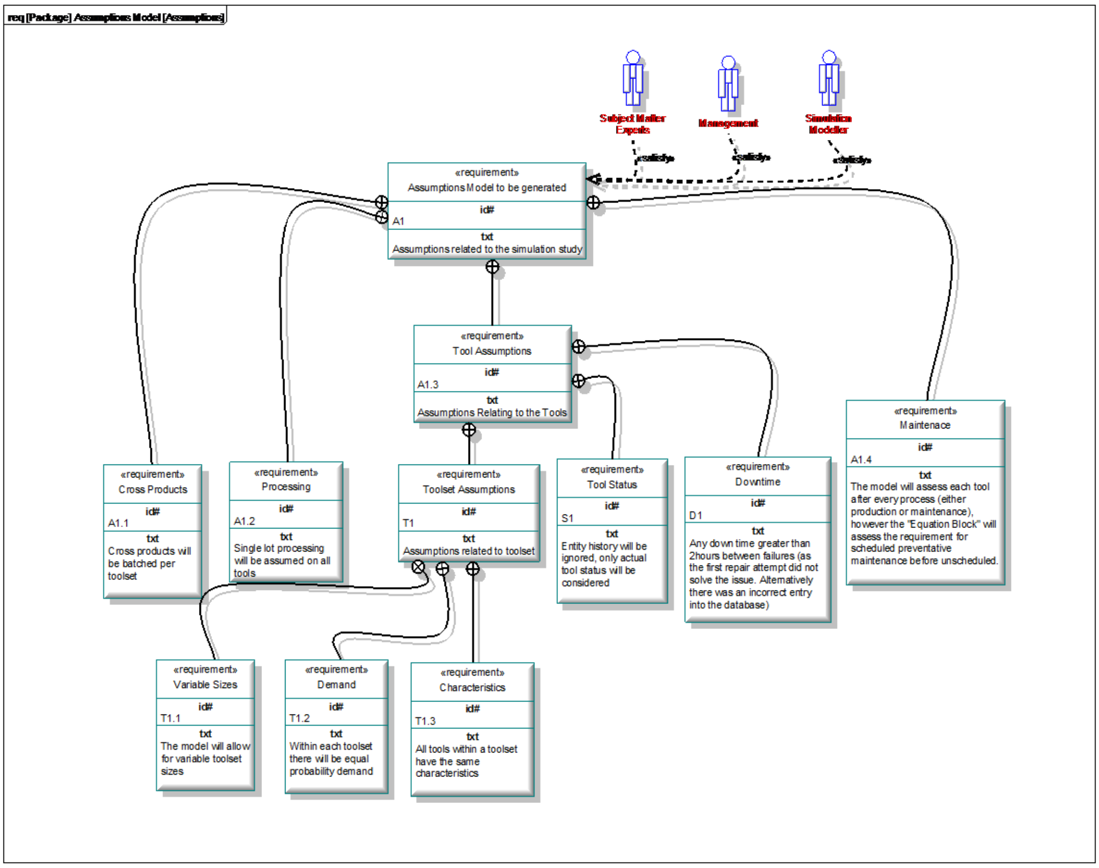


Figure 4-6: Requirements Diagram - Assumptions about Real World System

Similar to Figure 4-2, Figure 4-6 is also a requirements diagram. Within this diagram the requirements relating to the Assumptions model are contained. The parent requirement is "Assumptions Model to be Generated" (ID: A1) and the people involved that satisfy said requirement are the "Subject Matter Experts", the "Management", and the "Simulation Modeller". The hierarchical requirements show the areas that assumptions have been made about. In the case study assumptions relating to the tools are in the majority, with the "Toolset Assumptions" (ID: T1) requirement having child requirements of its own in the form of "Variable Sizes", "Demand" and "Characteristics" (ID: T1.1, T1.2, T1.3). The ID numbers are sequential, but each is grouped using a letter, i.e. A/S/D/T etc. These letters are chosen arbitrarily. In this case A was chosen for the higher level Assumptions, T for the Toolset assumptions etc.

The diagram in its entirety keeps the assumptions requirements clearly visible including the contained relationships between each parent and child requirements. The requirements can also be presented under the SysML format as a table as seen below in Figure 4-7.

[Package] Assumptions Requirements Table

Id#	Name	Txt	Satisfied By
A1	Assumptions Model to be generated	Assumptions related to the simulation study	«Actor» Management (Formualte The Problem) «Actor» Simulation Modeller (Formualte The Problem) «Actor» Subject Matter Experts (Formualte The Problem)
A1.3	Tool Assumptions	Assumptions Relating to the Tools	
A1.2	Processing	Single lot processing will be assumed on all tools	
A1.1	Cross Products	Cross products will be batched per toolset	
A1.4	Maintenace	The model will assess each tool after every process (either production or maintenance), however the "Equation Block" will assess the requirement for scheduled preventative maintenance before unscheduled.	
S1	Tool Status	Entity history will be ignored, only actual tool status will be considered	
D1	Downtime	Any down time greater than 2hours between failures (as the first repair attempt did not solve the issue. Alternatively there was an incorrect entry into the database)	
T1	Toolset Assumptions	Assumptions related to toolset	
T1.3	Characteristics	All tools within a toolset have the same characteristics	
T1.2	Demand	Within each toolset there will be equal probability demand	
T1.1	Variable Sizes	The model will allow for variable toolset sizes	

Figure 4-7: Assumptions Requirements Table

The information contained within this SysML Requirements Table (Figure 4-7) is the same as the requirements diagram in Figure 4-6. The information is sorted by ID tags and any supplementary information is also included such as the “satisfy connections”.

4.1.4 Assumptions Model Validation

For the simulation study in question here, an activity diagram has been generated to represent the process of Assumptions model validation.

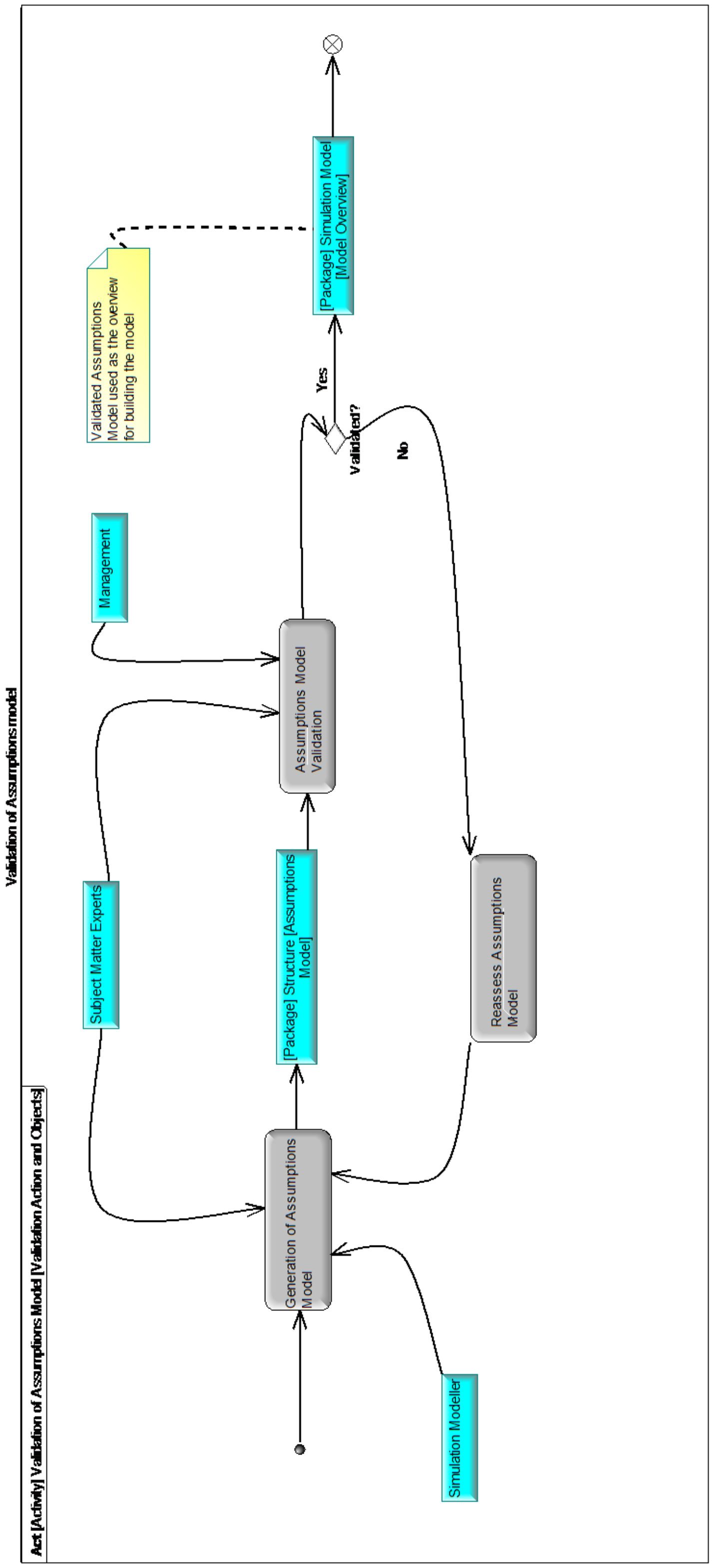


Figure 4-8: Assumptions Model Validation

Like the previously mentioned activity diagram (Figure 4-4), Figure 4-8 represents the process of the assumptions model validation through a number of actions and objects. This particular diagram contains the same actors as seen in Figure 4-2 and Figure 4-6, however these are now depicted here as objects as dictated by the SysML specification. Note Figure 4-8 is a separate diagram made up of its own elements. Both the “Simulation Modeller” and the “Subject Matter Experts” are involved in the action “Generation of Assumptions Model”. It is then the “Subject Matter Experts” and “Management” that are involved in the action of “Assumptions Model Validation”. When this action takes place a decision is sought as to whether the assumptions model has indeed been validated or whether it must be reassessed. Eventually when the management and subject matter experts have decided the assumptions model is valid, the object “Validated Assumptions Model” is the final activity element prior to the activity diagram flow ceasing.

4.1.5 Program The Model

Using block diagrams and internal block diagrams the step of programming the simulation model has been documented. The block diagram shows the interaction of the different sections of the model while the internal block diagram drills down to the interactions within each section. Blocks have been used to represent specific components contained within the simulation model. It is the interaction between these blocks that are important to capture in the SysML documentation. Figure 4-5 (The assumptions model) represents how the modeller intends to create the model and what different sections they intend to include, while Figure 4-9 is what was created and the various flows from each section.

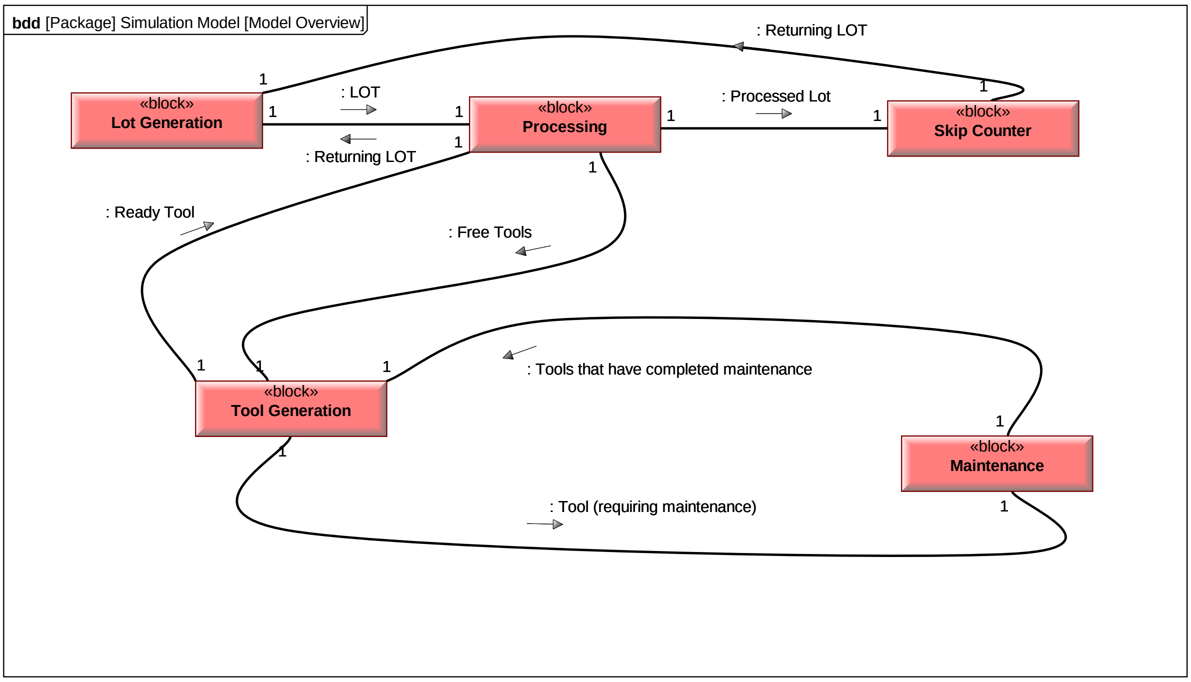


Figure 4-9: Block Diagram - Model Overview

Figure 4-5 is the basis for the structure of this Block diagram in Figure 4-9. Within this Block Diagram the actual Block flows have been included. It is seen that from "Lot Generation" a "Lot" enters into the processing block while a "Ready Tool" enters from the "Tool Generation" Block. From there a "Processed Lot" enters the "Skip Counter" while a "Used Tool" returns to the "Tool Generation Block". The Block flow to and from the "Maintenance" Block has also been specifically defined in this Block Diagram, placing more in depth information into the documentation at this point. The interactions have been included here to highlight what specifically happening in the simulation. This differs to the previous block diagram as Figure 4-5 shows what areas have to be modelled; it doesn't delve into how they will be modelled.

Further on, breaking the model down Internal Block Diagrams (IBD) were created showing the flow within each of the specific blocks from the model overview (Figure 4-9). Each block in Figure 4-10 is representative of a block from the simulation software used while completing the simulation study. In the simulation software the blocks have specific functions and attributes. In these internal block diagrams it is only the block flow and connections that are described. SysML does support populating these Blocks with further information which will be discussed later.

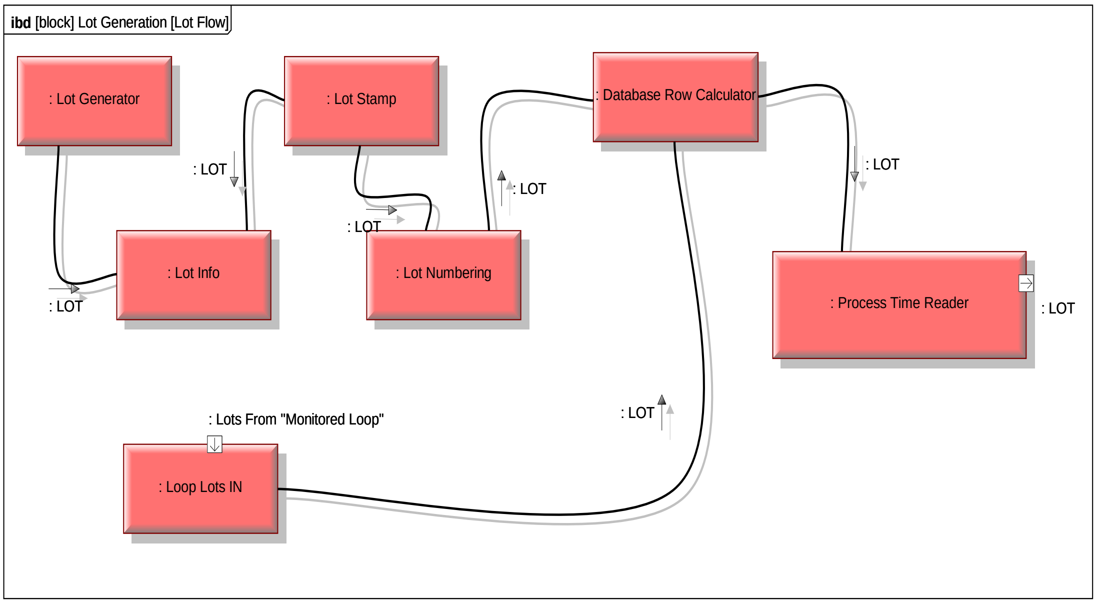


Figure 4-10: Internal Block Diagram - Lot Generation

Lots enter this overall block from either “Lot Generator” or “Loop Lots IN”. The Lots make their way through the simulation until they reach “’Process Time’ Reader”. Here the process time related to each block is read and the “Lot” exits. Connected to the ’Process Time’ Reader” Block is a flow port showing where “Lot[s]” exit the internal block diagram at this point. Similarly there is a flow port on the element “Loop Lots IN” showing an entrance of “Lots from the Monitored Loop” (i.e. another section of the simulation model”.

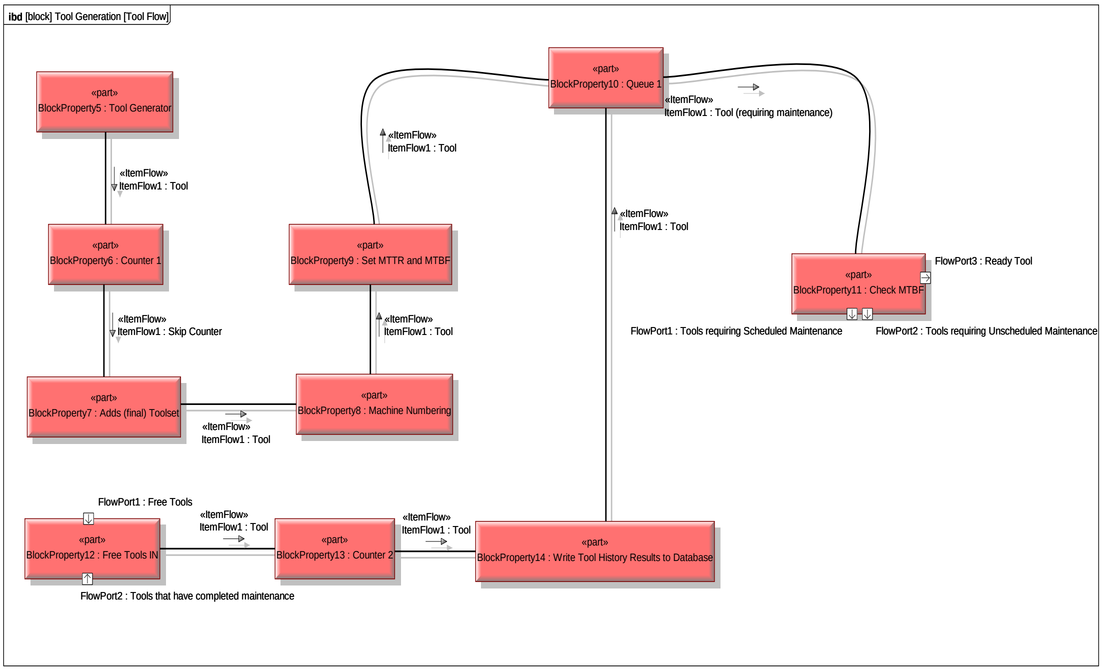


Figure 4-11: Internal Block Diagram - Tool Generation

The Internal Block Diagram (IBD) for the Block "Tool Generation" is similar in construction to the IBD for "Lot Generation". The Block parts here however have different functions. The "Free Tools IN" block has two Flow Ports where tools can re-enter the "Tool Generation" Block. "Free Tools" enter from the "Processing" Block and "Tools that have Completed Maintenance" return from the "Maintenance" Block. These ports show the information that has already been shown in Figure 4-9, i.e. the return flow from the said Blocks.

There are 3 out Flow Ports on the Block "Check MTBF". Depending on which the Tool (seen as the item flow between blocks in Figure 4-11) leaves through decides the next section of the simulation model that it enters, i.e. which internal block diagram it enters.

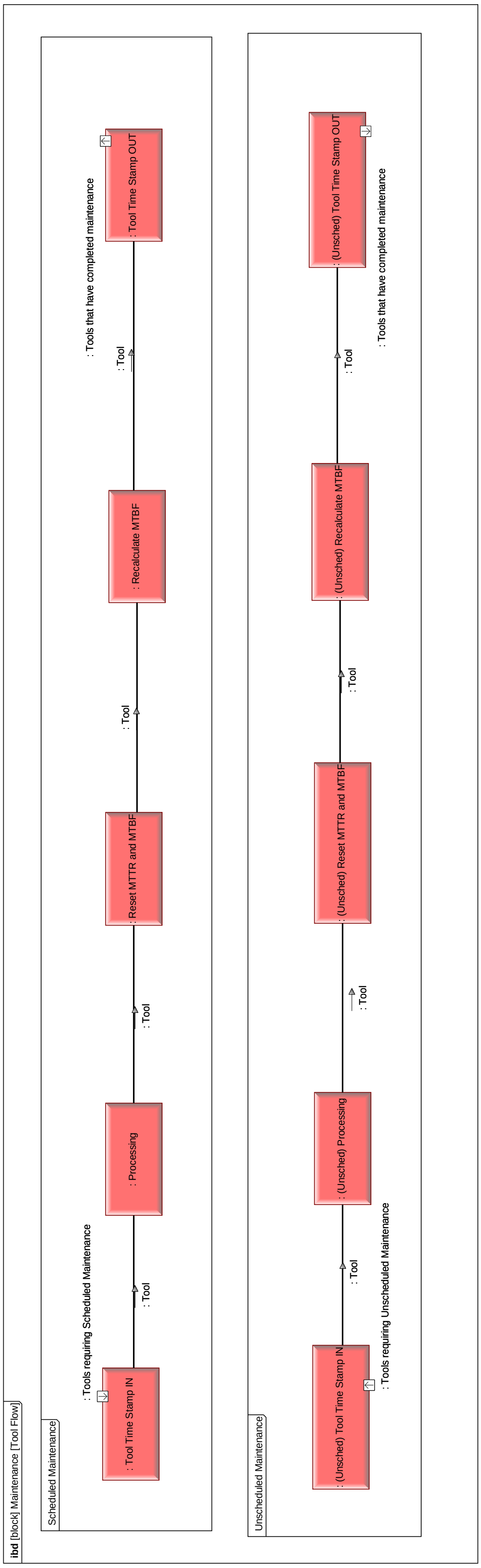


Figure 4-12: Internal Block Maintenance – Unscheduled and Scheduled Maintenance

The Blocks present in the SysML diagram representing the scheduled and unscheduled maintenance (Figure 4-12) are the same apart from the in Flow Port governing what enters in.

Having two sets of blocks in the simulation model was necessary as the simulation needed to represent the real work system whereby a tool:

1. May require maintenance, depending upon normal mean time before failure i.e. scheduled maintenance.

2. May not require any maintenance therefore exiting IBD Tool Generation through the out *flow port* “Ready Tool”.

3. May require maintenance, depending on the mean time before failure, when the MTBF has been set randomly at the Block “Set MTTR and MTBF” in i.e. Unscheduled Maintenance.

The Internal Block Diagram “Processing” documents where the processing occurs within the simulation model. The processing was defined by what “Lot” and what “Ready Tool” enters at each occurrence. Figure 4-13 an Internal Block Diagram showing the blocks that make up this section in the simulation model.

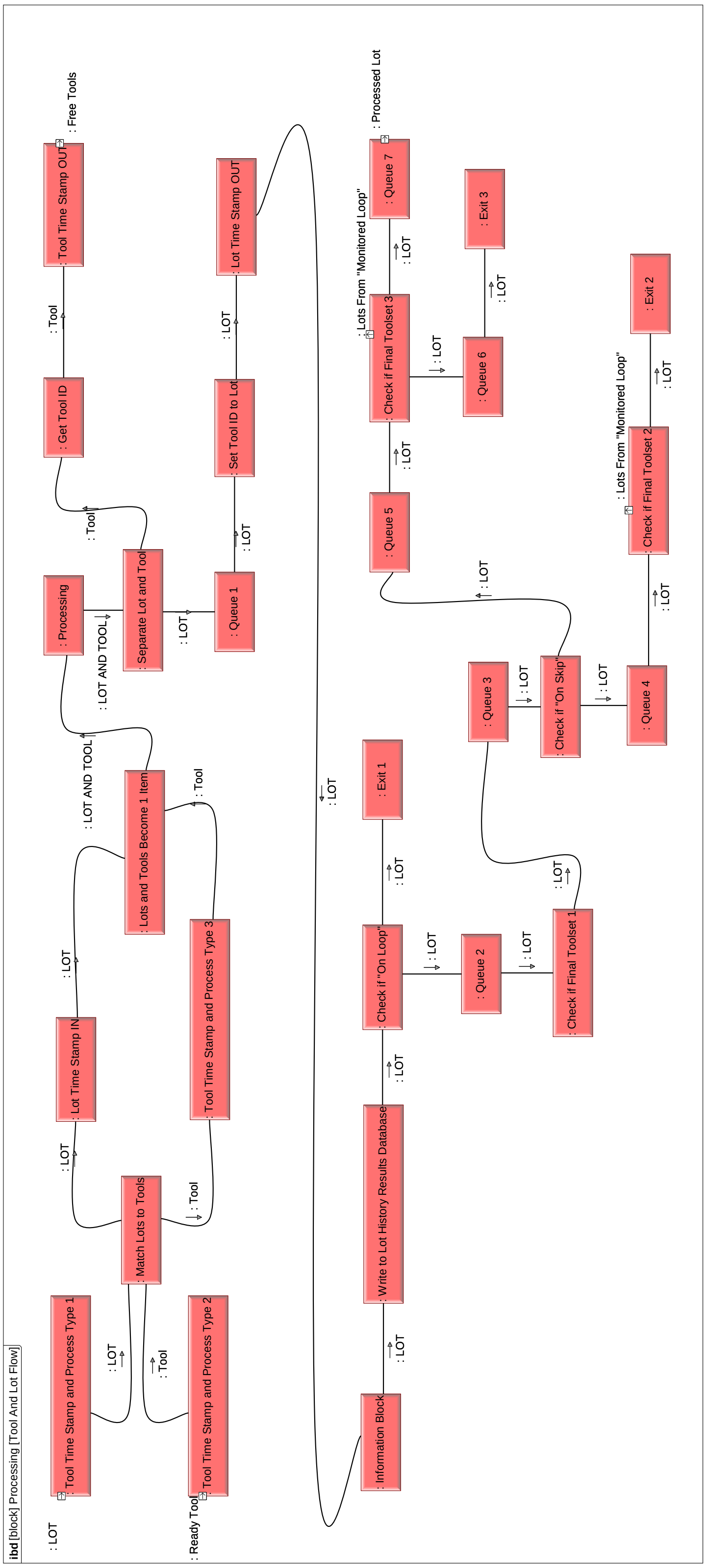


Figure 4-13: Internal Block Diagram – Processing

The “Tool” and “Lot” enter from their respective Blocks (Figure 4-10 and Figure 4-11) and are matched to one another. They become one item and are processed at “Processing” Block for a certain amount of time based on previously set parameters. They are separated afterwards and the “Lot” passes through certain screening to see whether that “Lot” is required for the ultimate goal of the simulation cycle. The “Tool” returns to the “Tool Generation” Block to be used again. The design of the simulation model models the scenario of tools being reused while processed lots continue through the manufacturing facility.

This particular simulation study deals with a manufacturing line where the lots require several bouts of processing by different tools before the manufacturing in its entirety is complete. This is the reason that “Lots from ‘Monitored Loop’” out Flow Ports are present creating a loop back where these same lots can be processed again until completion.

There are many “Queue” Blocks represented in Figure 4-13. These are needed in the simulation model as at these junctures build up may occur which in turn could skew the output results of the overall model.

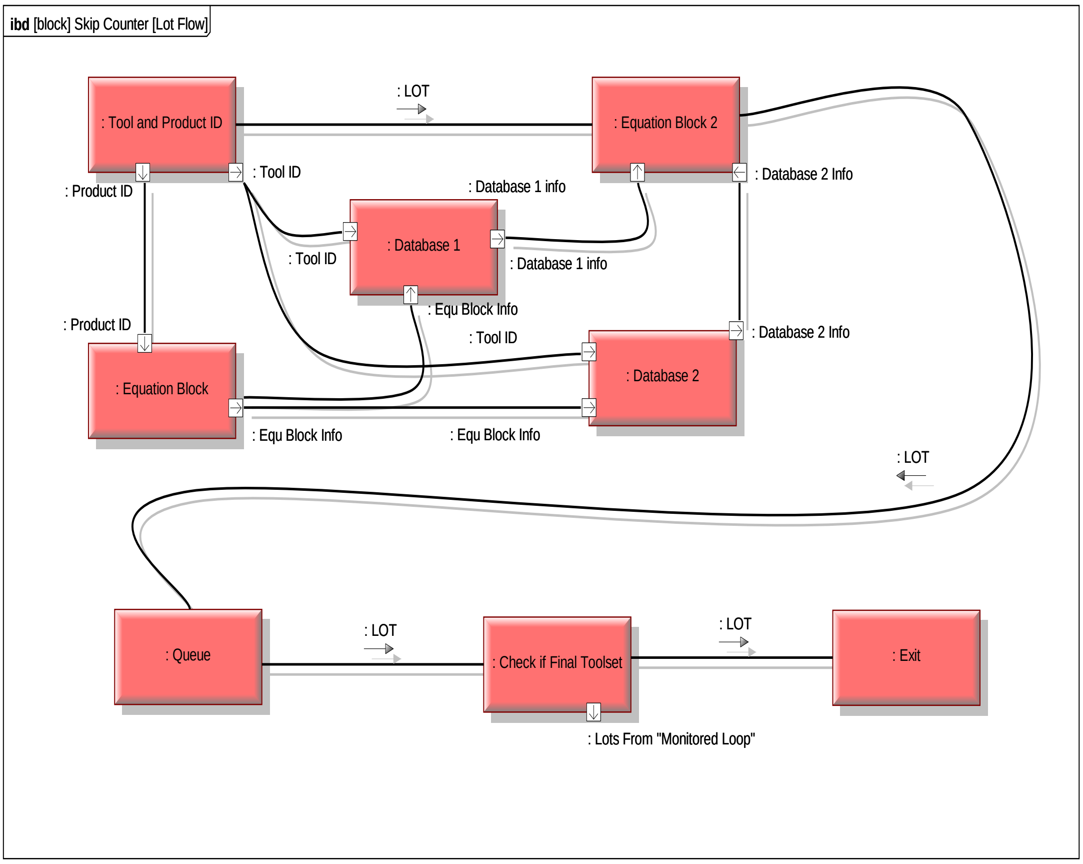


Figure 4-14: Internal Block Diagram - Skip Counter IBD

Figure 4-14 represents the Skip Counter in the form of an Internal Block Diagram and again each Block corresponds to a block from the simulation software package used by the simulation modeller. The "Processed Lot" enters through the IN flow port at the "Tool and Product ID" Block. The "Processed Lot" continues through the simulation model but *information* exits throughout Flow Ports in the form of "Product ID" and "Tool ID". Both pieces of information ("Product ID" and "Tool ID") exit from the same Block but are separate. It would have been possible to create another Internal Block Diagram for "Tool and Product ID" at this point. In union with the Databases and Equation Blocks the information is recorded and it is decided if the "Processed Lot" exits through the flow port "Lots from 'Monitored Loop'" or whether it continues and eventually exits the simulation model.

The internal block diagrams presented here show exactly what is included in the simulation model. The information contained may not contain very detailed information but is documentation to be stored by the modeller on how the simulation model was actually created. It is up to the simulation modeller on how detailed they want to make this Internal Block Diagrams. If it was require or wanted they could create internal block diagrams for some or all of the presented blocks from Figure 4-10 to Figure 4-14.

4.1.6 Validation of Simulation Model

Figure 4-15 showing the activity flow of Simulation Model Validation is analogous with Figure 4-8. The involved actors are objects in the activity diagram as the SysML specification doesn't have an "actor" element for activity diagrams.

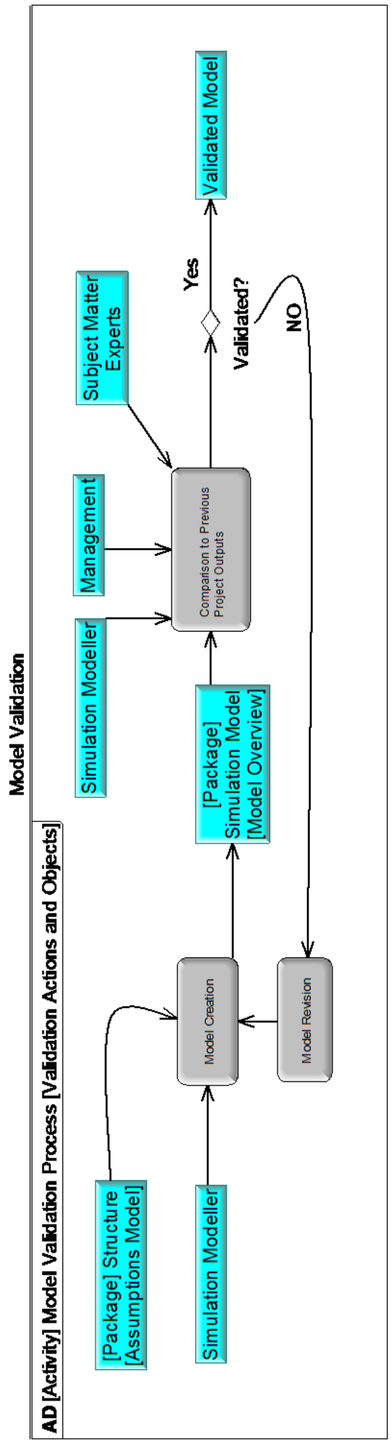


Figure 4-15: Activity Diagram - Simulation Model Validation

The model creation is now governed by both the “Simulation Modeller” and the “Assumptions Model” which was created during the Generation of the Assumptions Model Stage.

The validation of the simulation model was achieved by comparing the outputs to a similar, previous project that had been conducted in the same area. The “Simulation Modeller”, “Management” and “Subject Matter Experts” decided together upon the validity of the model, and whether any revisions were needed. After that decision the activity flow comes to the “Validate Model” object then terminates.

4.1.7 Experimentation and Output

A Use Case Diagram was created to represent what occurs during the later stages in this case's simulation study (see Figure 4-16). The four actors, that have been seen in other SysML diagrams are represented again here as people who may use the simulation model.

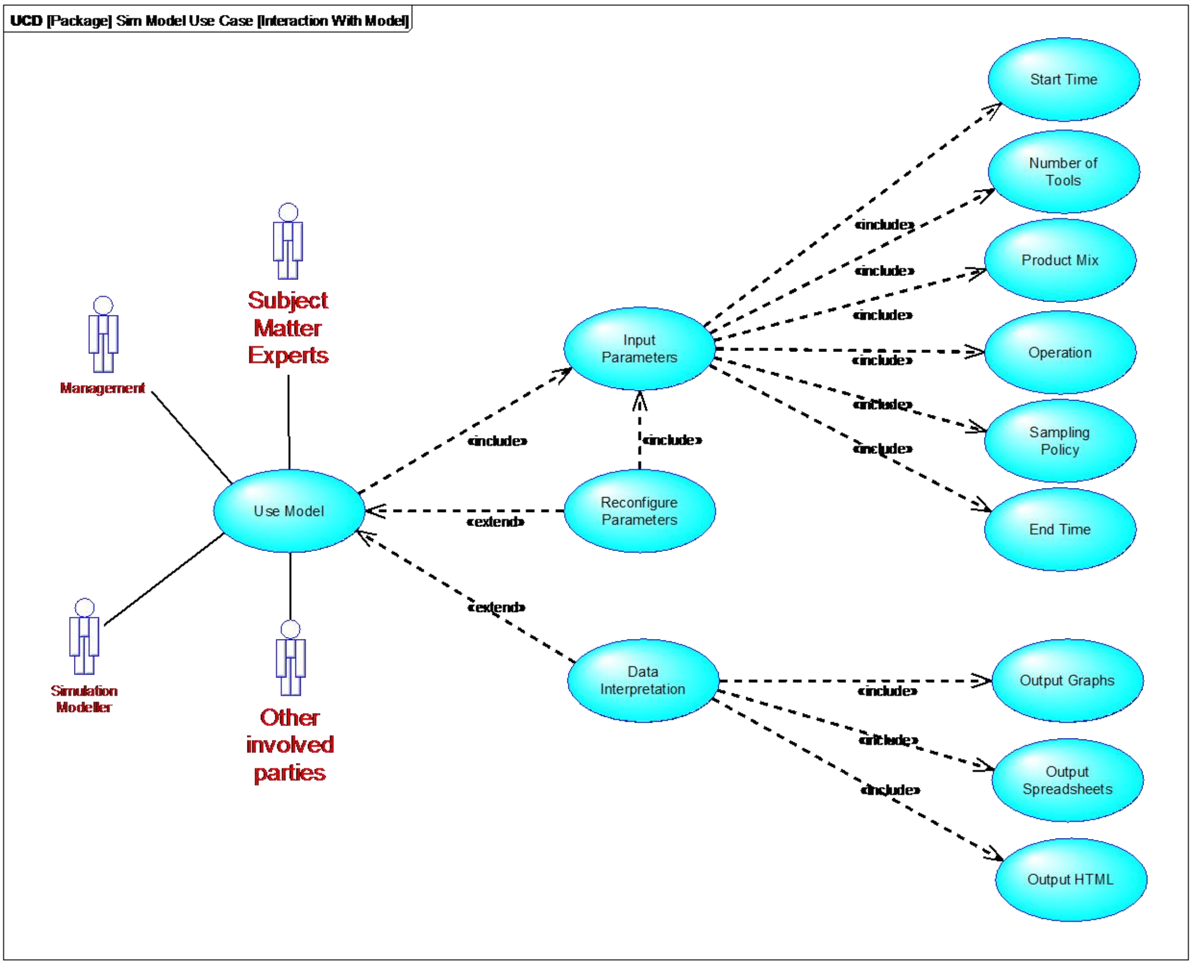


Figure 4-16: Use Case - Simulation Model Use

Figure 4-16 contains the *Use Case Nodes,* as well as the related *Association*, *Extension* and *Inclusion Paths.* The association connections show who will use the model (i.e. the actors). The include connections show that when a certain base use case is performed the included use case is also performed. The extend connection show the use cases that are not considered part of the normal use case but describes and exception in behaviour in the interaction, e.g. Every time the model is used "input parameter" is also used. "Reconfigurable Parameters" aren't always used as the model has to be run first and then the parameters can be reconfigured.

This simulation study was commissioned with the aim of creating a flexible simulation model. Figure 4-16 shows the case of an actor using the simulation model. For example by using the model the actor "Management" must also "Input Parameters" which are "Start Time", "Number of Tools", "Product Mix", "Operations", "Sampling Policy", and the "End Time". When the model completes its run it will output a certain amount of data in the form of graphs, spreadsheets and HTML, which the "Management" will have to interpret. This can be seen on the diagram as the "Data Interpretation" is connected to the "Use Model" node by an extension path, with "Output Graphs", "Output Spreadsheets" and "Output HTML" connected through three inclusion paths.

The Use Case node "Reconfigure Parameters" is also joined to "Use Model" via an extension path. This part of the model represents an actor reconfiguring the model prior to "Data Interpretation". If an actor decides upon reconfiguring the model it is necessary for them to input the changeable parameters again, hence the inclusion path between "Reconfigure Parameters" and "Input Parameters".

4.2 Independent Assessment of SysML Diagrams

After the application of SysML as an aid to help in the documentation of the simulation study an array of engineers and people of a technical background were asked to comment on the generated diagrams and score the generated diagrams as the case studies had been scored in Chapter 3. It was important to investigate how both simulation analysts and engineers from the industry reacted to SysML as a graphical aid to the study's documentation as well as having it assessed by someone who wouldn't be particularly familiar with simulation. It is important to note that it was a small cross section of people that were presented/surveyed with the created SysML diagrams. This in turn cannot be used conclusively as a survey of the entire simulation industry. Instead this assessment makes it easier to gain an understanding of how technically minded/educated people react to SysML as a graphical aid and potentially highlight the areas of SysML that worked well in aiding documentation. This survey may point towards how the simulation industry might react to SysML when used in conjunction with their own simulation studies and/or for presenting documentation to their relevant clients.

Five assessors were chosen to score the generated documentation and comment on how well they thought the SysML diagrams acted as a graphical aid for the simulation study in question. All the assessors came from a technical background. Each had a differing understanding of simulation and none had a particular knowledge of the Systems Modelling Language. Assessors 1, 2 and 4 were asked to participate as they had a great understanding of simulation in the manufacturing industry and had all (separately) produced simulation studies for clients as investigations for client's systems. Assessor 3 had a base in statistics and was well versed in dealing with statistical models. The engineer's knowledge extended somewhat to the simulation industry and they understood the context of simulation studies in manufacturing. Assessor 5's was chosen as they had a background in computer programming and physics. Their participation allowed an insight from a person who had a technical background but wouldn't be directly familiar with simulation especially in the manufacturing industry.

The assessors were supplied with the 15 SysML diagrams that are seen throughout this chapter (sections 4.1.1 - 4.1.7). Minimal information of the simulation study was given and the assessors were asked to score the overall documentation under the headings:

i. Communicative Ability.

ii. Structure.

iii. Clarity.

iv. Ease of use,

v. Ease of understanding.

vi. Ease of Reuse.

An explanation was also provided to the assessors of each area as per the same explanations in section 3.2. This allowed for the scoring to be comparable to the documentation previously from Case Studies A, B and C.

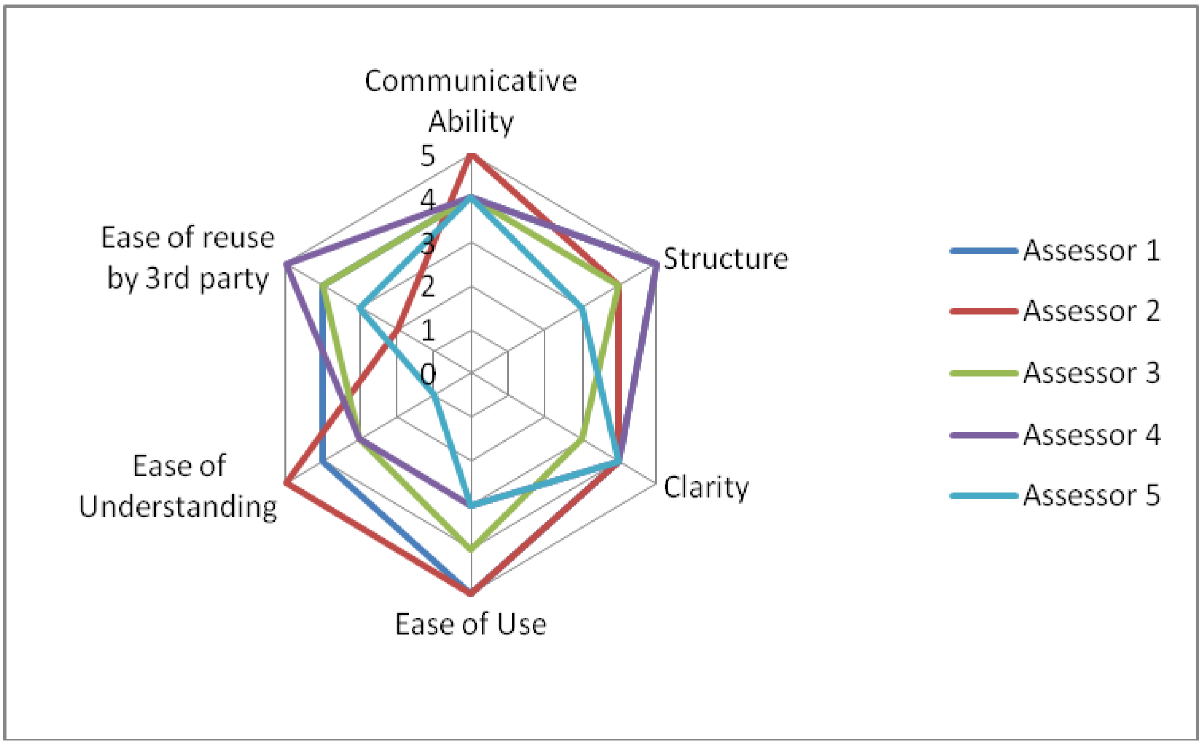


Figure 4-17: Independent Assessment of SysML as a graphical aid for Simulation Studies

Figure 4-17 plots the scoring of each of the independent assessors on a scale of 0 to 5 (0 being Nonexistent, 5 being Excellent) again as per section 3.2. The outcome of the independent assessments will be discussed in section 5.4 following. Each assessor also provided comments on the generated SysML diagrams underlining what they found particularly good about them and which diagrams were less intuitive. The comments can be found in full in Appendix A.

By independently assessing the generated documentation it was possible to investigate how SysML could/would be put to use in a simulation study as an aid and as a tool to communicate information between client and simulation team/analyst. It is important to know how the SysML documentation is perceived by people of a technical background for this reason.

5 Discussion

5.1 Introduction of Discussion

Often documentation isn't a particularly high priority during a simulation study. Documentation is needed in a simulation study. It may be costly to implement good documentation but bad or a lack of documentation can be more costly [21]. This chapter deals with the findings from applying documentation with a definite structure to a simulation study, as well as a discussion on what problems were posed and solved during this project. Without good documentation practice important data/information could be lost in the reams of information collected at the beginning of a study [36]. We have seen in chapters 2 and 4 that it is possible to create SysML diagrams to capture specific information relevant to a simulation study and have this information relayed to technical people.

Firstly however, the case studies presented previously are discussed. The gaps that appear in these case studies' documentation are also assessed highlighting what areas should be concentrated upon when applying good documentation practice to a simulation study. Within this chapter the approach of actually creating the SysML diagrams to assist in documentation will be scrutinized and an overview of how using a third party program specifically designed for working with SysML can be a major aid for documenting a simulation study.

5.2 Case Studies Comparison

The areas assessed and presented using the radar diagrams were those areas that were believed to be most use with regards to documenting a simulation study and were easily comparable over these three particular case studies; these could be potentially applied to other simulation studies in the future. From the literature review these were the areas that were pivotal to creating a successful simulation study.

Methodology was chosen as a topical area as there is no current standard on how to form and complete the documentation for a simulation study. By taking three separate cases and assessing them under the same topics, some insight could be gained into how different modellers, analysts and teams approach their documentation as a whole.

The other areas assessed followed loosely the steps needed to create a successful simulation study (see Law [24]). The reasoning behind basing the assessment on this was for any simulation study to be successful the simulation must follow these steps.

It appears that in the majority of publicised material, available in the simulation industry, there is a consensus that simulation studies should follow certain steps to completion. In reality these steps seem to be skimmed over or rounded only to be revisited (or ignored until needed). Again there is no standard on how to approach simulation studies let alone their documentation, but many resources can be wasted ignoring the steps of a simulation that the industry's peers have highlighted time and time again.

For the most part the involved parties were not overly concerned with the documentation, but rather more concerned with creating a working model that was going to output good results for their clients. This can be a common trap to fall into. Inherently good documentation will assist the simulation study and in turn lead the better results being outputted for the client involved. The following sections delve into the comparison of each case study in more detail.

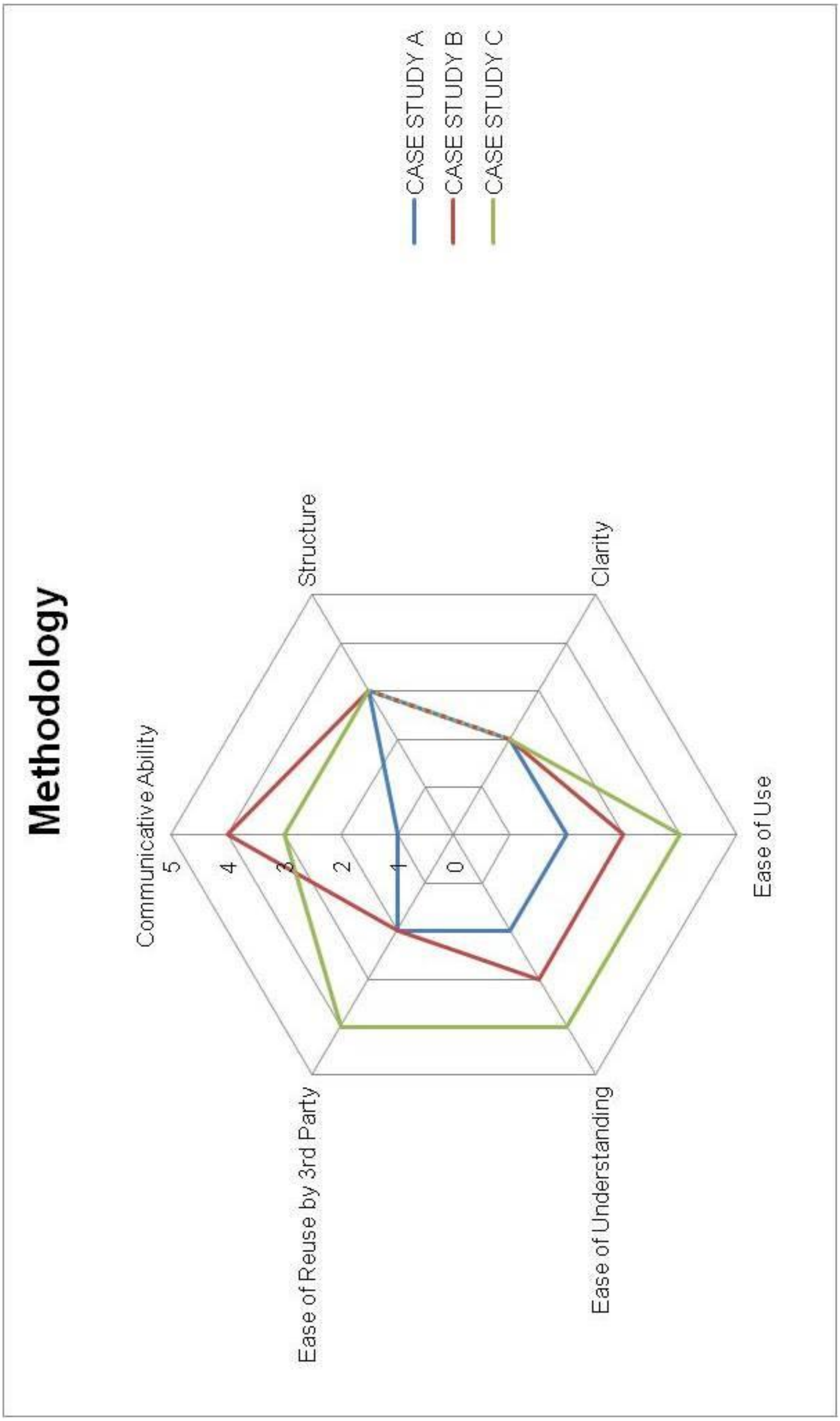


Figure 5-1: Methodology Assessment Comparison

5.2.1 Methodology Assessment Comparison

Figure 5-1 displays the 6 separate areas of considered in each case study (see Chapter 3) with respect to methodology. The first area to be assessed in the case studies was how the simulation modeller or simulation team approached the overall documentation for each stage in the simulation study.

Across the three case studies it is seen that a fair structure was common while the clarity lacked to some extent. Highlighting the point that the documentation was created for nobody else's use (other than the person's creating it) there was no need for the information to be extremely accessible or easy to find as the involved parties already had a good understanding of it.

Case study A's documentation wasn't set up to communicate the information well hence the reason that the ease of use and understanding scored lower than Case Study B. Case Study C's documentation was short and to the point. The contained information was much easier to access and interact with, which is sharp contrast to the large amounts of documentation present in Case Studies A and B. The documentation from any study should act as an aid to the entire simulation project. Users and involved personnel shouldn't get bogged down in large amounts of notes and files. The simplicity of the documentation in case study C leads the way for the involved people to use is easily; the less complex the documentation the easier it is to use and implement. Packing the documentation with too much detailed information could have the opposite effect. A balance must be found between what the documentation includes and how easily it can be used. The documentation may be easy to use only because it contains very little that is applicable to the simulation study. There was scope within case study C to contain much more detailed information, that wouldn't cause it to be difficult to use, re-use or understand by anybody involved with the study in question.

With respect to reusing any of the methodologies Case Study A and B scored lower again than Case Study C. The concise nature of Case Study C would make it easier for a person outside of the project to apply the methodology to their own study. There was nothing too intricate about it unlike the other cases.

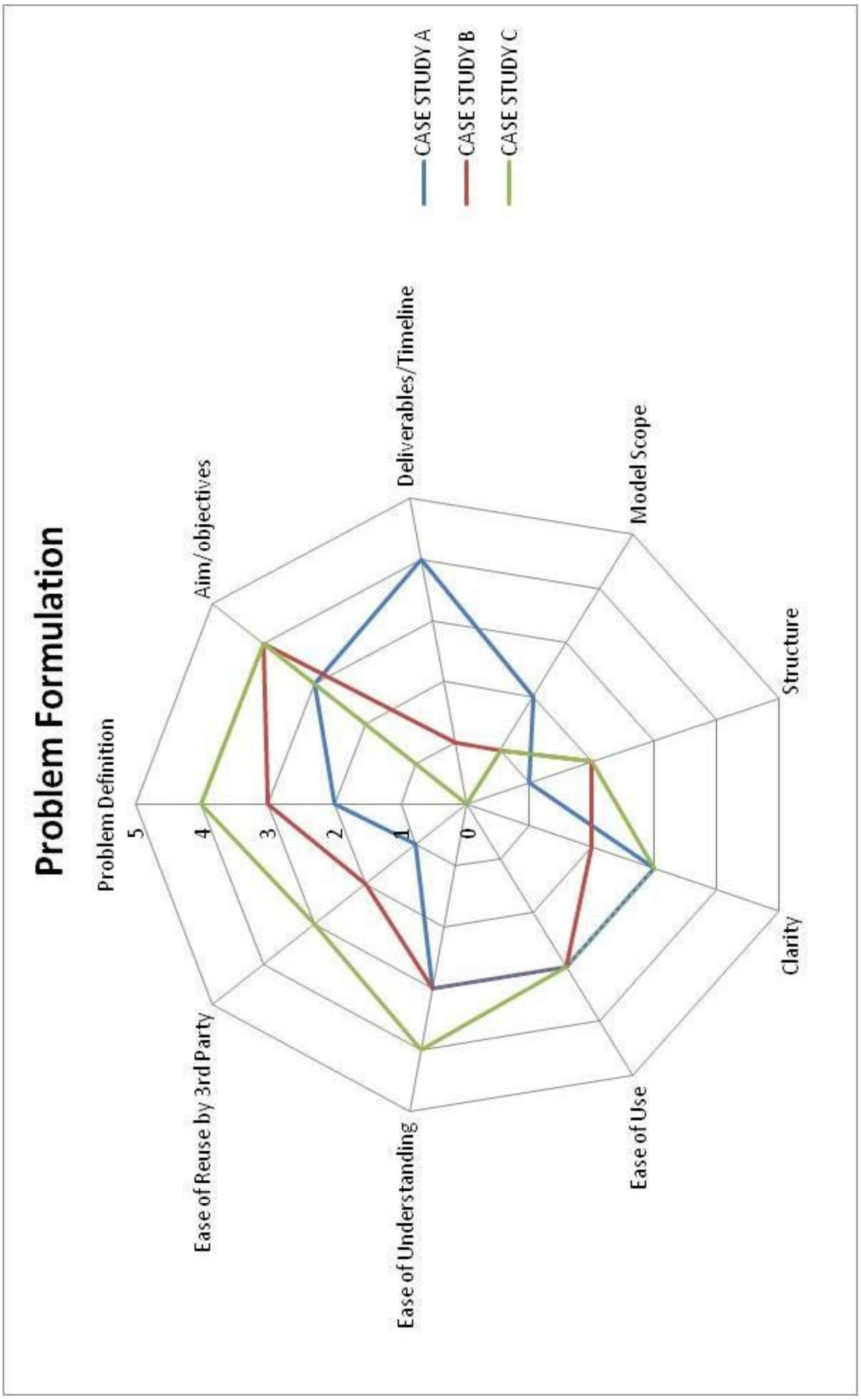


Figure 5-2: Problem Formulation Assessment Comparison

5.2.2 Problem Formulation Assessment Comparison

Each of the problem formulation assessments differed between case studies as seen by the radar diagram Figure 5-2. Case Studies A's client's wanted a clear picture of project cycle which led the simulation modeller to spelling out the deliverables and timeline quite clearly. Case Study B and C waned in this area. Case Study B was quite a long term project which may be one reason that the deliverables and timeline weren't particularly well documented while Case Study C was still in its infancy. It would have been very easy for either the documentation in Case Study B or C to include this information. This information should be included for any client based project let alone a simulation study. The aims of each on the other hand were better documented. While each case study scored differently on how well the problem they were approaching was documented, each gave a good account of what the specific aims were. All the case studies initial meetings revolved around deciding upon what those aims and objectives were to be, but these aims and objects could not be defined without defining the problem at hand. Case Study A's Problem Definition was least well explained. The modeller knew exactly what the problem was that needed solving but the documentation didn't reflect this knowledge. As the simulation study was a long arduous process it is potentially possible that the modeller could lose sight of the problem the aims and the timeline they need to work on. If these are (to some extent) ignored it is possible for the study to evolve adding cost to the entire project.

An area over looked by all the case studies was the Model Scope. When specific documentation exists relating to the scope of the model can again save unneeded model evolution. Similarly it can act as an exact guide for the simulation modeller/team. For example a client may request the modeller or team change a model from a static to a dynamic model, or include extra elements to represent the real world system more accurately mid way through a study. It will often be the case that the client's demands would have to be met, but having a document highlighting the scope, the aims and deliverables etc, will beget easily explaining that was not the original decision for the simulation study.

None of the studies had a particular structure for this part of the assessment. The documentation of each wasn't approached with these headings in mind. However where each lacked structure the other areas of ease of use and suffered.

With regards the ease of use by a third party the problem formulation stage of any simulation study is very specific to that study. Unless one was to conduct an extremely similar study it would be hard to reuse any of the information save only applying the same structure. This is mainly the reason that each of the case studies scored low on that particular assessment.

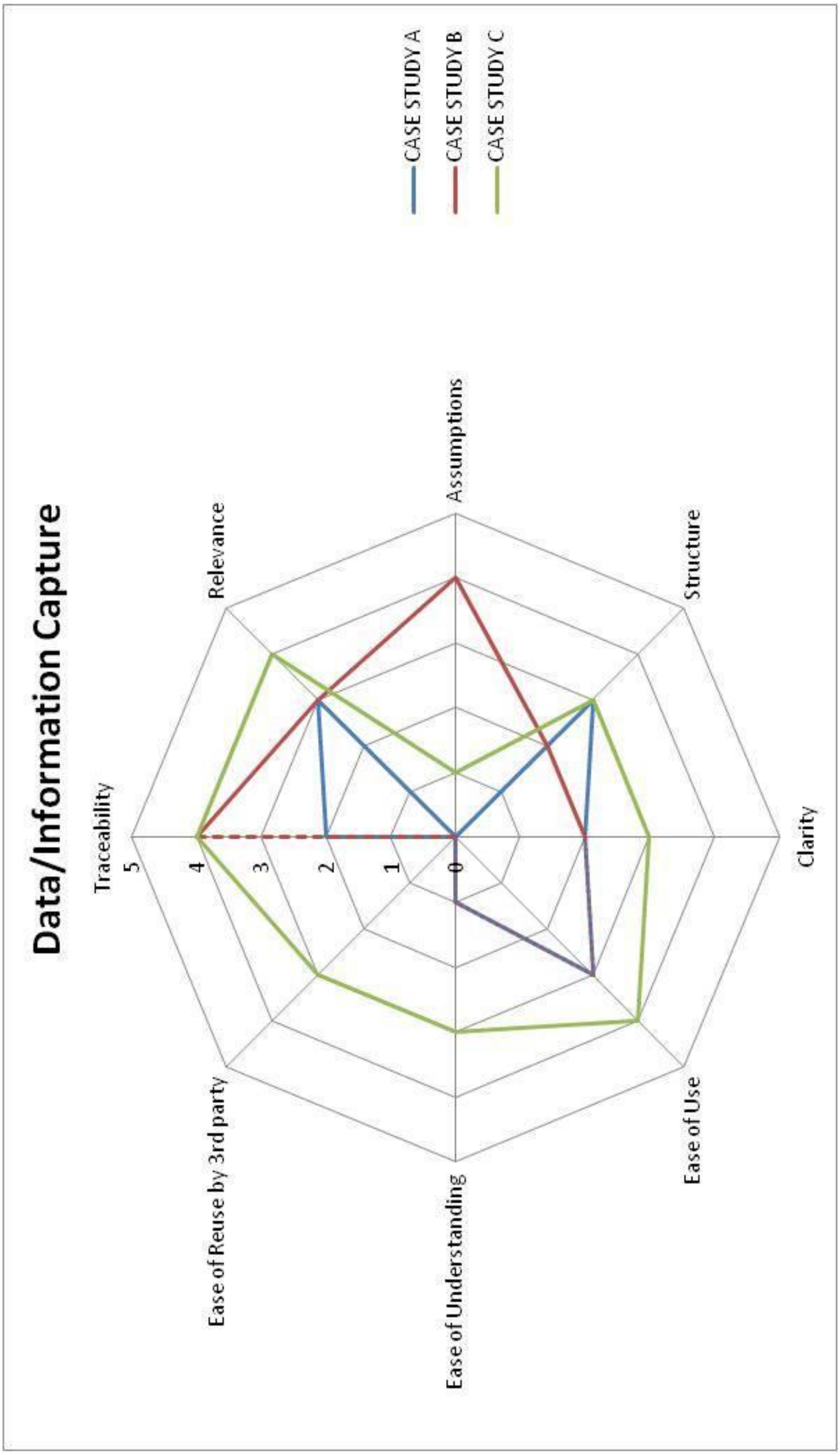


Figure 5-3: Data/Information Assessment Comparison

5.2.3 Data/Information Assessment Comparison

When comparing the three case studies under the heading of Data and Information Capture, it is clear that each performed differently (see Figure 5-3). The documentation for Case study A was sprawling across many different files and folders and included much information that wasn't relevant to the study. More so where the information was obtained wasn't particularly clear for a person not intimately involved in the study. The importance of knowing where the data and information has come from leads to easier validation and proving to your clients that a credible model has been constructed. The simulation modeller from Case Study A knew where the information / data was obtained but there was no paper trail showing this. The involved parties of Case Study B and C saw the importance of using traceable data and information in their documentation. Case Study B scored similarly to Case study A with the exception of the aforementioned Traceability and the assessment of the assumptions area. It was case study B alone that concentrated on including good documentation of the assumptions for their simulation study. The assumptions are a hugely important part of simulation study. Again the involved parties of Case Study A and Case Study C knew what assumptions were being made about their respective simulation studies but this information hasn't been recorded. With regards Case Study A, having these documented would lead to easier validation and greater credibility while in the case of C having the assumptions documented, along with numerous other areas would help get the simulation study commissioned in full by the client.

Once again the succinct nature of Case Study C saw it score better in most areas than the other cases. As the compactness of C's documentation could be seen as a positive, it is equally possible that this section of the short document didn't contain enough information overall for the simulation to be successful. The included information was mostly relevant but more data and information could have been needed so that the simulation could be verified, validation and presented as a credible study to a client.

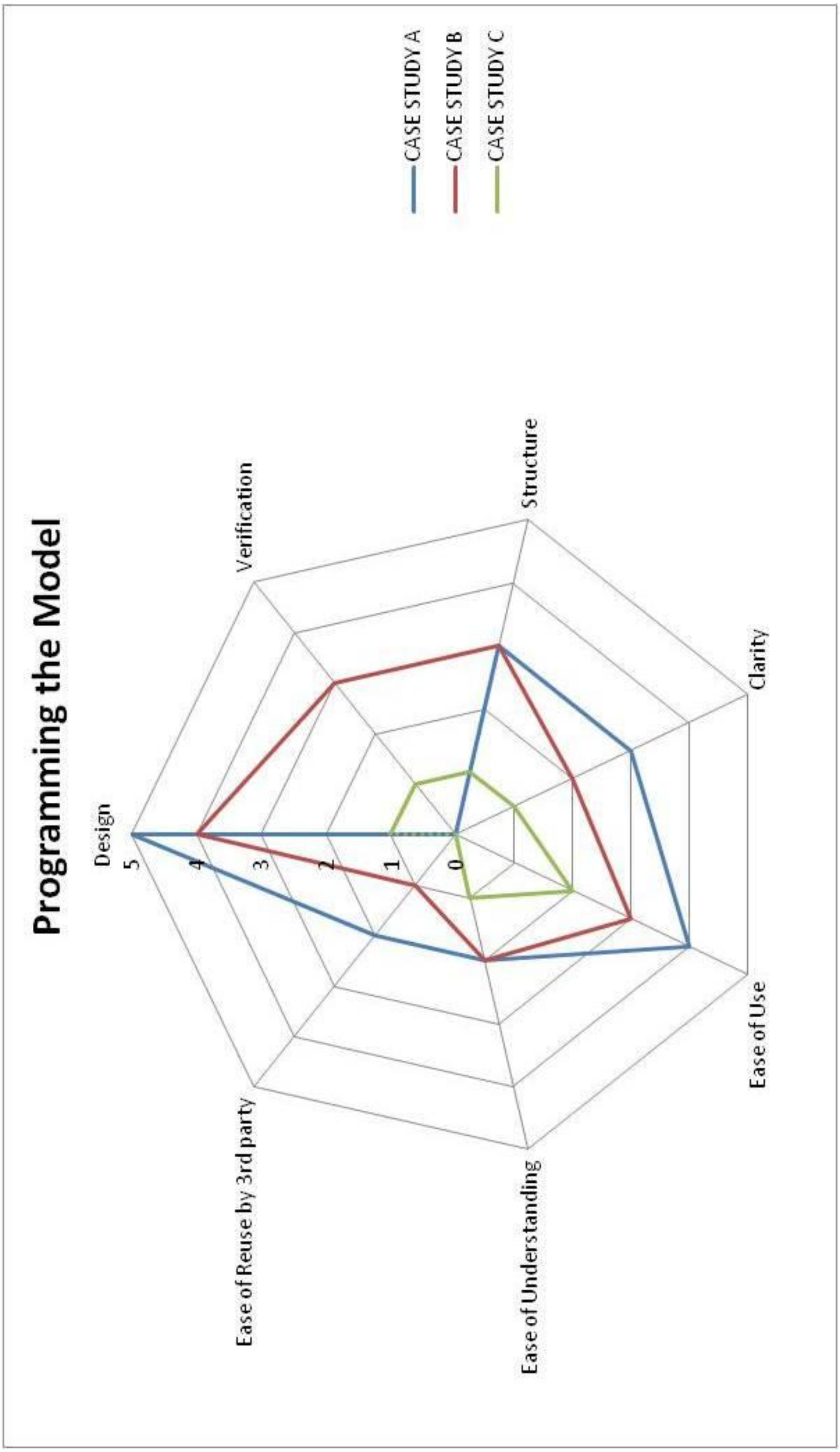


Figure 5-4: Programming the Model Assessment Comparison

5.2.4 Programming the Model Assessment Comparison

Programming the model is more often than not the stage that the most amount of work goes into during a simulation study. This can be seen in the high scores that the Design of both Case Studies A and B score during the assessment. The documentation for each of the cases studies for design was highly detailed explaining exactly what the simulation model was designed for and capable of. The result of which can be seen in Figure 5-4 when comparing it to Case Study C.

Both Case Studies A and B's documentation had a structure applied throughout this stage which led to the specific modeller being able to use the documented material to good effect. Although this phase of the studies' documentation was more concentrated upon, one needed quite a great deal of knowledge pertaining to the study in question to have a good understanding of it.

A third party would find it quite difficult to reuse any documentation from the programming the model phase without the information being quite exact and clear, which wasn't particularly so in the case of A and B. This documentation wasn't created with reusability in mind; with better structure and clarity the potential for reusability could be exploited.

Case study A contained no documentation dealing with how the verification process for the simulation model would take place. Verification is an ongoing process during the simulation model creation but in this case hasn't been recorded.

Case Study C scored lowly across this entire section. The actual simulation model hadn't been created at the time of assessment but importantly this shows the simulation team hadn't any appropriate documentation for this reason. The importance of good documentation though is that it can exist without the specific stage of a simulation study being complete. Case Study C could have documented how the model was to be verified during its creation or could have detailed the design from previous work with the problem formulation and model assumptions.

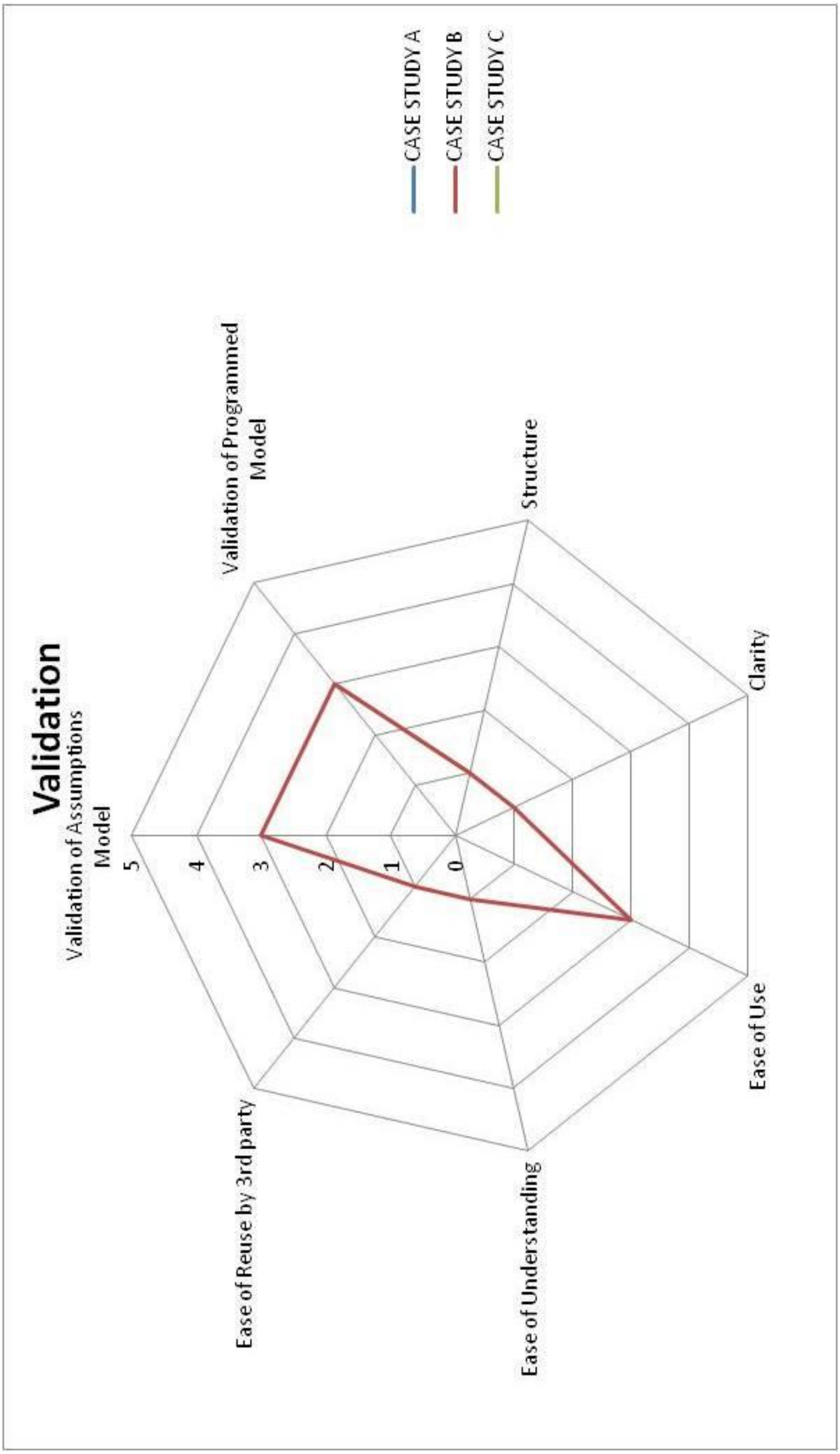


Figure 5-5: Validation Assessment Comparison

5.2.5 Validation Comparison

Figure 5-5 depicts the scoring given to validation from chapter 3, however Case Study A and C contained no documentation for validation. Validation of any simulation model is paramount. Without a valid model the results and other outputs are next to meaningless for the simulation study. Sargent [27] says that good documentation is necessary and leads to easier validation and verification of a model but all the case studies suffered when it came to documenting this area.

Case study A contains absolutely no references to how the assumptions model and the programmed model were to be validated. The simulation analyst knew exactly how the validation process was going to happen yet had not documented the process that was to be undertaken for the validation process.

Case Study C also contained no documentation for this phase. Even though this case study hadn't been fully commissioned, stating how the assumptions model or the programmed model would be validated could have easily been documented.

Case Study B included information on the validation process of the assumptions model and the programmed model but only briefly recorded it. The overall assessment of validation for the three cases backs up the idea that many areas of a simulation study are overlooked prior to when those particular areas are needed. The simulation modeller of Case Study A doesn't need to prove the simulation model is valid until they are presenting the results to the client. Only then must the information be presented to prove the entire study's credibility. This is by no means a proper way to conduct a simulation. The same can be said for Case Studies B and C. The important step of validation should not be ignored by the modeller or team and only revisited late in the study treating it as a postscript.

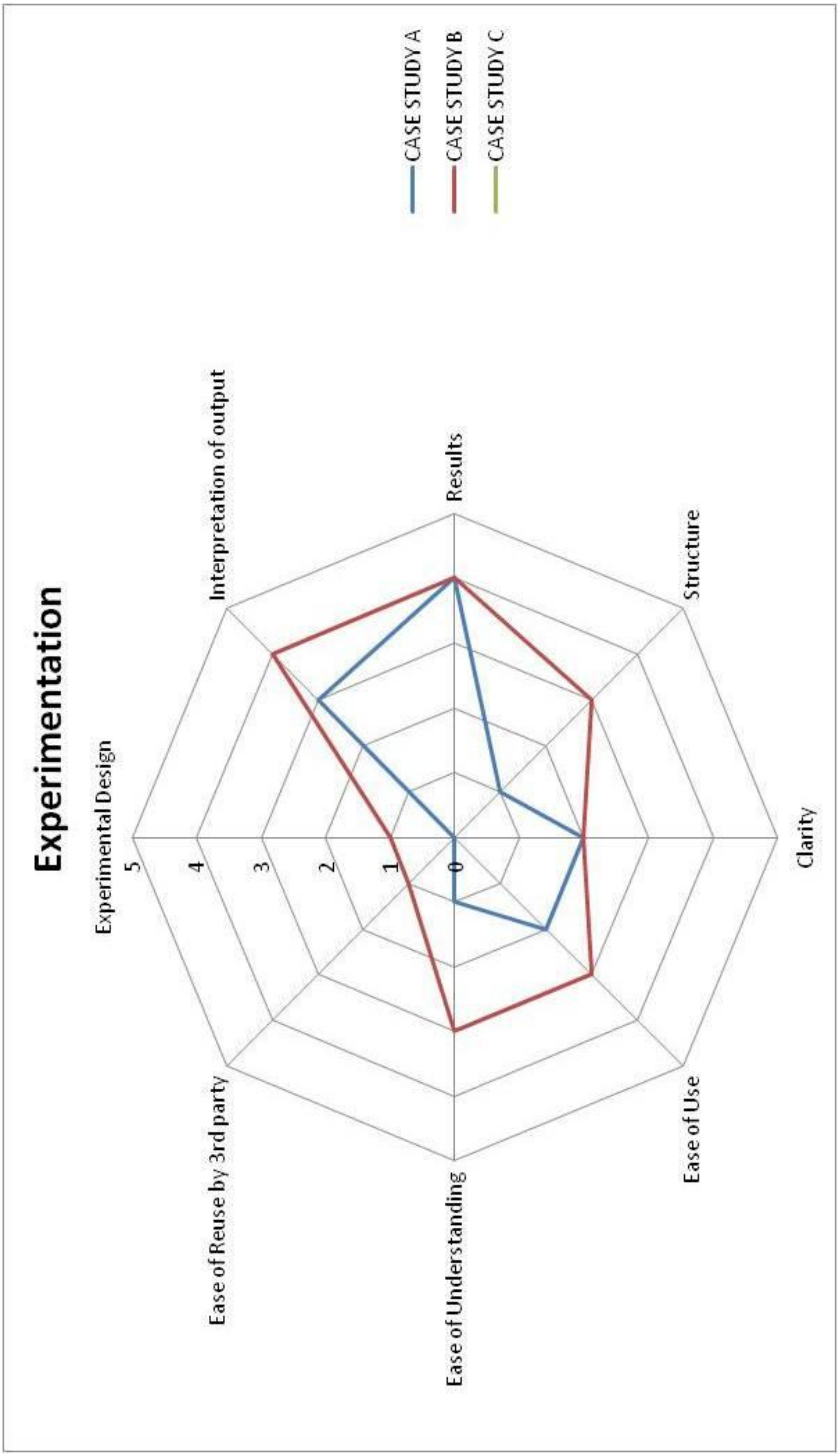


Figure 5-6: Experimentation Assessment Comparison

5.2.6 Experimentation Assessment Comparison

The assessment of experimentation incorporates the experimental design as well as the interpretation of output and the results. Refer to Figure 5-6 for the direct comparison of the case studies to one another.

Perera et al [31] highlight how bad documentation can scupper a simulation study and if it's poorly planned how it can lead to lengthy experiments in turn costing time and money. Case Study C contains no information relating to this area and was also nonexistent in Case Study A. The experimental design wasn't covered adequately in Case Study B either. Case Study A's model was designed with the plan to have workers technicians and engineers conduct their own experiments but no examples or bench marks were included within its documentation..

Both Case Studies A and B documented the results of their studies admirably but it was Case Study B that documented the interpretation of the models output to a greater degree. As it stood after the assessment Case Study B was in a much stronger position to present their results to their client. Again it appeared that the simulation modeller in Case Study A would only document the needed areas when it was required i.e. when they needed to present the simulation to the client.

The overall documentation for case study A scored poorly. The recorded information wasn't clear and lacked structure which affected the other assessed areas. This area needed to be revisited by the simulation modeller before any formal presentation could be made to the involved client.

Case Study B's approach to the Experimentation Documentation was better if however unclear. It did apply an acceptable structure which helped towards a better scoring for ease of use and ease of understanding. Even though the documentation would have to be reassessed before a client presentation the simulation modeller is in a better position than their peer in Case Study A.

5.2.7 Overall Assessment - Conclusion

Importantly none of the people involved knew that their studies would be used as a case study so all documentation was created naturally covering the aspects of the study that the simulation team / modeller believed they needed to cover. If the involved parties were aware of the headings and subheadings that their documentation would be assessed under they may have concentrated on generating documentation for these areas. For example if they knew each section would be scored on how well it was structured they may have rearranged the documentation to follow a stricter pattern and so on. This gave a better insight into how simulation studies are created.

Each case study differed when considering the documentation that each modeller or team applied to their study. From the assessments good structure and clarity leads to better ease of use and in turn opens the door to the possibility of reuse. For the most part the involved parties were not overly concerned with the documentation, but rather more concerned with creating a working model that was going to output good results for their clients. The documentation was just a by product of the simulation study rather than any specific concentration being applied to it. This can be a common trap to fall into. Inherently good documentation will assist the simulation study and in turn lead the better results being outputted for the client involved. The documentation from any study should act as an aid to the entire simulation project. Users and involved personnel shouldn't get bogged down in large amounts of notes and files. In some instances clarity was deficient in Case Study B as is was written as a report style document with large volumes of text. Steering away from large amounts of text and representing the information through graphics could help increase the clarity of contained information.

The simplicity of the documentation in case study C allows the involved people to use it easily. The less complex the documentation the easier it is, to both use and implement. Packing the documentation with too much detailed information would have the opposite effect. A balance must be found between what the documentation includes and how easy it can be used. Again this is a positive argument for the implementation of easy to read graphics as a way of representing relevant information to aid simulation studies.

Over Case Study A, B and C, Clarity was a common attribute that scored lowly. None of the people involved with these simulation studies specifically set out a methodology that their documentation was to follow. This contributed greatly to the downfall in this area.

All the case studies defined their respective problems to different extents. The grades of aim/objectives, deliverables/timeline and model scope also differed across the three cases. When the case studies were interrogated the general consensus was that anybody involved within these particular case studies knew what their goals were and what they wanted to achieve with the simulation study. Nevertheless these articles were not recorded suitably within the respective documentation for each. A person not directly involved in the project could not tell what the simulation aimed to accomplish by the documentation presented. This issue wouldn’t be a particular problem if clients were closely involved with the goings on of the simulation study that they had commissioned; everyone intimately involved would know exactly what outcome the study was aiming towards.

Aims should be decided upon at the outset of the study. The benefit of having the aforementioned aspects well documented would lead to simulation analysts having a concrete document to refer to if the simulation study evolved or if more was demanded of them than was needed other than what was agreed early in the life cycle of the simulation study. If a project was relatively long or indeed if an analyst was revisiting an old project having good documentation in place would help them understand and keep track of exactly what they set out to achieve.

One common theme across this section was that the area of problem formulation was generally well documented and easy to understand. None of the aim/objectives, deliverables/timeline and model scope required an in depth knowledge or any special training to understand. The problem definition sections were treated simply and as a result the involved parties could use and understand the information easily.

The relevance of the data and information varied over each case study. Case Study A contained a huge amount of information but much of which wasn’t directly useful to the simulation study. Case study C on the other hand only contained data/information which was relevant to the simulation study. With the intention of having a short concise document the analyst ended up recording data/information that was relevant to the study. That said Case Study C may not have had enough information/data to construct a valid credible model. The people involved have to strike a balance between the amount of information/data they want to record and how detailed they want their documentation to be.

Case Studies A and B contained detailed designs of each of their respective models. Most simulation studies concentrate on the model rather than the study and the scoring here backs up this. Where the design of the model was well documented in Case Studies A and B, the simulation model of Case Study C was still to be built so no documentation existed on the model as of yet. The approach of Case Studies A and B follow suit with an attitude of building the simulation model first and revisiting documentation afterwards.

None of the documentation assessed had been developed with the idea of reuse in mind, therefore explaining the low scoring of this across each. If one were to implement a structured specific way of documenting a simulation study it would be possible to make it easier to reuse the contained information.

According to the available literature there are certain steps which must be undertaken in a simulation study for that study to be completed (Law [24]). The gaps (in some cases of whole sections) in the presented case studies show that not all of this information or activities are being documented. The personnel involved know what is needed to validate their models. This information needs to be addressed at some stage of the study as a model with no validation, whether the personnel know what was or has to be done, is in no way credible. Having a credible model is paramount for the study to be a success. In the case of the third case study, C, the whole project had not been commissioned yet. The reasoning behind no documentation existing on validation was that the actual model hadn't yet been created. While this is a legitimate point, documentation could have been easily generated regarding what should be done for the validation steps in this case. This documentation could be used as a template for the actual validation process and whatever changes were needed could be made simply to already existing documentation.

Across all stages of the assessment there were areas over looked by the people generating the documentation for the respective studies. The simulation studies have to be presented to a client on completion. Applying good documentation practice throughout a simulation study can only save a user time when it comes to the end of the study life cycle. Documentation is pivotal in a simulation study. At the end of a simulation project the final stage is to be present the simulation study [12][24][25]. The case studies highlighted the areas that the SysML diagrams were developed to help aid and assist, with the aim of having a concise document so the simulation study could be presented.

5.3 SysML Diagrams

Davis et al [35] believe that a major problem is starting to build a model with no clear purpose. Starting to build a simulation model using SysML as a graphical aid puts a definite structure on the study. This allows the simulation modeller or involved party to address the simulation study step by step. This is seen in the steps taken to build the SysML diagrams designed for this thesis.

The systems modelling language has the capacity for creating a huge number of diagrams that could all in their own ways be used in conjunction with a simulation study. The SysML specification has been created with each included diagram having a specific purpose in describing a (or part of a) system. This thesis sees some of these diagrams applied to different areas within a simulation study.

Rubinstein [6] notes that good modelling must be a trade off between realism and simplicity. The same can be said for the creation of SysML diagrams. When approaching the diagrams it was decided that a balance was needed between ease of generation and how much detail the diagrams should include. By this approach the diagrams could be considered useful aids in the simulation study and with such a balance could be of further benefit when the diagram types would be applied to additional simulation studies. It was important to find this balance between how much time was invested into the SysML diagrams and how complex the diagrams needed to be, because a simulation modeller spending too much time on creating SysML diagrams would draw away from them concentrating on the simulation. The SysML diagrams are suppose to be a graphical aid making the overall simulation study easier not to create a hindrance for the simulation modeller. Each diagram's role will be further explained within this section.

Formulating the Problem

One of the major problems Davis et al [35] mention is the translation of requirements for simulation into components applicable to the job. Researchers find this an extremely difficult problem to deal with and generally ignored it. Davis et al believe a heavier emphasis should be put on it. The requirements diagram of SysML gives the user a tool to easily capture the requirements of a simulation study and document them for the duration of the project.

Requirement Diagrams were specially developed for the Systems Modelling Language. As they can represent numerous necessities that need fulfilling within a system it was chosen to represent those which must be completed within a simulation study, i.e. all the requirements generated when formulating the problem at the outset of the study. The SysML requirements diagrams are relatively intuitive in relation to the majority of SysML. This is a huge positive, as whoever is using the SysML requirements diagram won't need a great knowledge of SysML or particular training to understand the diagram. The hierarchical nature of the requirements diagram allows the user to see the higher level requirements i.e. the more important requirements that have to be fulfilled. As the lower level requirements are also clear the user knows what has to happen prior for those higher level requirements to be fulfilled. The diagram shows the interrelation between several of the requirements.

These connections highlight several of the requirements that have been derived as a result of others. Similarly the satisfy connections show what (or who) has to be present for the requirements to be fulfilled. In this case it is “who” that has to be present in the form of the actors visible on the requirements diagram. These four actors are tied to the highest level requirement by the satisfy connection. Due to the hierarchical nature of this requirements diagram the satisfy connection also translates down through the diagram to the lower level requirements. It is very possible in another study that a person (or actor) is only needed to satisfy a certain requirement.

The importance of the diagram created for this step in the simulation study is that it is easily created, understood and could easily be reapplied throughout any simulation study.

Data/Information

The problem when approaching a diagram using SysML to aid the Data/Information stage of a study is that the information or data can come in numerous forms or formats. As this was the case in the example used for this project it wasn’t feasible to approach the data/information saying a certain number of things need to be recorded for a successful simulation study as studies differ to greatly from one case to the next. The approach taken was to use the SysML Package diagram as a means of collating and organising the data and information. Through this method the user can see what information is related without having sprawling information recorded in many different places. This was noticed in the case studies of chapter 3 so again the aim was to produce a simple intuitive diagram through SysML that would aid a user in their simulation study. Having all the information documented through the same method would be beneficial when re-approaching the data/information, something that would have to be done through-out the study anyway. The user could collate the information through the package diagram for ease of presentation to the client at the culmination of their simulation study.

Assumptions Model

An assumptions document is essential in a simulation study and the information contained shouldn’t be difficult to understand [12].The assumptions model is an important part in a simulation study and is paramount to the creation of the simulation model itself. Again applying the requirements diagram to this stage of the simulation study highlighted what assumptions were made about the simulation in such a way as to easily convey the information to the reader/user. The development of the diagram was similar to the diagram to aid “Formulating the Problem”. In SysML the requirements diagrams can be represented in a tabular format. This is an especially useful way of viewing what the specific assumptions are when the simulation modeller is building the model.

Having an overall view of how the simulation of the real world system is going to work gives the simulation analyst/team a view on how to approach the problem they are addressing. The overall view was recorded using the SysML Activity Diagram. The diagram itself resembles a generic flow chart but has subtle differences. Like a flow chart it doesn’t take any specific training to understand. Actions and objects are represented by elements with rounded corners and elements with square corners respectively. Each action has to be completed in conjunction with each object or objects for the simulation to move from the initial input to the output.

In relation to the actual simulation model itself the Assumptions Model can be represented through the SysML Block Diagram. The idea behind the implementation of this diagram was to break down the different sections needed in the simulation model for the user and also highlight the sections that had to be addressed when creating the actual simulation model.

Together the three diagrams formed a strong grounding to aid the Assumptions Model phase of a simulation study. Without successful and strong documentation at this point in a study the model may suffer later with much time lost and in turn money being wasted.

Validation of the Assumptions Model

Working hard on the Assumptions Model and implementing it without any validation leads to the simulation model lacking any credibility. The validation at this point is a monumentally important step. However validation can differ from study to study and how a simulation team validate their Assumptions Model is up to them. Rather than develop a diagram on how to validate the assumptions model the approach was taken to develop a diagram on how the assumptions model was validated. This way the same method could be applied to other simulation studies that may or may not be using a different validation process. Using the SysML Activity Diagram the validation process of this study was documented using a number of actions and objects. The same actors found in the previously mentioned requirement diagrams are represented as objects here. This is just one example that can happen within SysML where each diagram is linked to another through the contained elements. There is also an interaction of objects that have been developed previously i.e. the "Assumptions Model". The diamond element shows where a decision is made within the activity diagram, which is whether or not the Assumptions Model has been validated. The loop created here shows that when validation doesn't occur the validation process begins anew until all the criteria have been met for said validation to occur, a loop that should be definitely represented in any simulation study.

Program the Model

Simulation models can be created in a number of different ways. It was decided that the simulation model should be documented using the SysML block diagram and internal block diagram as these diagrams and the method employed could be used elsewhere to represent other simulation studies. Not all simulation studies are conducted using the same programs or methods. By using blocks parts or pieces of a graphical simulation program could be as easily documented as a collection of code. The Block Diagram was used as an overall view of the simulation model based upon the block diagram that represents the assumptions model, with extra detail representing the flow between those blocks. The internal block diagrams then represent the inner workings of these blocks and can be drilled down by the modeller to whatever detail they find appropriate for documenting their simulation model.

In this particular case the simulation model was created using a graphical simulation tool. The translation of this to the internal block diagrams is that every block part in the internal block diagram represents a piece/part that was used in the simulation tool. Delving down to such a level of detail was not productive for this simulation as recreating the model using the internal block diagrams and then building the model again through the graphical simulation tool just adds an extra step. This was done during this section to show that it can be done and could serve better when other simulation styles are used such as representing lines of code. Documenting the code using Block and Internal block diagrams would make it easier for the modeller to visualise the simulation model and present it to any other people involved in the study. The internal block diagrams show that this methodology can be applied to where such a break down would help explain what is happening within the simulation model. The block diagram on the other hand is extremely useful when a simulation becomes large and complex, breaking it up into different sections can aid the modeller in presenting it to a client. In each block diagram one can see flow ports on each "block". Especially in instances where more than one flow exited/entered a block the flow ports dictated which blocks could connect to one another so no mismatches could occur. The specification doesn't allow for such mismatching to occur and using a program specifically designed for SysML would also help with this.

Validating the Simulation model

The approach to constructing this diagram was very similar to the aid created for the Assumptions Model Validation. The SysML Activity diagram represents how the validation process occurs rather than depicting a method to validate all simulation studies. The application of Activity Diagram is based around a number of objects and activities that define the validation process that was implemented. There is also a loop present showing that the validation process of this simulation study isn't a linear as several of the activities may have to be completed more than once. To produce a SysML Activity diagram to represent the validation process for a simulation model is a relatively easy task. The diagram was arranged so any other users will also find it easy to comprehend. Through SysML the activity diagram used objects generated previously in other diagrams again showing the interrelationship between all of SysML.

Experimentation

In this particular case, because of the nature of the simulation study, the SysML Use Case diagram was used solely as an aid showing how any of the potential users could interact with the model. The simulation study was commissioned with the aim of building a reconfigurable model so that engineers/technicians may experiment on the model themselves. The use case is important as it shows the engineers/technicians what interaction they have with the simulation model and what they can reconfigure. In other studies this could be easily applied using the same methodology whereby different involved personnel each have a different level of interaction.

In other studies where there are a defined set of experiments it may be prudent to use a requirements diagram in conjunction with a use case diagram to spell out exactly what experiments and parameters need to be set for the simulation.

All diagrams show something that needs to be presented to client in the final step of the study. Having good visual documentation to aid throughout the study can aid you right up until the simulation and results are presented to a client.

Simulation study Structure and SysML

Having a proper structure in place is important to a simulation study. By using Law's [24] 7 steps to creating a successful simulation study as a guideline it is possible to apply SysML as a tool to maintain a good structure throughout the study. Without a good structure and without proper documentation in place, the study can evolve unnecessarily [36]. As was seen in section 2.9.1 the reality is that the simulation modeller will often begin building the simulation model only to have to revisit and alter it to suit the needs of the client [45]. By using a definite structure aided by SysML a modeller can reduce the need for the aforementioned. Even when there is a need to readdress the model the involved people will have good structured documentation in place to reduce the time spent. Diagrams created using the system modelling language can act as a tool for easily communicating information. This can be seen in the independent assessment discussed in section 5.5.

5.4 Reusability

Reusability of simulation models is an area of simulation that sustains a lot of interest and research. Simulation models are more often than not built for a specific purpose. Where that specific purpose may be similar the model or particular model elements could be reused or rebuilt in the new simulation model being constructed. Whether it would be similar or not a project manager could easily make the decision on the model's validity for the study in question based on the information that would be available from the previous team/developers, if good documentation practice had been previously used. This could lead to a serious reduction of the time to complete a simulation study. However such translation could only be applied if the programming of the simulation model has indeed been described and documented extremely well. Without the simulation model's construction being described extremely well only the original modeller/developer would understand it to a degree where it would be implemented quickly and easily whereby time would be saved. There are obstacles that are innately connected with reusability. There is little motivation for developers to implement simulation processes to enable model reuse. By doing so they would be increasing model development cost while the benefit would be gained by the following developers using their work [39]. However if the simulation industry wanted to tend towards more reusability good documentation would be the place to start. By breaking the documentation down into sections (steps to complete a simulation study) it makes it easier for an analyst or simulation team to reuse each of these sections either separately or collectively.

Robinson et al. [39] also notes that reusability can raise issues with the confidence of a reuse model or component. They also point out that the time and cost that would be incurred to familiarise one's self with another's model may outweigh the time and cost benefits of reuse. This reasoning would highlight the importance to implement good documentation to help allay some of these obstacles.

Also there is a good case for the reuse of good documentation for the experimental planning stage of the simulation study, again depending on the applicability to another project. Regarding reusability, having high-quality documentation on hand could allow a modeller or project lead to discern whether or not the same "experiments" can be applied from one study to another. Even in the case where specific experiments are not useful, the methodology or approach can be mimicked potentially saving more time for the overall simulation study.

5.5 Independent Assessment

It was important to gain an evaluation of the generated diagrams from independent sources of how SysML faired as a graphical aid and what benefit it gave to the simulation study in question. As mentioned in section 4.2 it was only a small selection of technical people that were surveyed. Instead of giving a definitive overview of what the simulation industry thinks of SysML the results of the independent assessment can only point towards how technical minded people perceived the diagrams. Oscarsson and Moris cited Lehman (1977) [21] and discussed the three audiences that simulation documentation should be aimed towards as mentioned in section 2.5. These are (a) The Programmer or original simulation developer (b) The new engineer/simulation developer who going to either study the model or use it (c) Others who want to understand the main frame but not necessarily all details in the model. During the independent assessment it was vital that these be the type of people chosen to help assess. A much wider selection of engineers and analysts from the simulation industry would need to be surveyed to get a more definite overview of how effective SysML is in this instance. Unfortunately it wasn't possible to survey a wide selection of people in the course of this thesis, however from this assessment the best possible information was extracted from the assessors. It is important to note that the results gained from these assessments give some insight into how technical people absorbed the SysML diagrams. For example each was able to understand the diagrams separately with having little or no knowledge of SysML or the contained information previously. This shows the potential SysML has as a communication tool to relay good documentation practice between related parties of a simulation study.

The mean result of the scoring from the independent assessor can be seen below in Table 5-1. Please refer to Figure 4-17 for the individual scoring from each assessor. These results can be found in their entirety in Appendix A.

Table 5-1: Mean Scoring of Independent Assessment

	Assessed Area	Mean
(i)	Communicative Ability	4.2
(ii)	Structure	4.2
(iii)	Clarity	3.8
(iv)	Ease of Use	4
(v)	Ease of Understanding	3.2
(vi)	Ease of reuse by 3rd party	3.6

All aspects of the SysML documentation when averaged scored above “fair” using the scoring seen on Table 3-1 previously. What is evident from the scores is that all the assessors found that the documentation presented and communicated the information to the reader/user well.

“Structure” scored on average highly as it was seen that the structure of the SysML documentation was well arranged and could be manipulated to suit different simulation studies and needs.

The area of Clarity scored just below “good” but was still overall positive in the provided comments from the assessors as they found the information readily available in SysML diagrams and that information could easily be gleaned to help aid the overall study for a simulation modeller or team working on the study.

On average Ease of Use scored 4 (good), again showing how the assessors found the documentation to be easily be used by the people/person involved.

Both “Ease of Understanding” and “Ease of Reuse” scored slightly lower on average. The assessors had mixed views on whether the information was easily understood or whether the SysML documentation could be reused by a third party elsewhere. This stemmed from the different technical background the assessors came from.

Assessor 1 thought that the manner in which SysML presented the information was a great way to convey the information of the project. Assessor 2 echoed this sentiment but both assessor 1 and 2 thought that the diagrams needed some form of introduction to explain the scenario and overall aim. This was purposefully not done to see what overall information the assessors could take from the diagrams. For the most part the assessors believed those of a background in the simulation industry would easily understand the information while it was also noted that a person with little or no simulation experience would not gain the full understanding from the SysML documentation. Similarly the consensus was that with no simulation knowledge it would be next to impossible to reuse these particular diagrams, however even a small amount of knowledge of simulation would go a long way in their reuse.

Assessor 3 highlights what is an important point with regards to the structure of the documentation. Once the SysML diagrams are generated it is easy to alter them without any major rewrite of the documentation. This would be particular useful when considering the requirement diagrams used for documenting “Formulating the

Problem" (section 2.6.1) and the assumption decided upon prior to the building of the simulation model. Both these areas would be subject to much discussion before setting them in stone. The SysML diagrams allow the documentation to evolve with the study if/when it happens.

Assessor 4 praised the idea of standardising the implementation of steps in a simulation study. They also highlight how the documentation would be an excellent tool for communicating pitfalls, issues and steps encountered to another user to put them in a better position for reusing/conducting a simulation study.

As a person with no direct simulation or manufacturing knowledge assessor 5 made some interesting points. They noted that the full use of the Systems Modelling Language could not be implemented without having an in depth knowledge of the SysML syntax and structure. This was echoed when the SysML diagrams were created as an aid for the simulation study. The diagrams were chosen and designed in such a way so that needing special knowledge or training of SysML would be minimised. SysML requires a certain amount of training and knowledge for one to fully understand it. For the purposes of presenting to a client or in this case independent assessors (and in future cases), a legend of the important elements and connections that are found in the SysML documentation for simulation could greatly increase the understanding of the information that in contained within the SysML graphical aids.

Oscarsson and Moris [21] comment that a simulation model must be understood by others. Through the independent assessment it was possible to see that SysML could be used as a graphical aid so that the information contained within a simulation study could be communicated to technically educated people. Further study on this topic may include reaching out to a wider audience in the simulation industry to obtain their critical analysis of how effective SysML is at this now that it has been seen that it is possible on a small scale.

6 Conclusion

6.1 Introduction to Conclusion

Documentation has a major role to play in the simulation industry and it is the author of this work's belief that good documentation should play a pivotal piece in each and every simulation study. Following are some of the key conclusions from this work.

Documentation Neglected

In general when a simulation study is conducted documentation is so often an overlooked area, yet it should be pivotally important in forming a successful study. From what is seen in the case studies presented, documentation exists. It is just not completed to a degree whereby it can aid a study in saving time. Mostly documentation isn't complete to cover all the aspects a simulation study needs for it to be deemed a successful simulation study.

Good Documentation as a Time/Money Saver

Good Documentation practice, if implemented correctly can be a valuable tool to an entire simulation study from explaining what the study is and how the simulation is working to communicating much needed information between members of a simulation team or between the analyst and client. Using a standard for documentation can make this all the simpler as that documentation will have a definite structure. In addition using a specific graphical standard like SysML can make it easier to visualize the information. Having the documentation presented in such a way allows for it to be more easily used reused and explained. From that if one was to use a software package specifically designed for the implemented standard, e.g. Artisan Studio Uno, it allows the user to concentrate on the documentation more and not get bogged down by large volume of information contained within the standard itself.

Documentation can aid Reusability in Simulation

Reusability is a large area of research in the simulation community. Documentation for reuse in the simulation community is far less so. Reusing documentation from another study or a previous study of your own has the potential to save you a great deal of time. However the original documentation must be designed with this purpose in mind from the outset or one could spend longer than needed reusing said documentation. A specific standard of documentation using a purpose built software package will drive the documentation towards a point where reuse is a lot easier than if it had been recorded without. Training at the initial stages of such a thought process is vital especially with respect to (in this instance) SysML and Artisan Studio Uno. These may take some time to learn but as simulation studies often have long project life cycles it would pay off in the long term and wouldn't have to be relearned with each new project.

Final Thoughts

The thesis raises some interesting issues regarding documentation in the simulation industry but there is still potential for much more research to be conducted in this area. Some potential for further study and research is briefly discussed highlighting a direction future work could take.

To really assess the impact good documentation practice could have on a simulation study it would need to be implemented from the very beginning of the simulation study, right through until the end by the people involved. Ideally further work would include implementing it under several different constraints. Firstly implementing it in a study whereby the work is conducted by a single person and then continuing on to apply good documentation practice using SysML to a project carried out by a team. A slightly different avenue that should be taken regarding this work would be using good documentation to teach simulation with the aim of training a user to be proficient in execution of the simulation model.

As noted in this thesis the created diagrams were only assessed by a relatively small number of people. A further study could be made whereby a much larger sample of manufacturing engineers, simulation analysts and technical people are surveyed to gain a much better understanding of how effective SysML is as a graphical aid for simulation studies. This thesis has shown that SysML diagrams can be created to capture and relay information from the beginning of a simulation study right through until the final stages.

Documentation is a large area in simulation and there is huge scope for expanding on this to aid the simulation community and drive simulation studies forward.

7 References

[1] Groover, M.P., (2007), *Fundamentals of Modern Manufacturing,* (Third Edition ed.), John Wiley & Sons Inc, United States of America.

[2] Kelly, K. (2005), "How Technology Evolves", *Feb 2005,www.ted.com/talks/lang/en/kevin_kelly_on_how_technology_evolves.html* .

[3] Banks, J., (1998), *Handbook of Simulation*, John Wiley & Sons, Inc.

[4] Friedenthal, S., Moore, A. and Steiner, R., (2008), *A Practical Guide to SysML - The Systems Modeling Language*, MK OMG.

[5] Highland, H.J. (1977), "A Taxonomy Approach to Simulation Model Documentation", *Proceedings of the 1977 Winter Simulation Conference,* pp. 724-729.

[6] Rubinstein, R.Y. and Melamed, M., (1998), *Modern Simulation and Modeling,* John Wiley & Sons, Inc.

[7] Neelamkavil, F., (1987), *Computer Simulation and Modelling*, John Wiley & Sons Ltd.

[8] Maria, A. (1997), "Introduction to Modeling and Simulation", *Proceeding of the 1997 Winter Simulation Conference,* pp. 7-13.

[9] Shannon, R.E. (1998), "Introduction to the Art and Science of Simulation", *Proceeding of the 1998 Winter Simulation Conference,* pp. 7-14.

[10] Elizandro, D. and Taha, H., (2008), *Simulation of Industrial Systems*, Auerbach.

[11] Ingalls, R. G.(2008), "Introduction to Simulation", Proceeding of the 2008 Winter Simulation Conference *7-10 December,* pp. 17-26.

[12] Carson II, J. S. (2005), "Introduction to Modelling and Simulation", Proceedings of the 2005 Winter Simulation Conference *4-7 December,* pp. 16-24.

[13] Song, L., Ramos, F. and Arnold, K., (2008), "A Framework for Real-Time Simulation of Heavy Construction Operations", *Proceeding of the 2008 Winter Simulation Conference,* pp. 2387-2395.

[14] Wang, J., et al., (2006), "Simulation of Transfer Organization of Urban Public Transportation Hubs", *J Transpn Sys Eng & IT,* Vol.6 (6), pp. 96-102.

[15] De Angelis, V., Felici, G. and Impelluso, P., (2003), "Integrating Simulation and Optimization in Health Care Centre Management", *European Journal of Operational Research,* Vol.150 pp. 101-114.

[16] Miwa, K. and Takakuwa, S., (2008), "Simulation Modeling and Analysis for In-Store Merchandizing of Retail Stores with Enhanced Information Technology ", *Proceeding of the 2008 Winter Simulation Conference,* pp. 1702-1710.

[17] Bernard, C. and Lemieux, C., (2008), "Fast Simulation of Equity Linked Life Insurance Contracts with Surrender Option", *Proceeding of the 2008 Winter Simulation Conference,* pp. 444-452.

[18] Balci, O. 2003, "Verification, Validation, and Certification of Modelling and Simulation Applications", Proceedings of the 2003 Winter Simulation Conference *7-10 December,* pp. 150-158.

[19] Silbey, V. (1979), "Management Orientated Documentation of Simulation", *Proceedings of the 1979 Winter Simulation Conference,* pp. 29-34.

[20] Triebig, C. and Klugl, F., (2009), "Elements of A Documentation Framework for Agent-Based Simulation Models", *Cybernetics and Systems,* Vol.40 (5), pp. 441-474.

[21] Oscarsson, J. and Moris, M.U., (2002), "Documentation of Discrete Event Simulation Models for Manufacturing System Life Cycle Simulation", *Proceedings of the 2002 Winter Simulation Conference,* pp. 1073-1078.

[22] Gass, S.I. (1984), "Documenting a Computer Based Model", *Interfaces,* Vol.14 (3), pp. 84-93.

[23] Ryan, J. and Heavey, C., (2006), "Process Modelling for Simulation", *Computers in Industry,* Vol.57 pp. 437-450.

[24] Law, A. M. 2008, "How to Build Valid and Credible Simulation Models", Proceedings of the 2008 Winter Simulation Conference *7-10 December,* 39-47.

[25] Banks, J. 2000, "Introduction to Simulation", Proceedings of the 2000 Winter Simulation Conference *10-13 December,* pp. 9-16.

[26] Clema, J.K. (1978), "General Purpose Tools for System Simulation", *Proceedings of the 11th Annual Symposium on Simulation,* pp. 37-60.

[27] Sargent, R. G. 2008, "Verification and Validation of Simulation Models", Proceedings of the 2008 Winter Simulation Conference *7-10 December,* pp. 157-169.

[28] McNally, P. and Heavey, C., (2004), "Developing Simulation as a Desktop Resource", *International Journal of Integrated Manufacturing,* Vol.17 (5), pp. 435 - 450.

[29] Sadowski, D. A. 2007, "Tips for Successful Practice of Simulation", Proceedings of the 2007 Winter Simulation Conference *9-12 December,* pp. 107 - 112.

[30] Arthur, J. D., et al., 1999, "Verification and Validation: What Impact should Project Size and Complexity Have on Attendant V&V Activities and Supporting Infrastructure?", Proceedings of the 1999 Winter Simulation Conference *5-8 December,* pp. 148-155.

[31] Perera, T. and Liyanage, K., (2000), "Methodology For Rapid Identification and Collections of Input Data in the Simulation of Manufacturing Systems", *Simulation Practice and Theory,* Vol.7 pp. 645-656.

[32] Carson II, J. S. (2002), "Model Verification and Validation", Proceedings of the 2002 Winter Simulation Conference *8-11 December,* pp. 52-58.

[33] Stahl, I. (2000), "How Should We Teach Simulation?", Proceedings of the 2000 Winter Simulation Conference *10-13 December,* pp. 1602-1612.

[34] Banks, J., Carson, J.S. and Nelson, B.L., (1996)*, Discrete-Event System Simulation,* (2nd ed. ed.), Prentice-Hall.

[35] Davis, P. C., et al., (2000), "Model Composability as a Research Investment: Responses to the Featured Paper", Proceedings of the 2000 Winter Simulation Conference pp. 1585-1591.

[36] Campbell, R. and P. Young (2009), "Manufacturing System Description for Simulation Modelling - A Review of Current Practices and Tools", Proceedings of the 26th International Manufacturing Conference TCD, *09 Sept, pp* 313-320.

[37] Kettinger, W.J., Teng, J.T.C. and Guha, S., (1997), "Business Process Change: A Study of Methodologies, Techniques and Tools", *MIS Quarterly,* Vol.21 (1), pp. 55-80.

[38] Ang, C.L., Luo, M. and Gay, R.K.L., (1995), "Knowledge-based approach to generation of IDEF0 models", *Computer Integrated Manufacturing Systems Volume 8 Number 4,* pp. 279-290.

[39] Robinson, S., et al., (2004), "Simulation model reuse: definitions, benefits and obstacles", *Simulation Modelling Practice and Theory,* Vol.12 (7-8), pp. 479-494.

[40] Reese, R. and D. L. Wyatt (1987), "Software Reuse and Simulation", *Proceedings of the 19th conference on Winter Simulation,* Proceedings of the 19th conference on Winter Simulation, pp. 185-192.

[41] Pidd, M. 2002, "Simulation Software and Model Reuse: A polemic", Proceedings of the 2002 Winter Simulation Conference 772-775.

[42] Carnahan, J. C., Jn P. F. Reynolds and D. C. Brogan (2005), "Simulation-Specific Characteristics and Software Reuse", Proceedings of the 2005 Winter Simulation Conference, pp. 2492-2499.

[43] Kasputis, S. and H. C. Ng (2000), "Composable Simulations", *Proceedings of the 2000 Winter Simulation Conference,* Proceedings of the 2000 Winter Simulation Conference, pp. 1577-1584.

[44] Banks, J. (2000), "Simulation in the Future", Proceedings of the 2000 Winter Simulation Conference pp. 1568-1576.

[45] Stewart, D. B. (2001), "Twenty-Five Most common mistakes with Real-Time Software Development", ESC SF *April*.

[46] SysMl - Open Source Specification Project, [online], http://www.sysml.org/.

[47] Buede, D.M., (2009)*, The Engineering Design of Systems,* (2nd ed.), John Wiley & Sons, Inc, Hoboken, New Jersey.

[48] Mar, B. W. (1997), "Back to Basics again - A Scientific Definition of Systems Engineering", Proceedings of the 7th Annual International Symposium of the International Council on Systems Engineering.

[49] Herzog, E. and A. Pandikow (2005), "SysML - An Assessment", Proceedings of the 15th INCOSE International Symposium.

Appendix A

Appendix A contains the document circulated to the independent assessors and their scoring and comments as discussed in Chapter 5 as they were communicated.

Circulated Document for Independent Assessors

All,

Please find in the other attached document 13 separate SYSML diagrams. The diagrams are

"REQ" – requirements diagrams

"Bbd" – Block Diagrams

"Ibd" – Internal Block Diagrams

"Act" – Activity Diagrams

"Pkg" – Package Diagrams

"UCD" – Use Case Diagram

The diagrams were formed as a means of aiding the documentation of a Simulation Study.

The diagrams were also created loosely around the 7 steps needed to complete a successful simulation study.

1. Formulate the Problem
2. Collect Information and Form an Assumptions Model
3. Validation of the Assumptions Model
4. Program the Model
5. Validation of the Model
6. Design and conduct experimentation
7. Document and present the Study

Rating	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Very Poor
0	Non Existent

The aim here is to examine the documentation under the headings outlined below. Using the table above please score each from 0 – 5.

Communicative ability:

How well the documentation communicates the information to the reader/user. E.g. is this format better than presenting all the documentation as a written paragraph etc.

Structure:

The arrangement and organization of the findings in the documentation is considered here. Score on whether the structure is rigid or flexible, is adaptable, defines what information is documented or has the capacity to expand. Also score on whether the structure is consistent throughout the entire section is also scrutinized.

Clarity:

The clarity of the documentation deals with how readily available the documented material is and how clear it is to the person viewing it.

Ease of Use:

Ease of use is how easily the people/person involved could use and apply the documentation in their simulation study.

Ease of Understanding:

How comprehensible the documentation is when being examined by an internal or external party. The ease of understanding assessment examines if the documented knowledge is readily understandable or whether specialist training or specialist skills are needed.

Ease of Reuse by Third Party:

Can the documentation can be easily reused by a third party (not already knowledgeable of the project). Can some of it be reused and other parts not?

Scoring

Communicative Ability _______

Structure _______

Clarity _______

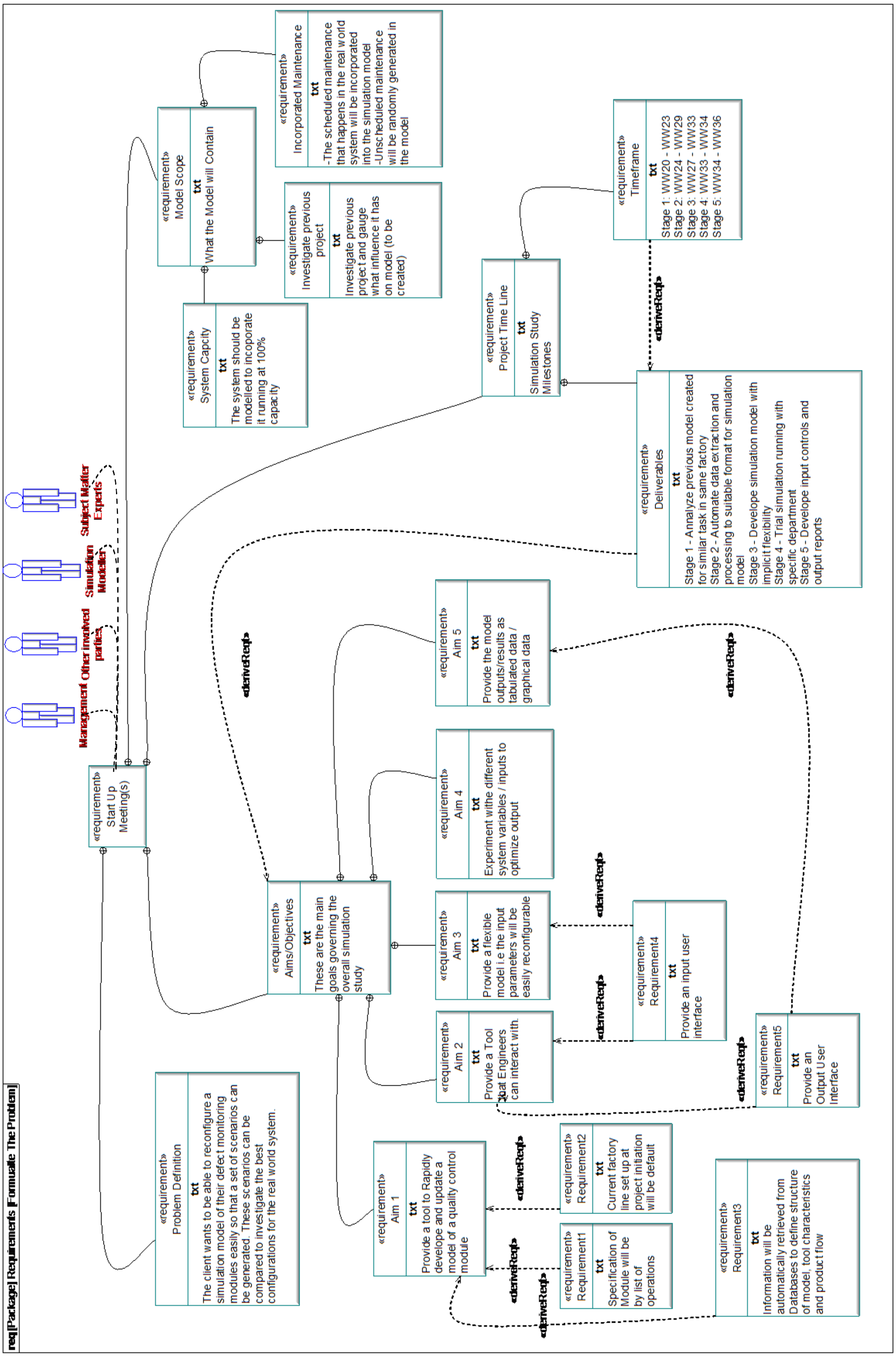
Ease of use _______

Ease of understanding _______

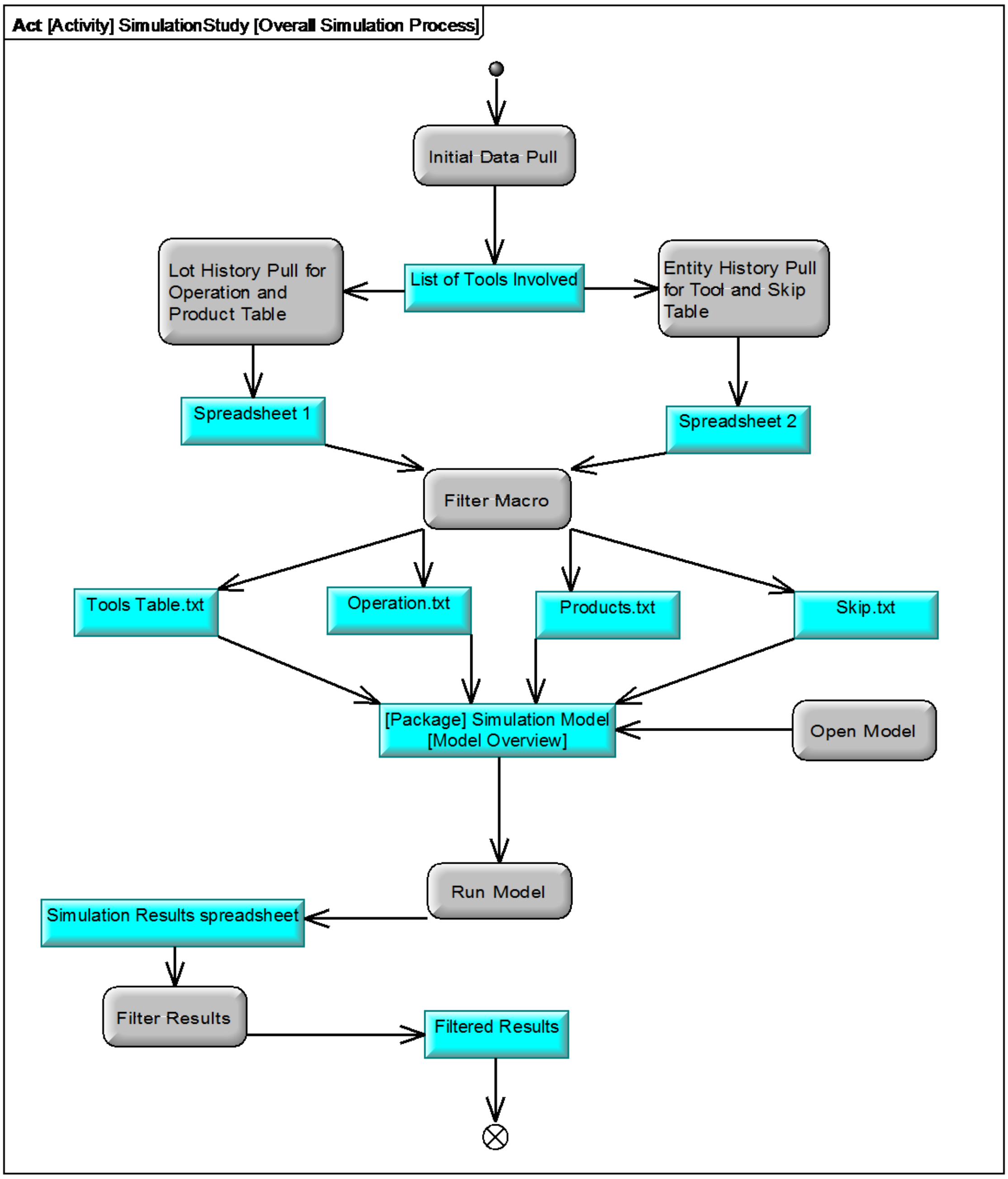
Ease of Reuse _______

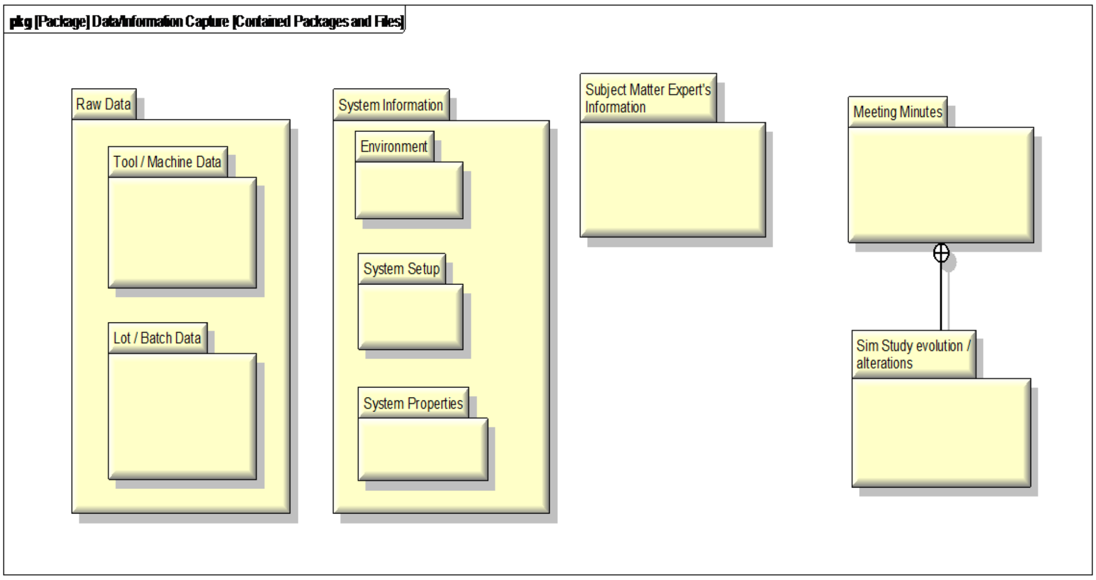
Please include some short comments on the attached SysML diagrams.

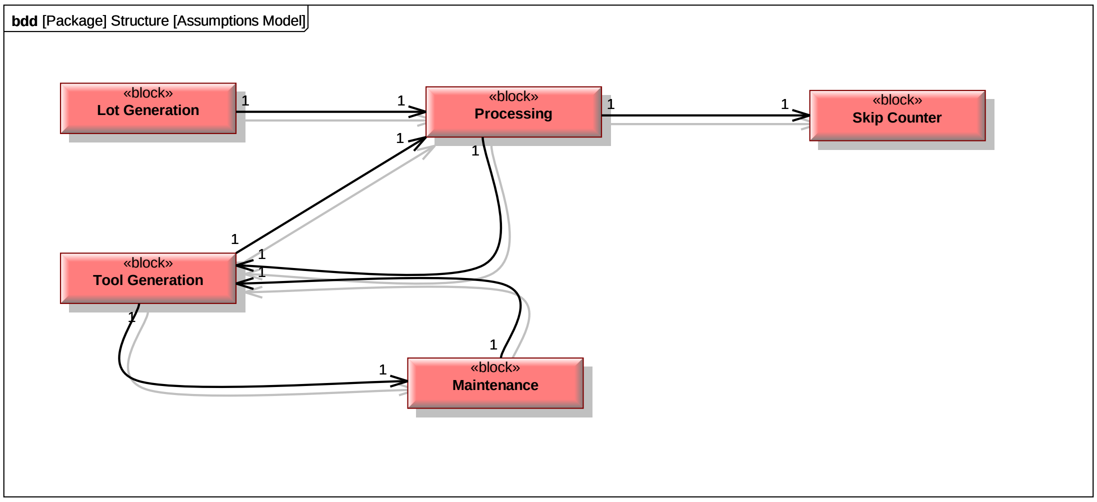
Diagrams For Assessment

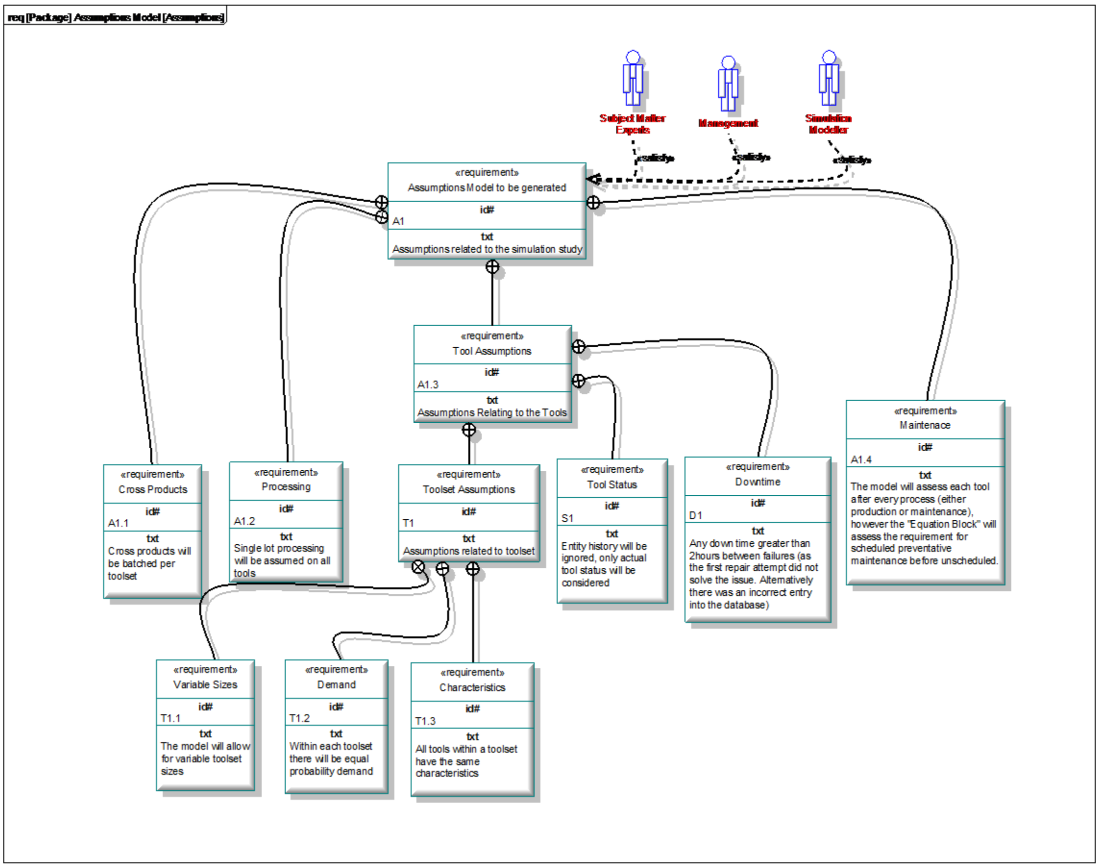


Overall Simulation Process




pkg [Package] Data/Information Capture [Contained Packages and Files]
Raw Data
Tool / Machine Data
Lot / Batch Data
System Information
Environment
System Setup
System Properties
Subject Matter Expert's Information
Meeting Minutes
Sim Study evolution / alterations

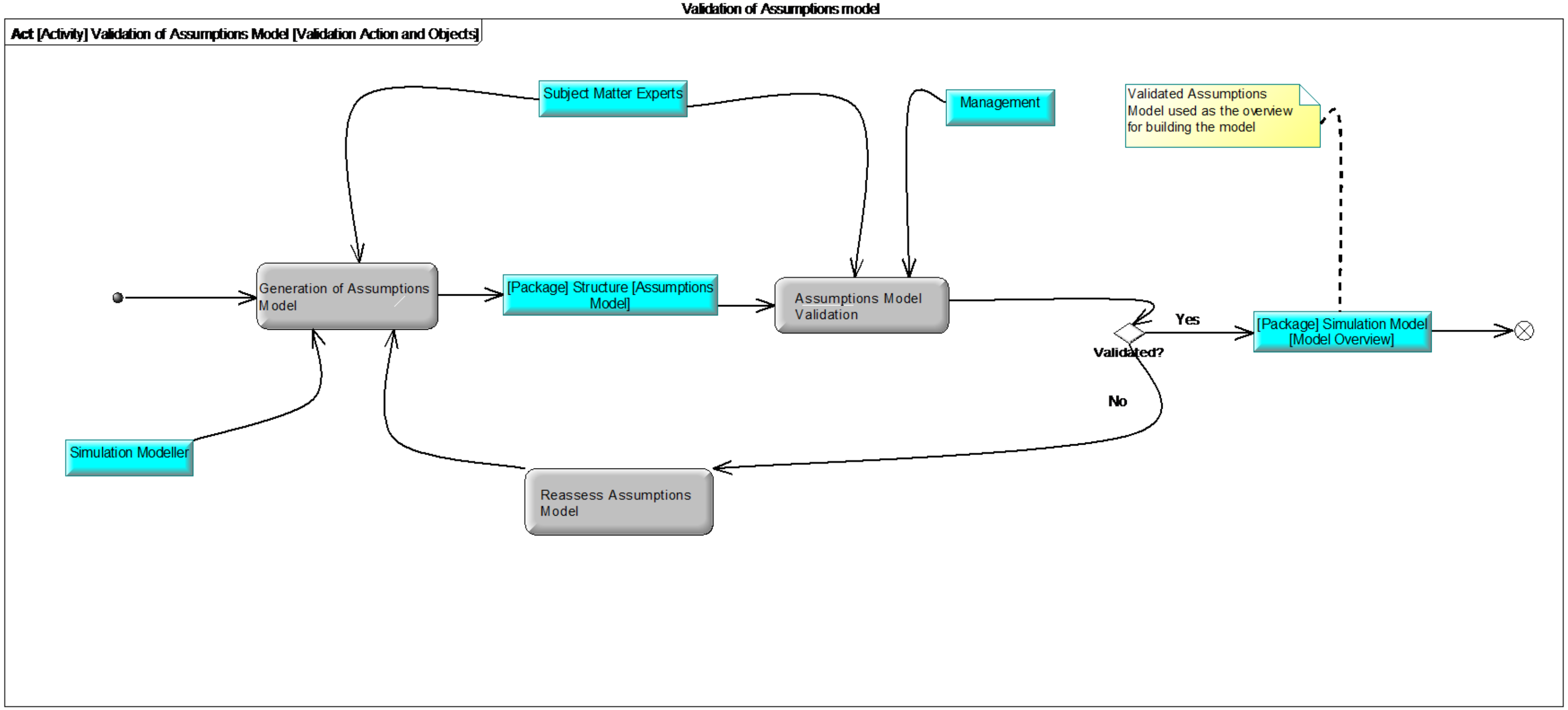

bdd [Package] Structure [Assumptions Model]
«block» Lot Generation
«block» Processing
«block» Skip Counter
«block» Tool Generation
«block» Maintenance
1

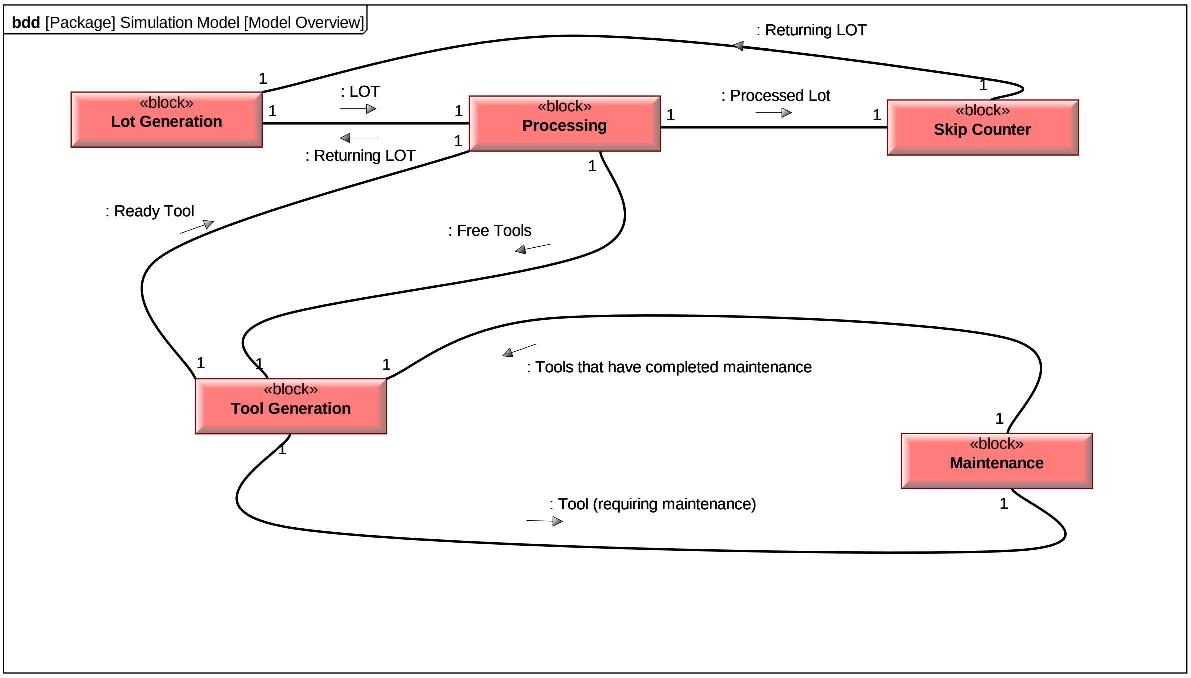


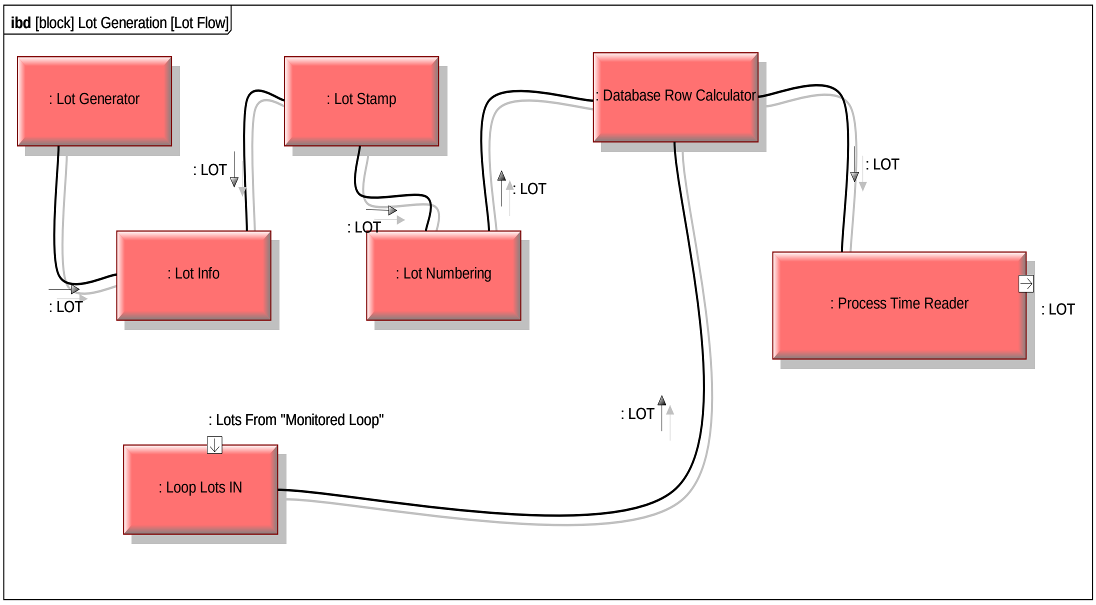
[Package] Assumptions Requirements Table

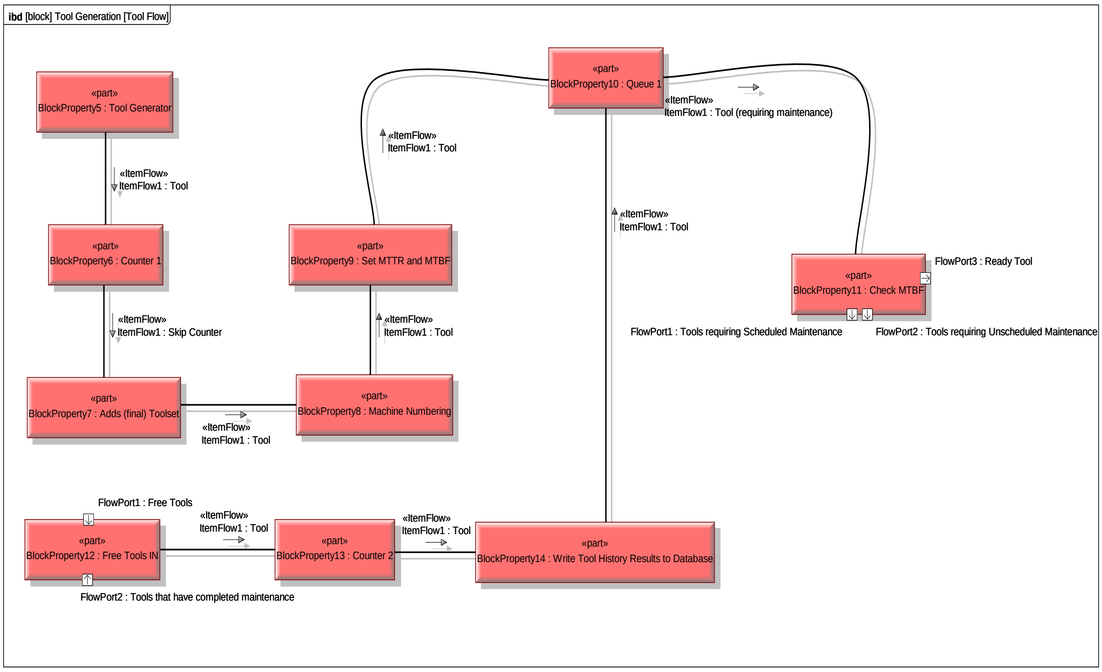
Id#	Name	Txt	Satisfied By
A1	Assumptions Model to be generated	Assumptions related to the simulation study	«Actor» Management (Formualte The Problem) «Actor» Simulation Modeller (Formualte The Problem) «Actor» Subject Matter Experts (Formualte The Problem)
A1.3	Tool Assumptions	Assumptions Relating to the Tools	
A1.2	Processing	Single lot processing will be assumed on all tools	
A1.1	Cross Products	Cross products will be batched per toolset	
A1.4	Maintenace	The model will assess each tool after every process (either production or maintenance), however the "Equation Block" will assess the requirement for scheduled preventative maintenance before unscheduled.	
S1	Tool Status	Entity history will be ignored, only actual tool status will be considered	
D1	Downtime	Any down time greater than 2hours between failures (as the first repair attempt did not solve the issue. Alternatively there was an incorrect entry into the database)	
T1	Toolset Assumptions	Assumptions related to toolset	
T1.3	Characteristics	All tools within a toolset have the same characteristics	
T1.2	Demand	Within each toolset there will be equal probability demand	
T1.1	Variable Sizes	The model will allow for variable toolset sizes	

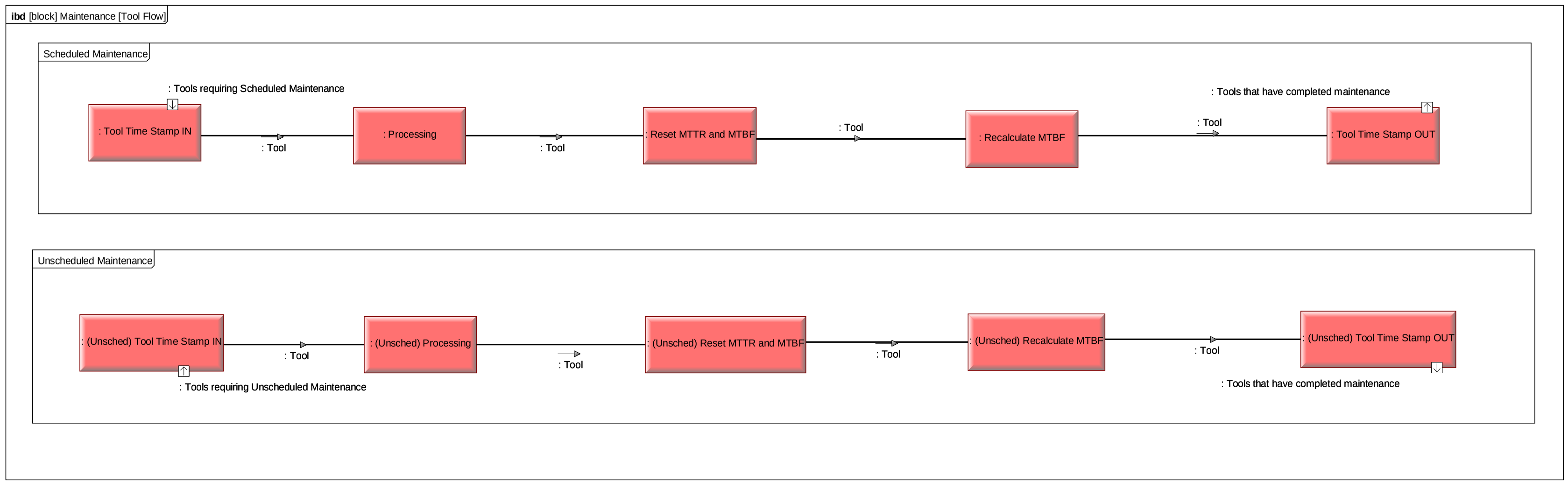
Validation of Assumptions model

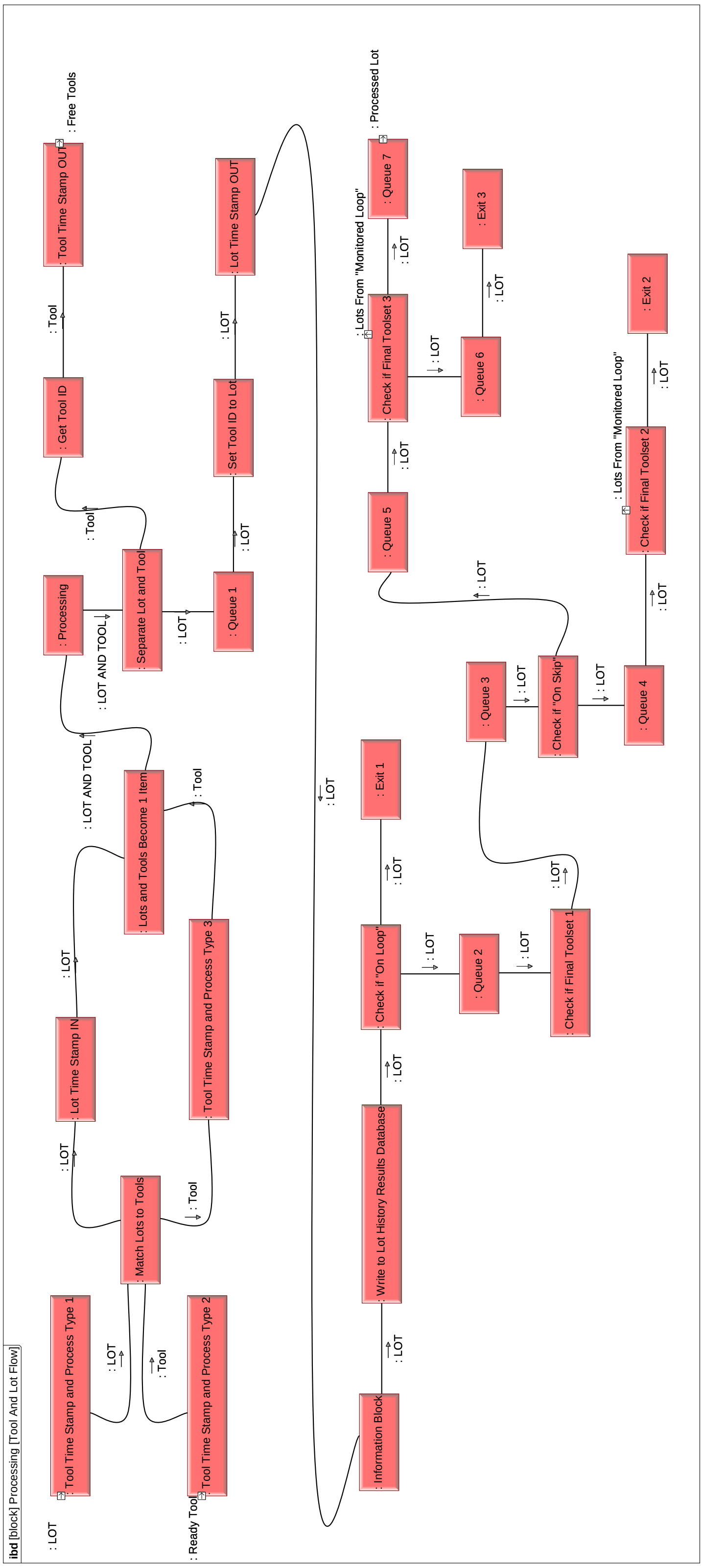


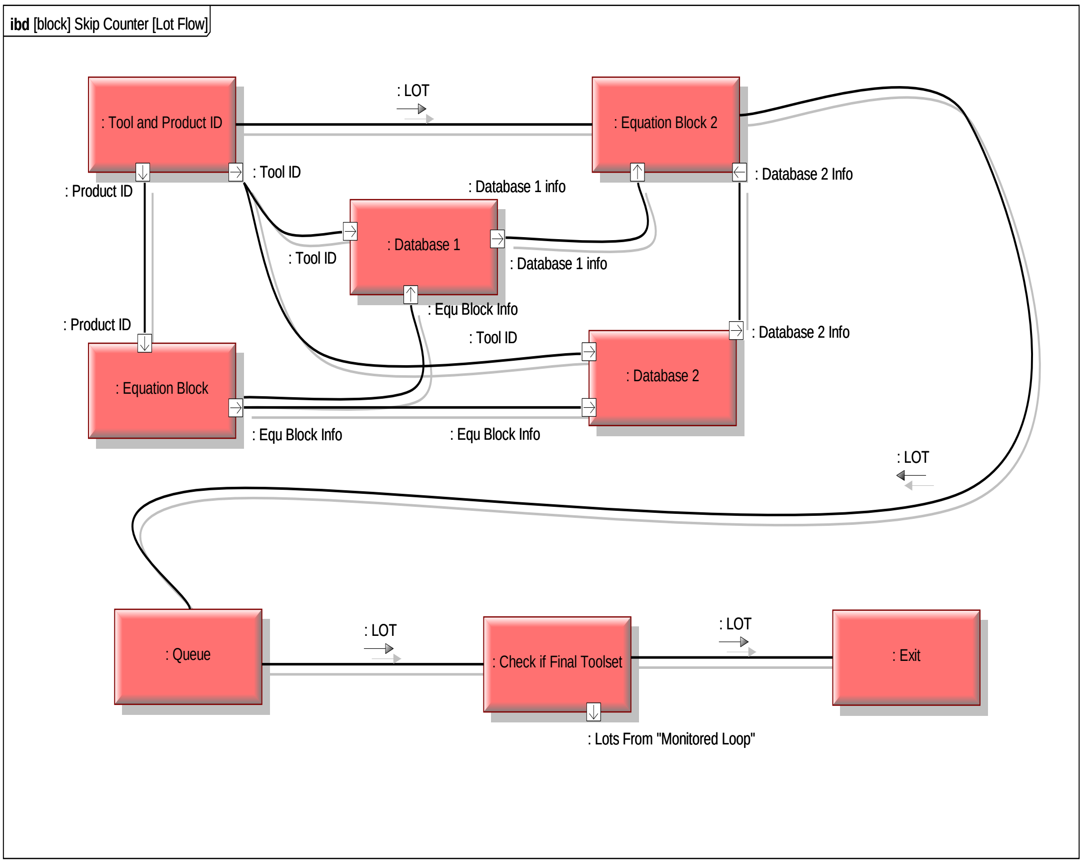

bdd [Package] Simulation Model [Model Overview]
: Returning LOT
«block»
Lot Generation
: LOT
: Returning LOT
«block»
Processing
: Processed Lot
«block»
Skip Counter
: Ready Tool
: Free Tools
«block»
Tool Generation
: Tools that have completed maintenance
«block»
Maintenance
: Tool (requiring maintenance)


ibd [block] Lot Generation [Lot Flow]
: Lot Generator
: Lot Stamp
: Database Row Calculator
: LOT
: Lot Info
: Lot Numbering
: Process Time Reader
: Lots From "Monitored Loop"
: Loop Lots IN


ibd [block] Tool Generation [Tool Flow]
«part»
BlockProperty5 : Tool Generator
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty6 : Counter 1
«ItemFlow»
ItemFlow1 : Skip Counter
«part»
BlockProperty7 : Adds (final) Toolset
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty8 : Machine Numbering
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty9 : Set MTTR and MTBF
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty10 : Queue 1
«ItemFlow»
ItemFlow1 : Tool (requiring maintenance)
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty11 : Check MTBF
FlowPort3 : Ready Tool
FlowPort1 : Tools requiring Scheduled Maintenance
FlowPort2 : Tools requiring Unscheduled Maintenance
FlowPort1 : Free Tools
«part»
BlockProperty12 : Free Tools IN
FlowPort2 : Tools that have completed maintenance
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty13 : Counter 2
«ItemFlow»
ItemFlow1 : Tool
«part»
BlockProperty14 : Write Tool History Results to Database


ibd [block] Maintenance [Tool Flow]
Scheduled Maintenance
: Tools requiring Scheduled Maintenance
: Tool Time Stamp IN
: Tool
: Processing
: Tool
: Reset MTTR and MTBF
: Tool
: Recalculate MTBF
: Tool
: Tools that have completed maintenance
: Tool Time Stamp OUT
Unscheduled Maintenance
: (Unsched) Tool Time Stamp IN
: Tools requiring Unscheduled Maintenance
: Tool
: (Unsched) Processing
: Tool
: (Unsched) Reset MTTR and MTBF
: Tool
: (Unsched) Recalculate MTBF
: Tool
: (Unsched) Tool Time Stamp OUT
: Tools that have completed maintenance


ibd [block] Processing [Tool And Lot Flow]
: LOT
: Ready Tool
: Tool Time Stamp and Process Type 1
: Tool Time Stamp and Process Type 2
: Match Lots to Tools
: Lot Time Stamp IN
: Tool Time Stamp and Process Type 3
: Lots and Tools Become 1 Item
: LOT AND TOOL
: Processing
: Separate Lot and Tool
: Queue 1
: Tool
: Get Tool ID
: Tool Time Stamp OUT
: Free Tools
: Set Tool ID to Lot
: Lot Time Stamp OUT
: Information Block
: Write to Lot History Results Database
: Check if "On Loop"
: Exit 1
: Queue 2
: Check if Final Toolset 1
: Queue 3
: Check if "On Skip"
: Queue 4
: Check if Final Toolset 2
: Lots From "Monitored Loop"
: Exit 2
: Queue 5
: Check if Final Toolset 3
: Queue 6
: Exit 3
: Queue 7
: Processed Lot



ibd [block] Skip Counter [Lot Flow]
: Tool and Product ID
: LOT
: Equation Block 2
: Product ID
: Tool ID
: Database 1 info
: Database 2 Info
: Database 1
: Tool ID
: Database 1 info
: Equ Block Info
: Product ID
: Tool ID
: Database 2 Info
: Database 2
: Equation Block
: Equ Block Info
: Equ Block Info
: LOT
: Queue
: LOT
: Check if Final Toolset
: LOT
: Exit
: Lots From "Monitored Loop"

Feedback from Assessors

Assessor 1

I believe the SysML diagrams are a great way to represent the contents of a project. I would suggest the introduction of some explanatory text (e.g. an appendix with a brief description of the meaning of some blocks and connections, particularly those in the IBDs) in order to improve the ease of understanding of the document when examined by an external party. I would also introduce some cross-reference IDs (e.g. diagrams number and page numbers) in diagrams and blocks, where appropriate (e.g. in BDDs for the corresponding IBDs), to highlight the interconnections between the different diagrams and improve the clarity of the document.

A minor thing...I would stress the hierarchical structure of the REQ diagrams by allocating the block in a tidier way on the diagram. I would also use different background colours for the different blocks according to their own hierarchical level. I believe this would improve the readability of the diagram.

Communicative Ability	___4____
Structure	___5____
Clarity	___4____
Ease of use	___5____
Ease of understanding	___4____
Ease of Reuse	___4____

Assessor 2

I think that these diagrams are excellent for communicating the concept and help audience follow the developer's train of thought, but for new users you still need to have an opening paragraph or two to explain the scenario and the overall aim.

It was fine for me to follow as they deciphered my concept, hence it was easy to use and understand, but not sure about reusability. They can help in reusability, but are not reusable in themselves in my view!

Communicative Ability ___5____

Structure ___4____

Clarity ___4____

Ease of use ___5____

Ease of understanding ___5____

Ease of Reuse ___2____

Assessor 3

Communicative Ability: I gave this a score of 4 as it's a good replacement for lots of text however I felt that the audience who would be interpreting the diagrams may have needed some simulation experience. For example showing a tool travelling to maintenance block and back again may make no sense to someone who is thinking about the shop floor literally as per the layout.

Structure: I think this structure is definitely a lot more flexible in that extra features can easily be added to the diagrams without the requirement for major document re-write.

Clarity: commenting on on the exact format with which I was given I felt it was only fair as the orientation was different across the diagrams which it annoying to read, the size of the fonts again were different sometimes too small.

Ease of use: I gave this a 4 as I thought with some basic background in simulation you got a sense of the design of the model and how it could be implemented.

Ease of understanding: I think someone who doesn't think in a simulation modelling type way may need some guidance as to what is going on.

Ease of Reuse: I wasn't knowledgeable in the project and found that I had a decent understanding of the problem and what I needed in order to implement a solution by the end of the diagram analysis, however please note that my minimal experience in simulation did go a long way in my understanding of the problem.

Communicative Ability	__4____
Structure	__4____
Clarity	__3____
Ease of use	__4____
Ease of understanding	__3____
Ease of Reuse	__4____

Assessor 4

It is likely that the reader would require a basic understanding of the relationships between SYSML diagrams in order interpret them, (more of an issue with SYSML)

The reasons why tools are being generated may not be apparent to a non-simulationist. Therefore might be necessary to include the perspective of the block diagrams explaining the modelling tool that was used i.e. a graphical block based simulation software.

The implementation of the steps of a simulation study in a proper standardised format is an excellent idea, it would be very useful for third party (not already knowledgeable of the project) in examining the pitfalls, issues, steps encountered during the simulation project which would put them in a stronger position (in terms of reuse and further development) than if only the model was documented.

Communicative Ability ___4____

Structure ___5____

Clarity ___4____

Ease of use ___3____

Ease of understanding ___3____

Ease of Reuse ___5____

Assessor 5

The attached diagrams vary in their usefulness to an individual without prior experience of simulation studies. The requirements and use case diagrams are reasonably intuitive, and some information can be gleaned from them by such individuals. However the use of sysML probably reduces the usefulness of the diagrams in this case, since a knowledge of its syntax and structure is necessary to fully appreciate the meaning of the diagram. The block diagrams and internal block diagrams are almost incomprehensible without some previous knowledge of simulation studies, as is the package diagram. Finally the activity diagrams are probably the easiest to understand, since their structure resembles that of a simple flow chart.

Communicative Ability: __4___

Structure: __3___

Clarity: __4___

Ease of Use: __3___

Ease of Understanding: __1___

Ease of Reuse by Third Party: __3___