Facilitating the working process and documentation of DES projects

A study within a complex manufacturing system

Caroline Svensson and Isabella Tehler

DIVISION OF PACKAGING LOGISTICS | DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2016

MASTER THESIS

Facilitating the working process and documentation of DES projects

A study within a complex manufacturing system

Caroline Svensson and Isabella Tehler

LUND UNIVERSITY

Facilitating the working process and documentation of DES projects

A study within a complex manufacturing system

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Applied logistics simulation (MTTN55), Simulation of industrial processes and logistical system (MION40), Operations research (MIO310), Production and inventory control (MIOF10), Management of production and inventory systems (MION01), Quality management (MION50)

Division: Packaging Logistics
Supervisor: Klas Hjort
Co-supervisor: Pernilla Derwik
Examiner: Mats Johnsson

Abstract

Background: Following a growing global competition, complexity, flexibility and product variability of manufacturing systems are increasing. This has led to improvement projects becoming more difficult to carry out. Discrete event simulation (DES) has in literature been identified as having the capability to solve these challenges, but as complexity of a manufacturing system increases so does also the complexity of the DES project. At the same time, the increased dynamic of manufacturing systems requires quicker analysis of improvement projects. This leads to a situation of DES projects having to be carried out on more complex system with shorter project lead time than before.

Purpose: Given the stated situation of having to run more complex DES projects even faster than before, this study aimed to:

1. get an understanding of the process used for working with DES projects,
2. understand what underlying steps have the potential to be facilitated and
3. make an attempt to facilitate working with these with the aim of reducing project lead time.

Methodology: In order to achieve stated purpose a literature study was carried out, resulting in a general framework aimed at facilitating the first steps of the DES project process. The framework was then tested and evaluated in a single case study and developed further into a final version.

Conclusion: Following the literature study it was concluded that the five first steps of the DES project process were the most general and had the most potential to be facilitated by a general framework. The final version of the framework consisted of a three step method;

1. System description
2. Visualisation and data collection
3. Model translation

Keywords: Discrete event simulation, facilitation of the DES project process, DES framework, simulation of complex manufacturing systems, reducing simulation project lead time

Sammanfattning

Bakgrund: Till följd av att den globala konkurrensen tilltar, ökar komplexiteten, flexibiliteten och produktvariationen inom produktion. Detta leder till att förbättringsprojekt blir allt svårare att genomföra. Händelsestyrd simulering har i litteraturen blivit identifierat som ett verktyg med förmågan att lösa dessa utmaningar, men till följd av att komplexiteten i en produktion ökar, växer också komplexiteten i simuleringsprojektet. Samtidigt kräver den växande dynamiken inom produktion en snabbare analys av förbättringsprojekt. Detta leder till en situation där simuleringsprojekt måste utföras på komplexare system men med en kortare ledtid än förut.

Syfte: Givet ovan nämnda situation syftade denna studie till att:

1. få en förståelse för arbetsprocessen kopplat till simuleringsprojekt,
2. förstå vilka underliggande steg som har potentialen att underlättas och
3. försöka stödja arbetet med dessa med målet att minska den totala projektledtiden.

Metod: För att uppnå det uppsatta syftet gjordes en litteraturstudie. Denna resulterade i ett tre-stegsramverk konstruerat med målet att underlätta de fem första stegen i processen för simuleringsprojekt. Detta ramverk testades och utvärderades därefter i en fallstudie för att vidare kunna utvecklas till en slutversion.

Slutsats: Genom litteraturstudien kunde slutsatsen dras att de fem första stegen i simuleringsprojektprocessen var de mest generella och de med störst potential att kunna stödjas med hjälp av ett ramverk. Det slutgiltiga ramverket innehöll en trestegs-metod med följande delar;

1. Systembeskrivning
2. Visualisering och datainsamling
3. Modellöversättning

Nyckelord: Händelsestyrd simulering, underlättning av projektprocessen för simulering, ramverk för händelsestyrd simulering, simulering av komplexa produktionssystem, förkorta projektledtiden för simulering

Acknowledgments

First and foremost, we want to thank Rozle for his huge support, good advice and simply for being such a good friend. There were times when this project felt a bit too complex and would not have gone on without you!

We also want to say a big thank you to LTH, the institution of Packaging Logistics and, most importantly, our supervisor Pernilla for her great assistance throughout the project.

In addition, we want to thank Andreas and Mathias, for believing in us and supporting us, and other ongoing master theses students at The Company, for keeping us sane. Also from The Company, we want to thank other involved such as Michael, Filip, David, Niklas, Manuel, Krister and others whose help and kind comments have contributed to this thesis.

We want to thank people from The Site: Anna for her amazing help, driving personality and cookies. Przemek for his support, kind words and trust. Artur, Tomac, Karol, Lamek-guy, technicians, foremen and management for their involvement. The case could not have been done without you.

From Högskolan i Skövde (HiS), we want to thank Amos and Matias for help, inspiring presentations and for teaching us about the OPT software. Although not mentioned in this report it was really cool to learn and use.

Finally, we want to thank ÅF and Pär for making it all happen.

Lund, June 2016

Caroline Svensson and Isabella Tehler

Table of contents

1 Introduction

In the following sections the introduction to this project will be covered. First the background to the study will be introduced. Then, the found problem will be defined and formulated and the purpose and goal of the study will be listed. Finally, the delimitations made to achieve the final results will be stated.

1.1 Background

In recent years, global competition has been ever growing. Customers have become more and more demanding and request higher product variability, lower prices and shorter lead times. As a result, companies have to be able to respond faster to the customers and make quicker changes, leading to manufacturing processes becoming more dynamic and more complex. Improvement work, such as Total Quality Management (TQM), Lean, Just-in-time (JIT), Six Sigma etc., have for many years been popular ways to meet this new type of demand but as complexity within production increases they become harder to carry out.

When reading published literature for means to enable success of improvement projects for more complex manufacturing systems, discrete event simulation (DES) is a tool often mentioned as beneficial. Ferrin, Muthler and Miller (2002) describe simulation as something that brings a new level of innovation to the table when working with Six Sigma and the tools within. Similarly, Aghaie and Popplewell (1997) state that simulation can make a significant contribution when working with TQM and have concluded several benefits. On the same subject, benefits of simulation to JIT manufacturing processes have been discussed by Smith and Chan (1993) and Welgama and Mills (1995). Finally, in recent years, numerous researchers have found integrating DES with lean improvement projects and value stream mapping (VSM) very successful, for example Solding and Gullander (2009), Erikshammar, Weizhuo, Stehn and Olofsson (2013) and Standridge and Marvel (2006).

All above mentioned publishers, and many more, share the prasing of DES as something bringing benefits to improvement projects and having the capacity to solve existing challenges within these methodologies. However, none of them mention the fact that simulation of complex systems brings challenges of its own. As complexity in manufacturing systems increases so does the complexity of the simulation projects. On this matter Fowler and Rose (2004) list three grand challenges of a technical nature:

1. Reduce problem solving cycle time.
2. Develop real-time simulation-based problem solving capability.
3. Plug-and-play interoperability of simulations and supporting software within a specific application domain.

They also define one big challenge of a more social nature:

1. Greater accptance of modeling and simulation.

Although all important, it can be assumed that no work to solve the first three grand ones will be of any value until the big challenge is solved within a company. There has to be acceptence of simulation before there is any reason to improve the methodology itself. However, when discussing simulation of complex systems Massey and Wang (2007) define reduced project leadtime as a crucial step to gain acceptance, thus combining all three grand challenges concerning lead time and the big challenge of Fowler and Rose. They believe the growing dynamics of manufacturing processes also grows the need for faster answers. If simulation projects take too long, the decision will already have been made and there will be no need for simulation, hence, no acceptance.

All of above leaves the simulationist with the situation of having to run simulation projects and building models of more complex systems and at the same time deliver results much faster. An equation that does not really add up.

1.2 Problem formulation

Given that DES has been stated as beneficial and a way to assist improvement projects of more complex manufacturing systems, but at the same time needs to deliver results faster than before, the problem arises:

How can the working process of discrete event simulation projects, within complex manufacturing systems, be facilitated to shorten the project lead time?

1.3 Purpose and goal

Based on above formulated problem, the purpose of this study was to;

4. get an understanding of the process used for working with DES projects,
5. understand what underlying steps have the potential to be facilitated and
6. make an attempt to facilitate working with these with the aim of reducing project lead time.

1.4 Delimitations

To make sure a study of how to facilitate the working process of simulation of complex systems was possible to carry out, within the planned time frame, some delimitations had to be made. The following section will explain these.

1.4.1 **Type of systems**

When working with DES many different processes could be simulated, all with their own dimensions of complexity. However, given the current situation of simulation being well established in the manufacturing world, and the fact that many kinds of improvement projects are mainly aimed at improving production processes, the decision was made that this study will only focus on simulation of complex *manufacturing* systems. The final result may of course be applicable also when working with simulation of other types of systems but it was not in the initial scope.

1.4.2 **Definition of complex systems**

To be able to facilitate working with DES of complex manufacturing systems a definition of what system actually qualifies as complex had to be made. Since publishers state that;

1. DES can solve arising challenges when working with a system too complex for VSM (Solding & Gullander, 2009) (McDonald, Van Aken, & Rentes, 2002),
2. a VSM map should only cover one flow group (Rother & Shook, 1999) and
3. a rule of thumb is that the targeted value stream should include no more than twelve tasks or process stations (Yu, Tweed, Al-Hussein, & Nasseri, 2009)

the definition was set to:

A manufacturing system is considered complex when the actual system is "too big" for VSM, i.e. consisting of more than one flow group and/or more than twelve process stations.

2 Methodology

In the following sections the methodology used to carry out this project will be covered. First, the literature study used to find existing theory will be described. Then the process of setting up a facilitating framework will be outlined. Lastly, the application of the framework on a case study will be covered.

2.1 Literature study

To be able to get a deeper understanding of the methodology, and its included steps, used within DES projects a literature study was carried out. Following the classifications published by Cooper (1985) the literature study made can be categorised as;

- focusing on *theories* and *practices or applications*,
- with the goal to *synthesize* past literature and *identify central issues*,
- in a *neutral* perspective,
- covering a *central or pivotal sample*,
- organised *methodologically* and
- written for *practitioners*.

This categorisation can be made since the study focused on understanding further what areas within the DES project process that needed to be facilitated, and also finding work published around what have already been done or said to facilitate these areas. The purpose was to summarise what have been said and identify issues with improvement potential and then present the results in an unbiased manner. The selection of publications and books to review was made in an unstructured fashion, searching for several keywords related to the subject and also through following reference lists in publications found relevant. Several different search engines were used to find a central sample deemed fitting.

Since many publishers within DES describe their project methodology as one following the model published by Banks (1996), or one very similar to this model, the study was organised following the steps in such a model and the result will also be presented according to these steps, see a simplified version in figure 1 below. Articles published regarding problem formulation and scope, conceptual modelling, data collection, model translation/design specification, model building, verification and validation, experimentation and documentation was studied and findings within that was deemed interesting was summarised. Once a good overview of the general methodology of working with DES projects had been obtained a more in-depth search for simulation and mapping of complex manufacturing systems was made. The aim was to find inspiration for mainly conceptual modelling and visualisation but also to find challenges concentrated around working with more complex systems.

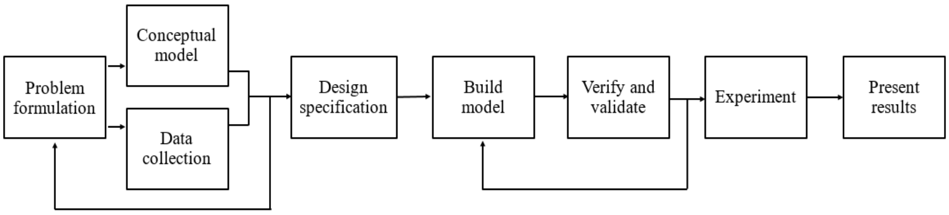


Figure 1: A simplified flowchart of Bank's model (1996)

2.2 Setting up a framework

Following the literature study it was decided that a framework for facilitating working with DES projects, mainly within complex manufacturing systems, should be set up. The initial framework was entirely based upon findings from the literature study. It attempted to include as many of the aspects publishers had stated as important in the DES project methodology, when reaching for shorter project lead time. With the aim of using an easy accessible and commonly used software for data handling, the framework was structured in MS Excel.

2.3 Single case study

After an initial framework, created to facilitate the DES project process, was set up it was tried in a single case study with the purpose of testing and evaluating it. The case was done in cooperation with a Swedish manufacturing company based in south Sweden, referred to as The Company, with a production site in Eastern Europe contracted by The Company, referred to as The Site. A more thorough description of the case can be found in section 5.1 Case description.

Every part of the initial framework, for concerned steps of the DES project process, was tried during the case study and changed if needed, making the case study an embedded study (Yin, 2014). The framework was also additionally developed during the study when further needs arose that were not already covered. All changes and developments were then tried, evaluated and, if not deemed enough, changed again. This resulted in an iterative process of trying and changing the framework from its initial structure until a structure considered fitting for the case was established. The unit of analysis, i.e. if a part of the framework or a change was deemed enough, was very abstract and fully relied upon the feeling that a change or new development contributed in a positive manner to the project process used during the case.

3 Literature study

In the following sections the results from the literature study will be presented. The first part will cover the results of studying the general methodology of working with simulation projects, stated in section 2.1 Literature study. The second part will then cover what has been found regarding working with more complex manufacturing systems.

3.1 DES project process

This first part will present found literature regarding the different steps in the DES project methodology, presented in section 2.1 Literature study.

3.1.1 **Problem formulation**

According to Banks (1998) every simulation study needs to start with a statement of the problem. He believes that regardless of whether it is stated by stakeholders or the simulation analysts it is extremely important to ensure that both parts understand it clearly and agree with the formulation. On a similar matter, de Vin (2012) declares that an ill-defined project, or a project starting too ambitious, is a common pitfall when running simulation projects.

Robinson (2008) describes that the requirement for a simulation model should always be driven by the need to improve a problem situation. Similar to previous mentioned authors it is, according to him, important that the problem situation is clearly understood and expressed. He also lists examples of objectives to agree upon in the problem formulation step, seen below in table 1.

Table 1: Objectives to cover during the problem formulation step (Robinson S. , 2008)

Aim and modelling objectives	***General project objectives***
Achievements	Flexibility
Performance	Run-speed
Constraints	Visual display
	Ease-of-use
	Model/component reuse

Law and McComas (1991) describe step one of a simulation study, formulate problem and plan the study, as the most important aspect of the study. They believe this step is often neglected due to a lack of understanding of the nature of simulation and the information this step may provide. Furthermore, they say that it is not possible to decide upon an appropriate level of detail without knowing precisely what issues are to be addressed by the model. The following six bullets have been listed as crucial to complete before the simulation project is started:

- Identify performance problems for the existing system
- State overall objectives and five to ten specific issues to be addressed
- Decide how the model will be used in the decision making process
- Determine who will be the model's end user
- Specify measures of performance to compare alternatives
- Delineate the system configurations to be studied (Law & McComas, 1991)

Similarly to Law and McComas, Robinson (2015) also mentions how vital it is that the objectives of the model are known. He means that it is created for a specific purpose and without knowing this purpose it is impossible to create appropriate simplifications. In his framework for conceptual modelling he lists five bullets very related to those listed by Law and McGomas:

- Understand the problem situation
- Determine the modelling and general project objectives
- Identify the model output, i.e. responses
- Identify the model inputs, i.e. experimental factors
- Determining the model content, i.e. scope and level of detail (Robinson S. , 2015)

3.1.2 Model Conceptualisation

The second step of working with simulation has already been touched upon by above mentioned framework for conceptual modelling but; once a solid understanding of the system and its problems is obtained, the next step of a simulation study is to conceptualise the investigated system in a conceptual model (Banks, 1998). According to Robinson (2008) conceptual modelling is generally agreed to be the most difficult, least understood and most important task to be carried out. A lot of literature can be found regarding this subject and the different methodologies of conceptual mapping seem to be as many as the publishers. Hernandez-Matias, Rios, Perez-Garzia and Vizan (2008) state that there is not one conceptual modelling method that is capable of entirely model a complex process.

Banks, Carson II, Nelson and Nicol (2005) describe model conceptualisation as being as much art as it is science and advice starting with a simple model and build towards greater complexity. Their advice is quite contradictive when comparing to other authors who instead seem to advice to limit the model complexity and level of detail already before starting to build the conceptual model. Chwif and Paul (2000) describe inability to model the problem (conceptual modelling) correctly as a common reason to why complexity increases. They believe modellers try to build the model "*as close to reality as possible*" and explain that it is the results, not the model itself, that should be close to reality.

Robinson (2008) defines a conceptual model as *"a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model."* Already in his framework previously described he outlines that a conceptual model should help with understanding the problem situation, objectives, in- and output and the model content. He believes a more detailed model does not always portray reality better and sometimes adding more detail comes with a higher cost than the improved output is worth. Figure 2 below shows how Robinson believes model accuracy improves with the complexity of the model. Adding to much complexity may even result in the accuracy of the model being reduced. This due to lack of knowledge or not having the data needed to support the high level of detail and instead replacing these with incorrect assumptions. Therefore, the goal should be building a model as close to point x as possible where a high level of accuracy is gained for a low level of complexity.

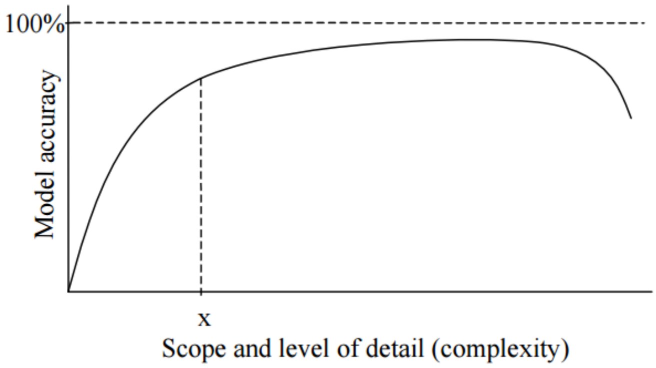


Figure 2: How simulation model accuracy changes with model complexity (Robinson S. , 2015)

As stated above, there can be found many different means to build a conceptual model. Law and McComas (1991) seem to mainly focus on the necessity to while collecting data also document all assumptions made. After being validated in a structured walk through this document embodies the conceptual model. Contrary to this way of building a conceptual model, Barra Montevechi et al. (2010) present a much more visual way of mapping the system with pre-defined context and symbols, named the IDEF-SIM system. Their system orginins from previous versions of IDEF0, IDEF3 and other flowcharts and diagram methods and inlcudes symbols for entities, functions, flow, resources, controls, rules, motions etc. In figure 3 below an example of this method of conceptual modelling can be seen.

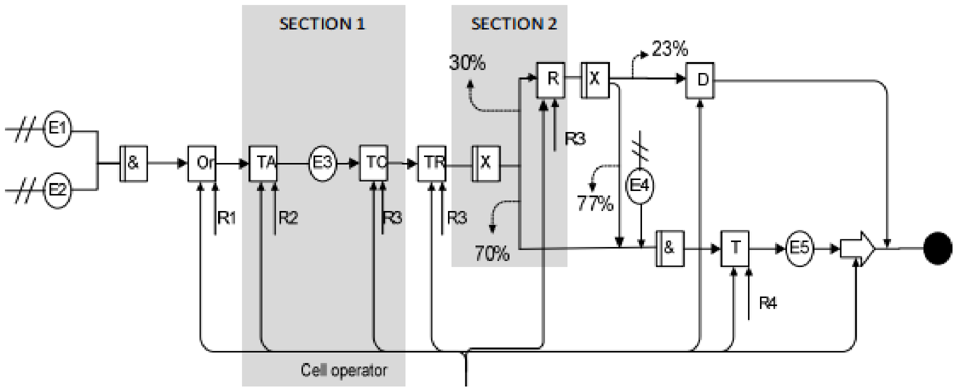


Figure 3: Example of mapping according to IDEF-SIM system (Barra Montevechi, et al., 2010)

Cetinkaya, Verbraeck and Seck (2010) also discuss IDEF diagrams when comparing different modelling methods. In addition, they mention event graphs, activity cycle diagrams, process flow diagrams, Petri nets etc. all very visual ways of modelling. They also touch upon the problem of translating a conceptual model into code. According to them different simulation modellers would, based on the same conceptual model, create very different simulation models. To solve this problem, they highlight the need for involvement of stakeholders in the conceptual modelling phase. In figure 4 below representations of their result of a component based conceptual modelling language can be seen.

Relation
Fixed Composition CompA **HAS** CompB
Temporary Composition CompA **GOES** CompB
Logical link CompA **ISLINKED** CompB
Physical link CompA **ISJOINED** CompB
‘Send-To’ relation CompA **SENDS** EntityC **TO** CompB
‘Send-To-Via’ relation CompA **SENDS** EntityC **TO** CompB **VIA** CompD

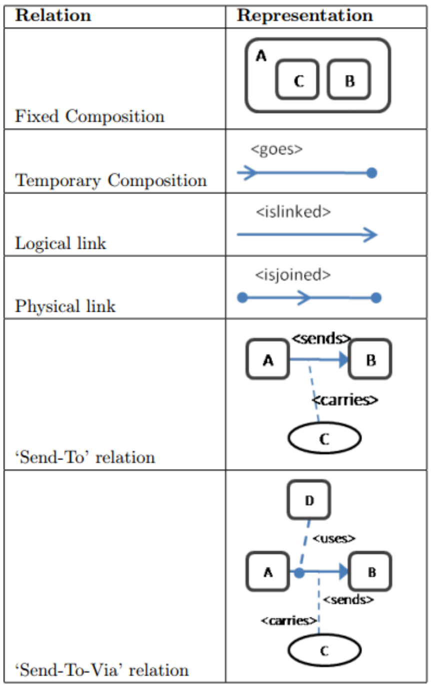


Figure 4: Textual representations and relations to use in conceptual modelling (Cetinkaya, Verbraeck, & Seck, 2010)

More similar to Law and McComas, Robinson (2014) uses a more text based way to conceptualise a system. He suggests listing components such as entities, activities, queues/buffers, resources etc. and then adding a column for if this specific component should be included or excluded in the scope. Then, in a second table, or building on the first table, he adds levels of details to every component and, for every detail, if it should be included or not. Finally, if it should be included, he lists how it is to be included. Figure 5 and figure 6 below shows example of these tables. Together with this, he suggests a conceptual model should also contain documentation of modelling assumptions and model simplifications. In addition Robinson (2008) discusses documentation of the conceptual model, what benefits it may bring and lists the following benefits:

- Minimizes the likelihood of incomplete, unclear, inconsistent and wrong requirements
- Helps build the credibility of the model
- Guides the development of the computer model
- Forms the basis for model verification and guides model validation
- Guides experimentation by expressing the modelling objectives, and model inputs and outputs
- Provides the basis of the model documentation

- Can act as an aid to independent verification and validation when it is required
- Helps determine the appropriateness of the model or its parts for model reuse and distributed simulation (Robinson S. , 2008)

Component	Include/exclude	Justification
Customers	Include	Flow through the service process
Staff – Service	Include	Experimental factor, required for staff utilization response
– Food preparation	Exclude	Material shortages are not significant
– Cleaning	Exclude	Do not interconnect with speed of service
Queues at service counters	Include	Required for waiting time and queue size response
Tables	Exclude	Not related to waiting for food
Kitchen	Exclude	Material shortages are not significant

Figure 5: Example of conceptual modelling as a table describing scope for a simulation project at a fast food restaurant. (Robinson S. , 2014)

Component	Detail	Include/exclude	Comment
Customers	Customer inter-arrival times	Include	Modelled as a distribution
	Size of customer order	Exclude	Represented in service time
Service staff	Service time	Include	Modelled as a distribution, taking account of variability in performance and size of order
	Staff rosters	Include	Experimental factor
	Absenteeism	Exclude	Not explicitly modelled, but could be represented by perturbations to the staff rosters
Queues	Queuing	Include	Required for waiting time and queue size response
	Capacity	Exclude	Assume no effective limit
	Queue behaviour (jockey, balk, leave)	Exclude (except assume join shortest queue)	Behaviour not well understood. Results will show imbalance of arrival and service rates (see Section 6.3.5)

Figure 6: Example of conceptual modelling as a table describing level of detail for a simulation project at a fast food restaurant. (Robinson S. , 2014)

3.1.3 Data collection

Data collection and conceptual modelling are parallel steps in the simulation project process (Banks, 1998). The data required for development of the computer model should according to Robinson (2014) be identified by the conceptual model. In a perfect world, where accurate data for any part of a process could easily be obtained, the conceptual model would then be designed without consideration of whether the data can be gathered or not. Unfortunately, not all data are readily available, collectable or adequate which could make the proposed conceptual model problematic and time consuming. According to Robinson and Bhatia (1995) data can be divided into three different categories:

- **Category A:** Available
- **Category B:** Not available but collectable
- **Category C:** Not available and not collectable (Robinson & Bhatia, 1995)

Based upon these three categories Skoogh and Johansson (2008) have made detailed guidelines for how to perform the crucial process of handling input data in a structured way. Their methodology includes thirteen steps listed below:

1. Identify and define relevant parameters
2. Specify accuracy requirements
3. Identify available data
4. Choose methods for gathering non-available data
5. Determine if all specified data will be found
6. Create data sheet
7. Compile available data
8. Gather non available data
9. Prepare statistical or empirical representation
10. Determine if the representation is sufficient
11. Validate data representation
12. Determine if data representation is valid
13. Finish final documentation (Skoogh & Johansson, 2008)

When it comes to the first two bullets, identifying relevant parameters and specifying accuracy requirements, Banks et al. (2005) believe there is a constant interplay between conceptual modelling and data collection. As the complexity of the conceptual model changes the need for data might also change. Needed data is also dependent on the objectives of the study and the type of system. (Banks, Carson II, Nelson, & Nicol, 2005)

Hatami (1990) has made an attempt to simplify the process of gathering data and identifying relevant parameters when working with simulation of manufacturing systems through listing examples of needed data. He believes the following data is needed:

- Process flow diagram to illustrate the scope of the project
- Storage/buffer space capacities
- Machine/Operator speeds
- Information about the processes
- Data on machine reliability
    - Time to failure/machine up-time
    - Mean time to repair
- Product mix and schedules
- Facility layouts
- Operating philosophies
- Material handling systems
- Lost production time data analysis
- Throughput data analysis (Hatami, 1990)

On the subject of gathering the actual data, Law and McComas (1991) highlights the importance of searching for and collecting data from several people with different positions within the company, such as machine operators, industrial and manufacturing engineers, production planners, managers and vendors. This since no single person or document will have all required information, and since information might not always match. They believe that obtaining good information on system operating procedures and control logic is key for a successful simulation project.

Also discussing information and how to transform data into information, Davenport and Prusak (1998) discuss five methods, all starting with the letter C:

- **Contextualisation:** Knowing the purpose of gathering the data
- **Categorisation:** Knowing the units of analysis or key components of the data
- **Calculation:** Knowing how the data was analysed
- **Correction:** Knowing what errors have been removed from the data
- **Condensation:** Knowing if the data have been summarised in a more concise form (Davenport & Prusak, 1998)

Perera and Liyanage (2000) state that data collection, i.e. gather and validate data, can stand for up to 40% of the project time. They have listed seven major causes to a time consuming data collection, all related to previous mentioned challenges with problem formulation, conceptual modelling and data collection:

1. Incorrect problem definition
2. Lack of clear objectives
3. High system complexity
4. Higher level of model details
5. Poor data availability
6. Difficulty in identifying available data sources
7. Limited data handling capability (Perera & Liyanage, 2000)

As an attempt to solve these issues they set up a method for rapid identification and collection of input data consisting of a functional model library, a reference data model and a mapping table to match functional models with reference data model. Figure 7 below shows an example of such mapping.

Functional modelling element	Required data	Reference model path (corresponding RM entity)	Corresponding ARENA template
Part	Part ID	PART	CREATE
	Part description	PART	CREATE
	Batch size	MACHINE_OPERATION	CREATE
	Max batches	MACHINE_OPERATION	CREATE
	Inter arrival time	MACHINE_OPERATION	CREATE
Machine	Machine ID	MACHINE	RESOURCE/SEIZE
	Machine description	MACHINE	RESOURCE/SEIZE
	MTBF	MACHINE_GROUP	RESOURCE
	MTTR	MACHINE_GROUP	RESOURCE
	Input buffer capacity	MACHINE_GROUP	RESOURCE OR STORAGE
	Output buffer capacity	MACHINE_GROUP	RESOURCE OR STORAGE
Operator	Operator ID	OPERATOR	RESOURCE/SEIZE
	Operator description	OPERATOR	RESOURCE/SEIZE
	Efficiency	OPERATOR	
	Skills	OPERATOR	
	Learning curve effect	OPERATOR	
Part Processing	Machine operation desc.	MACHINE_OPERATION	
	Machine time	MACHINE_OPERATION	DELAY
	Operator responsibilities	MACHINE_OPERATION	
Schedule	Schedule name	SCHEDULE	RESOURCE–>Schedule block
	Schedule description	SCHEDULE	RESOURCE–>Schedule block
	Schedule Time	SCHEDULE	RESOURCE–>Schedule block
	Duration	SCHEDULE	RESOURCE–>Schedule block
	Machine available time	MACHINE_SHEDULE	RESOURCE–>Schedule block
	Machine unavailable time	MACHINE_SHEDULE	RESOURCE–>Schedule block
	Operator available time	OPERATOR_SHEDULE	RESOURCE–>Schedule block
	Operator unavailable time	OPERATOR_SHEDULE	RESOURCE–>Schedule block
Instruction	Priority rule	MACHINE_OPERATION	SEIZE

Figure 7: Example of sample mapping table (Perera & Liyanage, 2000)

One matter that has not been clearly discussed by previous mentioned authors is data credibility and accuracy. While many have spoken about non-available data and non-collectable data none has yet mentioned data that is available but does not meet the need for sufficient credibility. In an attempt to provide guidance in creating a verification, validation and accreditation plan Balci, Ormsby, Carr III and Saadi (2000) discuss data accuracy. They present 11 indicators to determine data quality:

- **Accessibility**: the degree to which data is easy and quickly retrievable
- **Accuracy:** the degree to which data possesses sufficient transformational and representational correctness
- **Clarity**: the degree to which data is unambiguous and understandable
- **Completeness**: the degree to which all parts of the data are specified with no missing information
- **Consistency:** the degree to which data is using consistent measurement units, uniform terminology and to which one value does not conflict with another
- **Currency:** the degree to which the age of the data is appropriate for usage

- **Precision:** the degree to which data possesses sufficient number of significant digits
- **Relevance:** the degree to which the data is applicable for use
- **Resolution:** the degree to which the data possesses sufficient level of detail
- **Reputation:** the degree to which data is trusted or highly regarded in terms of their source or origin
- **Traceability:** the degree to which data is easily attributed to a source

Other authors discussing data accuracy are Robertson and Perera (2002). They state that data collection has a key role within simulation as the data must truly emulate the realities of the system to the level of accuracy and detail required. Failure to obtain this will lead to the model not accurately portraying the system and providing invalid results. When performing data collection, they believe accuracy, sources, data systems, data duplication and timeliness of data all need to be considered.

3.1.4 Design specification and model translation

According to Banks et al. (2005) the step that comes after data collection and conceptual modelling is building the actual model. However, many authors add one or several steps in-between. Robinson (2014) highlights how important it is to, before rushing to the computer and start coding, determine how to structure the model in the chosen software. He lists four objectives he believes are important to establish; speed of code, transparency, flexibility and run-speed. Robertson and Perera (2002) believe there should be a model formulation and a model representation before model programming can be started. Sargent (2008) instead declares that there should be one step called simulation model specification before the model building.

Balci (2011) discusses how to successfully conduct large scale modelling and simulation projects and states that after the conceptual modelling phase but before building models and sub models there should be an architecture specification and a design specification. According to him a design should be created based upon the requirements of the simulation model, the conceptual model and the previous mentioned architecture. A simpler way of defining the model content before starting to build the model is given by Seebacher, Winkler and Oberegger (2015), see figure 8 below.

System elements	Description	Characteristics
Entities	Product *A* Product *B* Product *C*	50 % 30 % 20 %
Resources	(1) R_Cutting (2) R_Milling (3) R_Lather (4) R_Drilling (5) R_Barrel_finishing (6) R_Priming (7) R_Enamelling (8) R_Printing (9) R_Assembly (10) R_Packaging	1 2 2 1 1 1 2 1 3 2
Internal transport	Number of transport vehicles Velocity of transport vehicles	2 83.3 m/min.
Input variables	Time between arrivals Average processing times Shift duration Break	EXP 7.50 min. Table III 480 min. 60 min.
Number of replications	50	
Start-up period	480 min.	
Replication length	9,600 min.	
Type of simulation	Non-terminating simulation	

Figure 8: Example of an overview of underlying system characteristics (Seebacher, Winkler, & Oberegger, 2015)

A similar example to figure 8 above is made by Perera and Liyanage (2000) in their attempt to set up a method for rapid identification and collection of input data. Since their mapping table to match functional models with reference data model also gives suggestions of corresponding templates, when using the simulation software ARENA, it can be seen as a summary of model content. A version of figure 7 with this model content part highlighted can be seen below in figure 9.

Functional modelling element	Required data	Reference model path (corresponding RM entity)	Corresponding ARENA template
Part	Part ID	PART	CREATE
	Part description	PART	CREATE
	Batch size	MACHINE_OPERATION	CREATE
	Max batches	MACHINE_OPERATION	CREATE
	Inter arrival time	MACHINE_OPERATION	CREATE
Machine	Machine ID	MACHINE	RESOURCE/SEIZE
	Machine description	MACHINE	RESOURCE/SEIZE
	MTBF	MACHINE_GROUP	RESOURCE
	MTTR	MACHINE_GROUP	RESOURCE
	Input buffer capacity	MACHINE_GROUP	RESOURCE OR STORAGE
	Output buffer capacity	MACHINE_GROUP	RESOURCE OR STORAGE
Operator	Operator ID	OPERATOR	RESOURCE/SEIZE
	Operator description	OPERATOR	RESOURCE/SEIZE
	Efficiency	OPERATOR	
	Skills	OPERATOR	
	Learning curve effect	OPERATOR	
Part Processing	Machine operation desc.	MACHINE_OPERATION	
	Machine time	MACHINE_OPERATION	DELAY
	Operator responsibilities	MACHINE_OPERATION	
Schedule	Schedule name	SCHEDULE	RESOURCE->Schedule block
	Schedule description	SCHEDULE	RESOURCE->Schedule block
	Schedule Time	SCHEDULE	RESOURCE->Schedule block
	Duration	SCHEDULE	RESOURCE->Schedule block
	Machine available time	MACHINE_SHEDULE	RESOURCE->Schedule block
	Machine unavailable time	MACHINE_SHEDULE	RESOURCE->Schedule block
	Operator available time	OPERATOR_SHEDULE	RESOURCE->Schedule block
	Operator unavailable time	OPERATOR_SHEDULE	RESOURCE->Schedule block
Instruction	Priority rule	MACHINE_OPERATION	SEIZE

Figure 9: Example of sample mapping table (Perera & Liyanage, 2000) with corresponding ARENA templates highlighted to give an example of how model content can be listed

On the same topic, Robinson (2008) adds comments to his conceptual model tables about how details should be modelled in code. An example of this has already been shown in figure 6 in section 3.1.2 Conceptual modelling. He also describes how decisions about code modelling, and what to include and exclude, have to be made for each entity, activity, queue and resource. In addition, he provides a template with a list of what could be needed to consider for each component type, see figure 10 below. The list is meant to be used as a useful starting point when determining the level of detail and model content and should not be seen as a complete list. (Robinson S. , 2008)

Component	*Detail*	*Description*
Entities	Quantity	Batching of arrivals and limits to number of entities
		Grouping so an entity represents more than one item
		Quantity produced
	Arrival pattern	How entities enter the model
	Attributes	Specific information required for each entity, for example type or size
	Routing	Route through model dependent on entity type/attributes, for example job shop routing
	Other	For example, display style
Activities	Quantity	Number of the activity
	Nature (X in Y out)	For example, representing assembly of entities
	Cycle time	
	Breakdown/repair	Nature and timing of breakdowns
	Set-up/changeover	Nature and timing of set-ups
	Resources	Resources required for the activity
	Shifts	Model working and break periods
	Routing	How entities are routed in and out of the activity
	Other	For example, scheduling
Queues	Quantity	Number of the queue
	Capacity	Space available for entities
	Dwell time	Time entities must spend in the queue
	Queue discipline	Sequence of entities into and out of the queue
	Breakdown/repair	Nature and timing of breakdowns
	Routing	How entities are routed in and out of the queue
	Other	For example, type of conveyor
Resources	Quantity	Number of the resource
	Where required	At which activities the resource is required
	Shifts	Working and break periods
	Other	For example, skill levels, interruption to tasks

Figure 10: Template for level of detail by component type (Robinson S. , 2008)

3.1.5 **Model building**

As previously mentioned, the step after model conceptualisation and data collection, according to Banks et al. (2005), is building the model. In their model for simulation project methodology this step is named model translation but for clarity it will be referred to as model building or coding. They state that one of the most important aspects of building the model is choosing the right type of software for the project and that the right decision can greatly reduce lead time. Law and McComas (1991) also highlight the importance of choosing an appropriate simulation software and utilising it correctly. They also list modelling system randomness in a reasonable manner as a key factor for success and discuss techniques for debugging such as modular programming, interactive debuggers, structured walk troughs etc. On the same subject, De Vin (2012) lists pitfalls in a simulation project, some related to model building;

- Model building and data acquisition take more time than planned
- The model's use is extended to address questions for which it was never designed
- An old model is dusted off for later use without being updated (de Vin, 2012)

Robinson (2014) covers, in-depth, model building and discusses how to structure the model, how to code and develop in small steps, how to work with random numbers and how to document the model building. He also highlights the usefulness of separating in-data, experimental factors and results from the code. By doing so he believes the following benefits may be achieved:

- *Familiarity*: no extensive training of using in-data is needed for end user
- *Ease of use*: no in-depth understanding of code is needed for end user
- *Presentation*: data is easily presentable
- *Further analysis*: data is easy available for further analysis
- *Version control*: easier to maintain a record of all experimental scenarios (Robinson S. , 2014)

3.1.6 **Verification and validation**

When the model is built it needs to be verified and validated. According to Banks et al. (2005) these are actually two different steps. Verification means making sure the computer model performs properly and validation is instead an iterative process of minimising discrepancies between the model and the actual system. Contrary to this split of the steps Robinson and Bhatia (1995) see validation as only one step after the model building and state that experimenting must not begin until both the modeller and the customer are satisfied that the model is valid. They suggest three common methods for validation:

- **Face validity:** on the surface the model appears to be reasonable
- **Comparison with the real system:** checking the model against data from the real system
- **Comparison with other models:** checking the simulation against data from other models, for example engineering calculations (Robinson & Bhatia, 1995)

The same three techniques listed above, and many more, are listed by Sargent (2008). The following bullets are examples of techniques from him:

- Animation,
- Degenerate tests

- Event validity
- Extreme condition tests
- Internal validity
- Multistage validation
- Operational graphics (Sargent, 2008)

On the topic of validation Law and McComas (1991) also discuss animation as a way to validate a model. They believe it is an especially good way to gain credibility for the model amongst stakeholders. They also suggest sensitivity analysis for understanding what aspects are the most important to validate since there is always a restriction on time and money in simulation projects. Another author discussing validation is Balci (2010) who lists what he thinks are the 20 golden rules for verification, validation, testing and certification. A few examples can be found in the following bullets with the corresponding number of the golden rule in a parenthesis after the rule:

- If a model is valid or not should not be seen as a binary variable where the accuracy is either perfect or totally imperfect (2)
- A model is built for a prescribed set of intended uses and its accuracy is judged with respect to those intended uses (3)
- Validation requires independence to prevent developer's bias (4)
- Validation is situation dependent (6)
- Complete testing is not possible for large and complex models (8)
- Validation activities should be planned and documented throughout the entire model development phase (10)
- Formulated problem accuracy greatly affects the acceptability and credibility of the results of the simulation model (14) (Balci, 2010)

3.1.7 **Experiment**

The experiment part is according to Banks et al. (2005) split into three steps; experimental design (1), production runs and analysis (2) and lastly a decision if more runs are needed (3). Similarly, Law and McComas (1991) split this part into designing experiments and making production runs. They also state that a complete decision cannot be made about what experiments to run until the first production runs have been made and results from these are obtained.

Robinson and Bhatia (1995) describe what they call phase three, experimentation, as containing both performing experiments and analysing the results. They also believe determining factors like warm-up, run-length and replications and selecting experiments belong to this phase.

3.1.8 **Documentation**

Many authors highlight the importance of documenting what is being done throughout the entire simulation project. Oscarsson and Urenda Moris (2002) state that not having documentation of the simulation may be costlier and cause more inconvenience than the opposite. Robinson (2014) motivates documentation for a number of reasons;

- To remember what has been done
- To be able to carry out further work at a later point
- To be able to re-use parts, or all, of the model
- To reach a better understanding for the model, the project and the results amongst less involved people
- To improve credibility of the model
- To facilitate verification and validation of the model (Robinson S. , 2014)

He also states three types of documentation that are required; *model documentation*, *project documentation* and *user documentation*, and lists useful forms of documentation for all three types, seen below in table 2.

Table 2: Examples of useful forms of documentation (Robinson S. , 2014)

Model documentation	Project documentation	User documentation
Conceptual model	Project specification	Project specification
Model assumptions	Minutes of meetings	Input data
Model simplifications	Verification and validation	Experimental factors
Model structure	Experimental scenario runs	Guide to run the model
Input data	Results	Results, accessing and interpreting
Experimental factors	Final report	
Format of results	Project review	
Names of components, variables etc.		
Comments and notes in code		
Visual display of model		

Similarly, Law and McComas (1991) also highlight the importance of an assumptions document, documentation of the program and making a summarising report of the results and conclusions. In addition to that, they mean that documenting the efforts made for verification and validation is of importance in order to enhance the credibility of the model, especially amongst those taking part of the result that were not involved with the details of the model building process. Banks (1998) states that making a clear and concise documentation of the experimentation analysis and of how the simulation model operates is beneficial if the model is to be modified or if the model should be used again later. It will give confidence to the simulation model and facilitate future decisions based on the analysis.

3.2 Simulation and mapping of complex manufacturing systems

Regarding simulation of complex manufacturing systems, the challenges of Fowler and Rose (2004) and the combination of reducing project lead time to increase acceptance, made by Massey and Wang (2007), have been mentioned already in section 1.1 Background. Adding to this topic of simulation of complex manufacturing systems, Balci (2011) provides a tutorial on how to succeed with large scale modelling and manufacturing projects. He presents a life cycle for modelling and simulation and how to organise processes, work products, quality assurance activities and project management activities required to develop, use, maintain and reuse a modelling and simulation application development. His model is similar to the one presented by Banks (1998) but includes more steps in the project process and defines an outcome from every step. In figure 11 below a schematic chart of his life cycle for modelling and simulation can be seen.

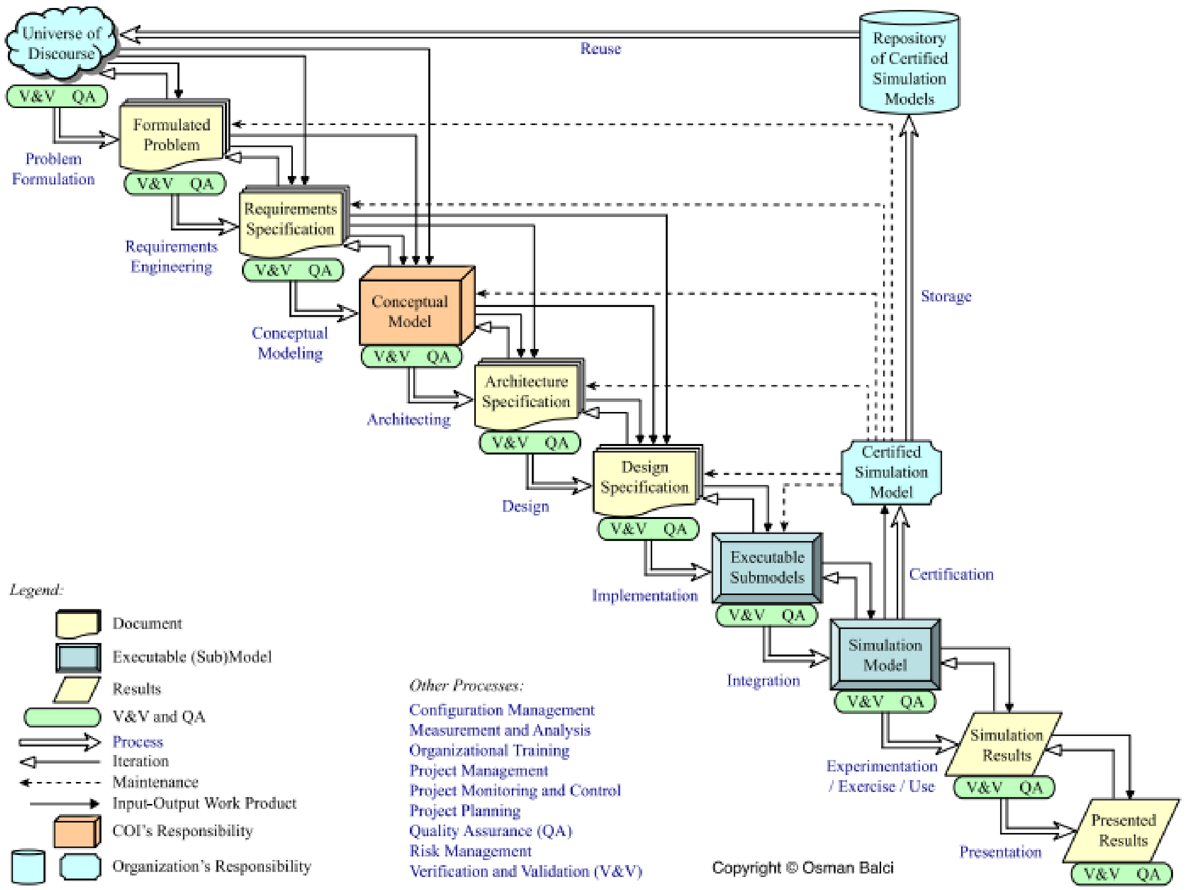


Figure 11: Life cycle for modelling and simulation

Another way of working with improving complex systems through visualisation of flow and data is value stream mapping (VSM). It is a well-known step within the lean methodology for fast analysis of a manufacturing system's production flow, from raw material to delivery (Rother & Shook, 1999). However, as mentioned in section 1.1 Background, when complexity of the system increases, VSM is not capable of handling the increased complexity. Authors, such as Solding and Gullander (2009) and Donatelli and Harris (2004), state that VSM no longer is enough since:

- Only the flow of one product or product type is analysed per VSM analysis
- The VSM gives only a snapshot of the situation on the shop floor at one specific moment
- The VSM map is a rough simplification of the real situation
- It is difficult to experiment with suggested new systems and layouts (Solding & Gullander, 2009), (Donatelli & Harris, 2004)

As an attempt to solve these issues Solding and Gullander (2009) combined VSM and DES and developed a tool called Simulation Based Value Stream Mapping. Their aim was to construct a dynamic value stream map that could represent the snapshot picture at any time during a time period. Their tool consists of two parts, a simulation model made in the simulation software AutoMod and a spreadsheet in MS Excel for adding data. The goal was to make the spreadsheet look as much like a standard VSM map as possible with the same type of icons and data visualized but with the possibility of showing all products, or flow groups, and not only one. Their VSM spreadsheet representation can be seen in figure 12 below.

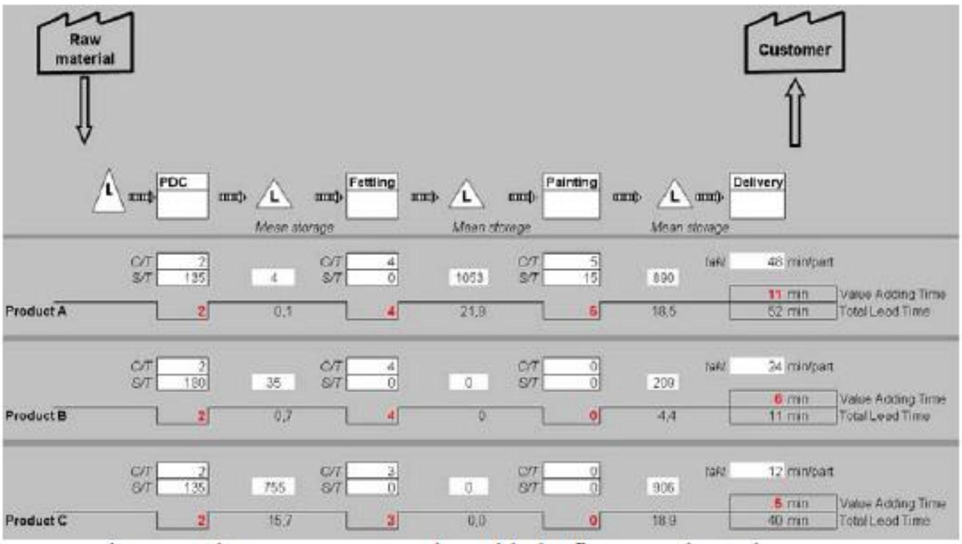


Figure 12: VSM representation in MS Excel spreadsheet (Solding & Gullander, 2009)

Similarly to Solding and Gullander (2009), Khaswaka and Irani (2001) also found the traditional VSM method inadequate for mapping more complex flows and instead developed an approach called Value Network Mapping (VNM). It is applicable in manufacturing facilities with multiple, merging flows of products with many components. Through integrating industrial engineering tools for material flow mapping with a software package for material flow analysis a visualisation and foundation for analysis of multiple flows can be created. Their proposed approach also helps to view a value stream at any, and all, levels of assembly. A VNM can be created following six steps; *form a product family*, *visualise the flow*, *collect data for the process boxes*, *merge similar routings*, *group similar routings into component families and draw the current state map.* In figure 13 below a level one VNM can be seen.

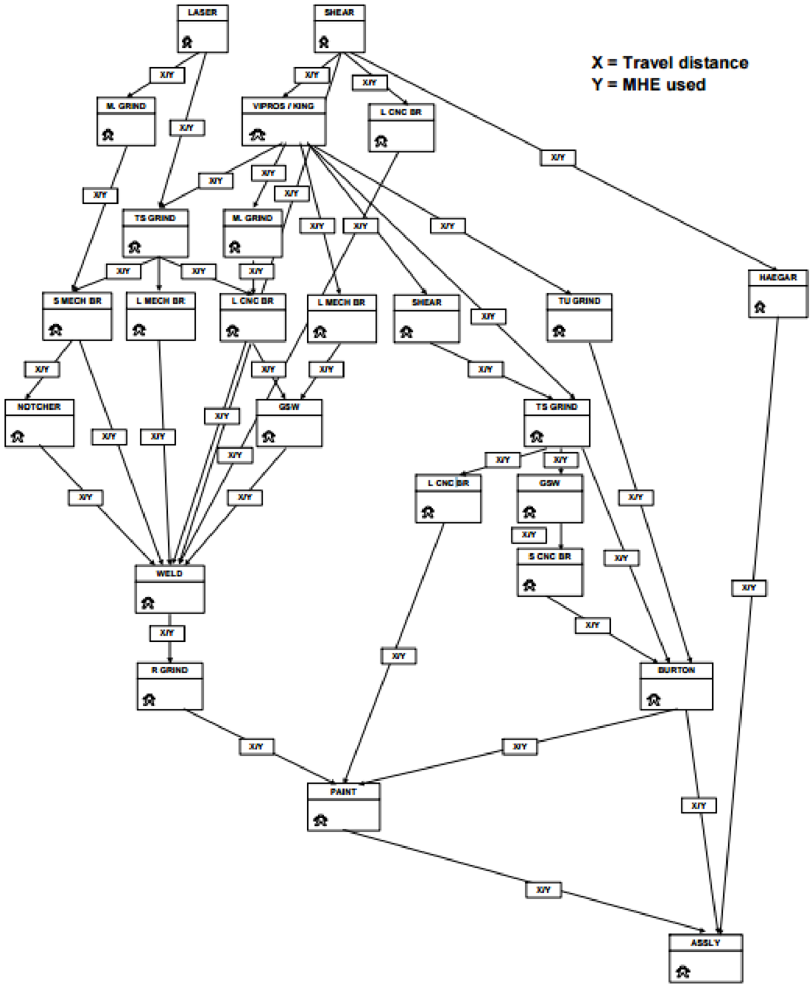


Figure 13: VNM at level one (Khaswaka & Irani, 2001)

4 Framework

In the following sections the initial framework, set up to facilitate the working process of simulation of complex manufacturing systems, will be described. First a brief summary will be given, then the three parts of the framework will be more in-depth defined.

4.1 Description of framework

In the literature study it was clear that many phases of the simulation project process have the potential and, in many cases, the need to be made faster. For example, Perera and Liyanage (2000) believe that it is inefficient data collection and lengthy model documentation, amongst others, that are preventing frequent deployment of simulation models. Likewise, Massey and Wang (2007) state that simulation is an extremely time-consuming process and discuss the importance of reducing the cycle time of data collection. In addition many authors, such as Balci (2010), highlight problem formulation and clear scope of the project as critical factors when working with simulation projects. Unrelated to a specific part of the DES project process, several authors, such as Oscarsson and Urenda Moris (2002) and Robinson (2014), highlight the necessity and benefits of documenting all that is done.

With all of this in regard, it was decided it was the first steps of the simulation project process: problem formulation, conceptual modelling, data collection, model translation/design specification and model building that had the most potential of facilitation. In addition, the later steps were deemed very specific for the type of project and hard to facilitate in a general manner. In published literature a lot of focus have been put on either one specific step or the entire project process. As an attempt to instead facilitate the working process with these first five steps the decision was made to set up a framework including all steps, and the transitions in-between, with the aim to reduce project lead time. Following what was discovered in the literature study the initial framework was set up with three example sheets in MS Excel, as frames, including;

1. System description
2. Visualisation and data collection
3. Model translation

Figure 14 below shows a schematic view of the initial framework and following sections will provide a more detailed description of each of the three parts.

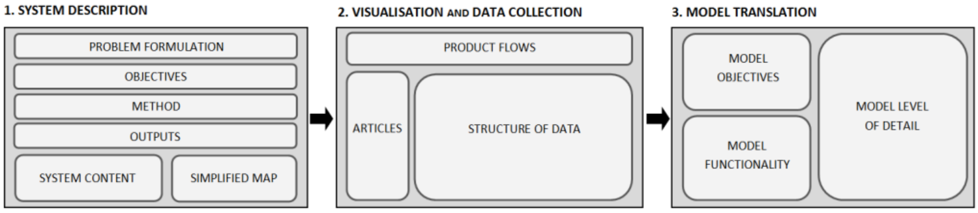


Figure 14: Schematic overview of initial framework

4.1.1 System description

The aim of the first part of the framework, *system description*, was to facilitate creating a structured summary of problem formulation, scope and objectives as well as partly the building of a conceptual model. This through providing a pre-set structure for describing the system. It was included in the framework as a result of several authors highlighting the importance of determining and agreeing on scope and conceptualising the system before starting the model building. Inspiration was taken from the bullet list of matters crucial to complete before a simulation project is started, published by Law and McComas (1991), and the framework for conceptual modelling, published by Robinson (2015). Four main headlines were chosen as a result:

- **Problem formulation:** to provide a clear description of the problem and give a background to the objectives
- **Project objectives:** to state the objectives of the project, i.e. what part of the problem situation the project aims to solve and how
- **Outputs:** to define what KPIs will be used to measure and compare results
- **Method:** to give a structured, summarised description of how the project objectives will be reached and with what potential experiments

A sixth headline, included under system description, was inspired by the level of detail and level of scope tables published by Robinson (2014):

- **System content:** to give a good view of what is to be included in the simulation model by listing products, operations, buffers etc. with a statement for every listed item if it is to be included or not in the simulation scope

In addition to this, since many authors seem to be using a more visualised way of conceptual modelling, such as Barra Montevechi et al (2010) and Cetinkaya, Verbraeck and Seck (2010), it is believed that a simplified map of the current state, layout or the current flow is beneficial to add under *system description*.

4.1.2 Visualisation and data collection

The aim of the second part of the framework, *visualisation and data collection*, was to facilitate data collection and, partly, conceptual modelling. This through providing visualisation of the system and a place to gather all needed data. Inspired by the Simulation Based Value Stream Mapping, published by Solding and Gullander (2009) an example sheet was created with features deemed helpful for getting an overview of the flow, collected data and missing data. In the top of the sheet the flow is to be represented from start to end with icons, similar to icons used in VSM (Rother & Shook, 1999), for buffers, processes, transports and decision points. The icons are then to be connected with lines to give an understanding of the different flows. In figure 15 below icons that were decided to be used to visualise the flow can be seen.

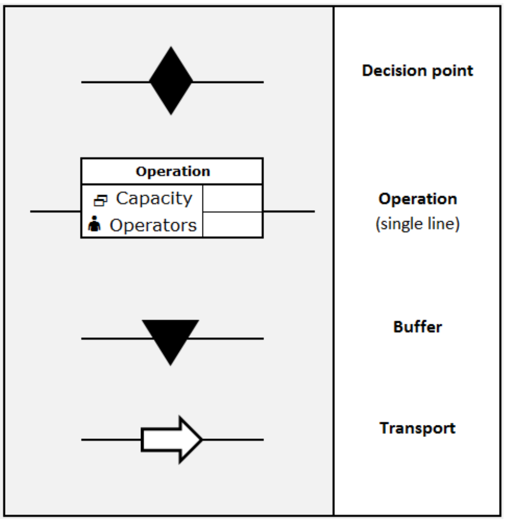


Figure 15: Icons used in the initial sheet for *visualisation and data collection*

Under each icon cells are to be prepared for storing data concerning buffer capacity, breakdowns, scrap, rework, transport times etc. All components are then to be listed to the left with all needed related data, for example name, ID, colours, dimensions etc. In case some of the needed data is missing the product name should be highlighted, preferably automatically. For each individual component and for each part of the flow a cell for data such as speed, transport time etc. should be present. In case the data is filled in, a colour should give an indication of that the specific component is going through that step in the flow. If not coloured, the component either does not go through this step of the flow, or data is missing. Several different colours could be used in this sheet to categorise data according to availability and/or accuracy. In figure 16 below a part of the initial example sheet can be seen.

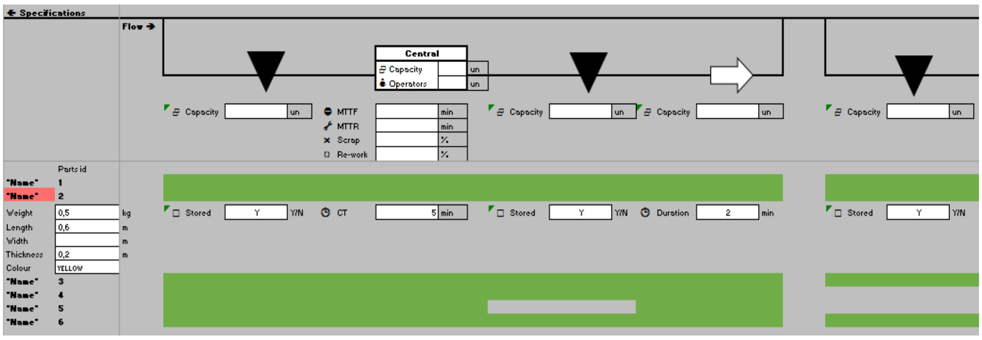


Figure 16: Example sheet for visualisation and data collection

The chosen structure of this sheet aimed to help A-B-C categorisation of data, published by Robinson and Bhatia (1995), and to help follow the detailed guidelines for data collection, published by Skoogh and Johansson (2008). It was also made to enable separation of data, stated beneficial by Robinson (2014), and give guidelines to what data could potentially be needed, taken from examples of needed data, published by both Hatami (1990) and Perera and Liyanage (2000). The process of structuring the sheet for the specific system, and adding all data, was meant to facilitate transforming data into information, inspired by the 5C's, published by Davenport and Prusak (1998).

4.1.3 Model translation

The aim of the third part of the framework, *model translation*, was to enable quicker model building. Since many authors have added a step in the DES project process before starting the model building, this sheet was meant to facilitate that specific step and, as a result, reduce time spent coding or building the model. The first part, *model objectives*, was inspired by the general project objectives, published by Robinson (2008), and the six bullets listed as crucial when planning a simulation study, published by Law and McComas (1991). It was meant to include the following headings:

- **Model flexibility:** to what extent the input data is changeable.
- **Run-speed:** whether code is written optimally for a faster model.
- **Visual display:** goals with animation.
- **Ease of use:** level of user friendliness of interface.
- **Model reuse:** information of whether model is built for reuse or not.

The second part of model translation was named *model functionality* and was inspired by the table showing an overview of underlying system characteristics, published by Seebacher, Winkler and Oberegger (2015). The purpose of this part was to, before starting to code, determine the functionality of the code to reduce the risk of having to re-do work. The following headings were meant to be included:

- **Run control**: information and data concerning the run control of the model:
  - Warm-up time
  - Run-time
  - Total length
  - Days per week
  - Shifts per day
  - Hours per shift
  - Cool down time to empty the system
  - Snap-length
- **Input data**: specification of needed input data and how it will be added in the model
- **Output data**: specification of output data and how it will be delivered by the model
- **Arrival pattern**: how loads/entities will be created in the model
- **Routing**: how routing decisions will be made in the model
- **Prioritisation**: how loads/entities will be prioritised in the routing decisions

The third, and last, part of *model translation* was meant to be used to set up a naming convention and structure the model building/coding. It was in some extent inspired by both the sample mapping table with corresponding ARENA templates, published by Perera and Liyanage (2000) and the template for level of detail, published by Robinson (2008). *Model level of detail* meant to include the following:

- **Loads/entities**, with description and related attributes
- **Resources**, with description and capacity
- **Processes/activities/functions**, with description and related resources
- **Queues**, with description, capacity and dwell time
- **Source files**, with related processes
- **Sub models**, with related source files

All above headings could of course be changed depending on the chosen software used for model building.

5 Empirics

In the following sections the case and the application of the framework on the case will be described. First a summary of the case, its purpose, objectives, scope and methodology will be given. Then, the usage of the initial framework on the case will be described, followed by what further add-ons were made and how the framework was additionally used.

5.1 Case description

The Site, in focus of this case study, currently consists of three different factories, Factory A, B and C, all producing in a shared production flow. Components starts in Factory A to go through Process 1, are sent to an external supplier and when received back go through Process 2, and sometimes Process 3. They are then divided between the three different factories since all three factories perform Process 4, 5 and 6 but for different types of components. However, all components produced fully in Factory A has to be packed, process 6, once in Factory A and then a second time together with other components in Factory B or C. To add to the complexity Factory A has functionality needed for components produced by Factory B and C so components can be sent back and forth between the three factories quite a lot. Apart from Process 1, 2, 3, 4, 5 and 6 there are some additional processes, that are minor but still has to be regarded, that adds to the complexity. In figure 17 below a simplified schematic view of the flow can be seen.

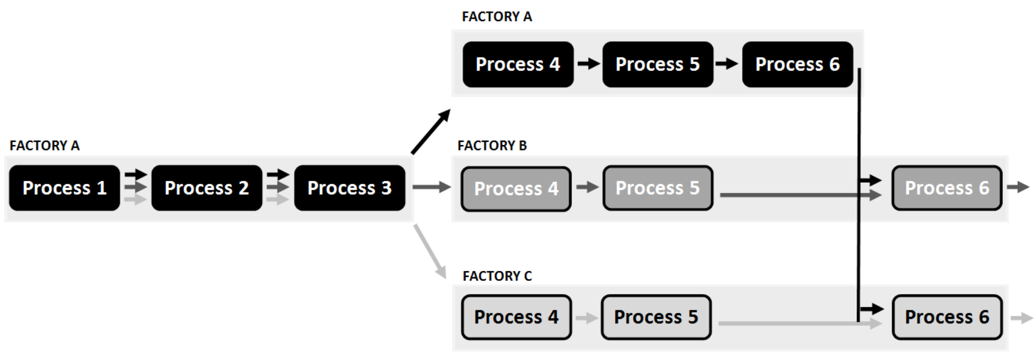


Figure 17: Simplified schematic view of the flow at The Site

The Site currently produces approximately 280 different articles, consisting of more than 800 components. Available capacity today consists of:

- Process 1: 1 machine in Factory A
- Process 2: 3 lines in Factory A
- Process 3: 2 lines in Factory A
- Process 4: 2 lines in Factory A, 3 in Factory B and 3 in Factory C
- Process 5: 1 line in Factory A, 2 in Factory B and 3 in Factory C
- Process 6: 2 lines in Factory A, 3 in Factory B and 3 in Factory C
- Additional processes: machines and lines in Factory A, B and C

With aims to improve this complex flow, The Site started a few months ago The Flow Project, a project focused on making transitions between Process 1, 2 and 3 run smoother. However, when looking at ways to solve similar issues in Factory A, B and C with Process 4, 5 and 6 changes and investments seemed too risky for The Site to dare doing without further analysis. To reduce the risk simulation was added to the picture.

5.1.1 Case purpose and objective

The purpose of the simulation project was to see if Process 4, 5 and 6 could be excluded from Factory A and done solely in Factory B and C. This because these processes in Factory A are located in a separate building from the rest of the factory, leading to high costs and time wasted for internal transports. Additionally, The Site hopes to ramp up production in the future and therefore wants to have free capacity in their lines to still have the ability to do this. The main objective of the case was hence to investigate utilisation of lines and buffer levels in Factory B and C.

To solve arising challenges when removing Process 4, 5 and 6 from Factory A, a new future state, compatible with current ongoing changes of The Flow Project, was set up. This new future state included a new, expanded, department for Process 6 with modernisation of lines and the option to move lines for Process 4 and 5 from Factory A if found needed in the simulation. With this new future state, the experimentation part of the simulation project aimed to answer the following questions:

1. Is capacity in current lines in Factory B and C for Process 4 and 5 enough to run a future ramped up production?
2. Can re-routing of components in existing lines in Factory B and C for Process 4 and 5 even out utilisations between lines?
3. Is moving a line from Factory A to Factory B, or investing in a new line, for Process 4 needed?

4. Is moving a line from Factory A to Factory B, or investing in a new line, for Process 5 needed?

5.1.2 Scope and method of the case

The simulation model built for the case study included only Process 4, 5, 6 and some of the additional smaller processes since it was only these lines that would be affected if removing Process 4, 5 and 6 in Factory A. It ran components from 12 weeks of production plans from 2015, hence limiting the number of different components in the model to approximately 500. The main output variables to compare different experiments with was free capacity of lines and buffer levels between lines.

The case study followed the project process previously defined and was carried out over 16 weeks. Four factory visits were made during the project but most time was spent at the head office of The Company. The simulation model was built in the software AutoMod since this is currently in use at The Company. All data was collected in the Excel structure set up as a part of the framework and then added to AutoMod by a pre-written macro also currently in use at The Company.

5.2 Initial framework applied on the case study

Once the initial framework was set up the case study was started. All three parts of the framework, *system description* (1), *visualisation and data collection* (2) and *model translation* (3) was tried on the case and changes were made when deemed needed. This section will cover how the three parts were used and the changes made to them.

5.2.1 System Description

The system description was filled in at an early stage during the case study. It included:

- *problem situation,*
- *project objectives,*
- *model outputs,*
- *method* and
- *system content.*

Problem situation could be summarised as "a lot of internal transport and complex flow of components". The *project objectives* then stated what future concept was to be analysed through simulation and that an analysis of re-routings and investments needed for this future concept would be investigated. *Model output* was specified to utilisation of lines, measured in percent and buffer levels, measured in number of baseboards. The *method* section briefly summarised how the objectives were to be achieved, for example through running historic production plans, try different re-routings etc. The *system content* gave a description of the number of articles produced by The Site and how many of them were to be included in the simulation model. It also listed all operations/lines currently in use at The Site, as well as buffers, and if these were to be included or not. Finally, a short summary of the production flow was given, if these different steps were to be included or not and what exceptions existed to the normal flow. In figure 18 below a part of *system content* in the *system description* applied on the case can be seen:

System content	
Products	
282 articles in total according to master plans from 2015 and 2016	
Included in model:	168 different components in Factory B
	327 different components in Factory C
Operations	
Process 1	Exclude
Process 2	Exclude
Process 3	Exclude
Process 4 Factory A	Exclude
Process 5 Factory A	Exclude
Additional process	Include
Process 6 Factory A	Include
Process 4 Factory B	Include
Additional process	Include
Process 5 Factory B	Include
Process 6 Factory B	Include
Process 4 Factory C	Include
Process 5 Factory C	Include
Process 6 Factory C	Include
RMW	Exclude
FGW	Exclude
Buffers	
Buffer 1	Exclude
Buffer 2	Exclude
Buffer 3	Include
Buffer 4	Include
Buffer 5	Include
Buffer 6	Include
Buffer 7	Include
Buffer 8	Exclude

Figure 18: The beginning of system content listed under system description applied on the case

In addition to these already defined headings a new one was created, *experiments*. Initially this was included under *method* but in the case study a lot of different experiments were run and by using it as a standalone section descriptions were made more detailed.

5.2.2 Visualisation and data collection

The *visualisation and data collection* sheet was filled in and further developed as soon as the data collection started. As in the initial sheet, the top visualises the flow of operations with lines indicating different flows for different components. Figure 19 below shows the icons used to visualise the flow.

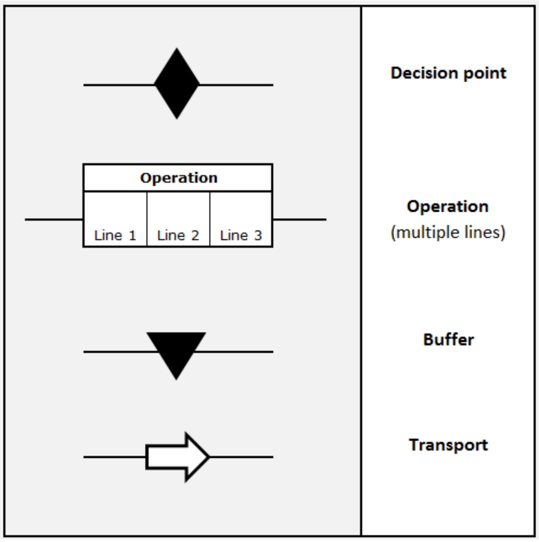


Figure 19: Icons used in the *visualisation and data collection* sheet

The icon representing operations was changed from representing one to representing multiple lines, under which speed data was filled in for each component. Under the icon representing decision points it was stated, for each component, which of the following lines was preferred. Data concerning scrap and re-work was not relevant for the case and was therefore not used. Buffer capacities was also removed since the model did not take these into account. Stops, both planned and unplanned, was initially thought to be in discussed sheet but was due to time restrictions simply added to the code.

All components were listed to the left with belonging article ID, component ID, parts per article, baseboard size, parts per baseboard, colour etc. Under the visualisation of the flow, white-marked cells for each operation and each component was added, representing which data was needed for what component. If a cell was marked grey no data was needed in this cell. The cells could be coloured in other colours according to a colour coding put in the top of the sheet, indicating possible routings, assumptions, exceptions in production, missing data, data accuracy etc. Figure 20 below shows a part of the *visualisation and data collection* sheet.

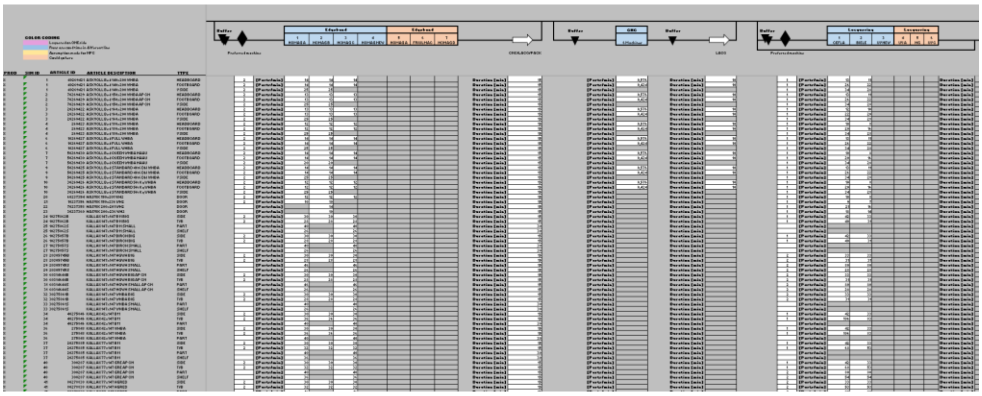


Figure 20: A part of the *visualisation and data collection* sheet

5.2.3 Model Translation

The first part of model translation, *model objectives* was filled in at an early stage of the model building. Decisions made for *model objectives* can be seen in figure 21 below.

MODEL OBJECTIVES	
Flexibility:	Only runs for set master plans
Run-speed:	Not prioritised
Visual display:	Simple animation
Ease-of-use:	User friendly GUI in extent possible
Model/component reuse:	Not prioritised

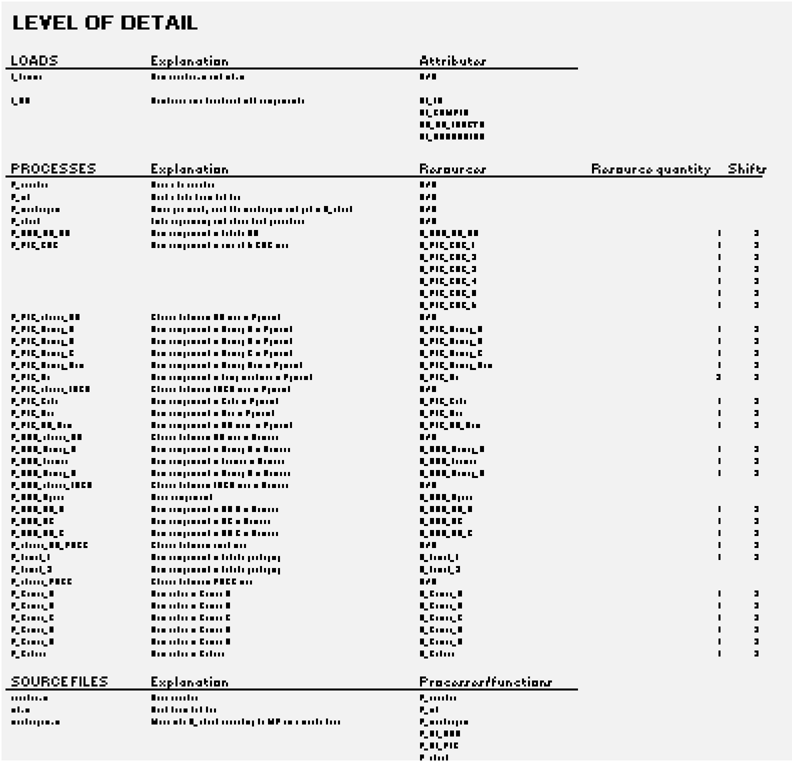
Figure 21: Model objectives as a part of the model translation applied on the case

After model objectives had been specified the functionality of the model was to be determined. Decisions were made in relation to what had already been stated in system description and some of them can be seen in figure 22 below.

MODEL FUNCTIONALITY		
Run-time		
Warm-up time:	5 days	
Run-time:	12 weeks	
Snap length:	8 hours	
Additional end time:	1 week	
Specifications:	5 days per week	
	24 hours per day	
	3 shifts per day	
	8 hours per shift	
Input data		
NPC Data	In "Summary" sheet	
	Per component and line	
	Unit: Parts/min	
MTTF, MTTR	According to fitted distributions	
	Added in code	
	Not considered for:	Process 6
		Additional processes
Set-up	Matrices for lines in Process 4	
	Matrices for colours in Process 5	
	All components have set-up IDs	
Allowances	Added in code	
Output data		
Utilisation	Shown states:	
	Breakdown	
	Planned stop	
	Set up	
	Idle	
Buffer lengths	Meters of utilised buffers, average and max	
Arrival pattern/Master plan		
Master plan read on a weekly basis, on Wednesday		
All components are put in Q_start on baseboards		

Figure 22: A part of the model functionality listed for the case

The heading level of detail was also filled in before the model building was started and then constantly updated during the coding. First, the needed loads were added, what they would represent and their belonging attributes. This followed by defining what processes that were to be included, with an explanation, and what resources was needed for every process, with resource quantities and number of working shifts. In the same way, the queues were defined with an explanation, stated capacity and related dwell time. Finally, the source file names were determined with an explanation and what processes would belong to what source file. A naming convention was set up dependent on factory, process and line. Figure 23 below shows a part of *level of detail* listed for the case.


LEVEL OF DETAIL

LOADS | Explanation | Attributes

PROCESSES | Explanation | Resources | Resource quantity | Shifts

SOURCE FILES | Explanation | Processes/functions

Figure 23: A part of *model translation* where the beginning of *level of detail* can be seen

5.3 Add-ons to the initial framework

During the case study the initial framework was used as a base and was constantly changed in order to facilitate and enable a faster process of the first steps of the simulation project methodology. Although the initial framework was of good use the following additional parts were deemed needed and added to the framework:

- Set-ups
- Assumptions
- Missing data
- Three-dimensional variable

Depending on type of manufacturing system set-ups can be handled quite differently. For this reason, set ups were not added to the initial framework but it was instead decided to wait with adding these until the structure of set ups at The Site were known. Because The Site deals with set up data as set up matrices based on components or colours for almost every line in the three factories, it was soon realised that to gather all set up data in the *visualisation and data collection* sheet was not possible. Instead an additional sheet was created to gather all setup matrices. However, this lead to the problem of components being unconnected to their related set up time. To solve this a system was created where each component in the *visualisation and data collection* sheet received a set-up ID corresponding to where this specific component belonged in the set up matrices. This enabled reading set up IDs as integers when reading input data and facilitated the connection of set ups IDs and set up matrices when coding.

The second add-on to the framework was an *assumption sheet* meant to be frequently sent back and forward to the stakeholders to confirm, correct and comment on all assumptions made in the simulation model. This was used during the study but was not decided to be a part of the framework until later in to the project. An example of the assumptions document used, and later added to the framework, can be seen in figure 24 below.

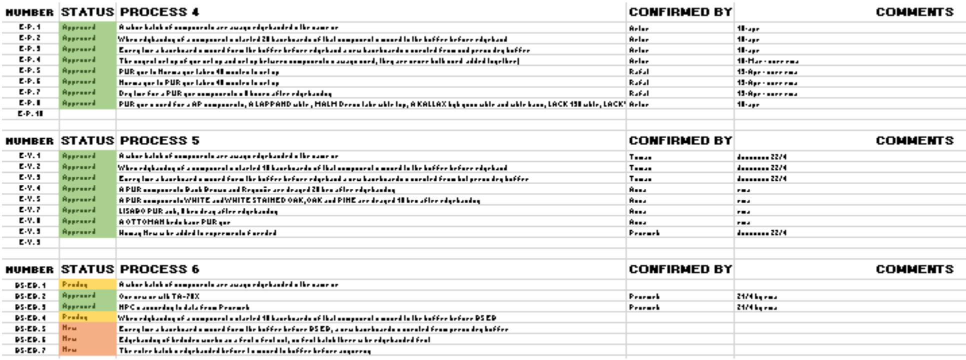


Figure 24: A part of the assumptions sheet

As a part of the initial framework the sheet for *visualisation and data collection* was meant to clearly highlight what data that was missing. However, as the complexity grew, so did the amount of questions regarding the system and already collected data. The initial sheet did not seem to facilitate the overview of what was still missing *and* related questions. Therefore, a *missing data* sheet was created where all missing data and questions regarding understanding of the system were added. This sheet was meant to be used to keep track of what data or information had been requested for, to whom and when it was sent and what needed to be confirmed or corrected. It was constantly updated during the project.

The fourth and last add-on to the framework came from the wish to store all data added to the *visualisation and data collection* sheet in one variable. However, doing so seemed troublesome since components were defined by both an article ID and a component ID. To solve this problem, a three-dimensional variable was created where article IDs would be on one axis, component IDs on a second axis and different types of data on the third. This way of storing data made coding easy since loads only needed article ID and component ID as attributes to be able to access almost all data related to that specific component. To be able to easily access the different types of data, variables were created for every type with the respective number of where that type of data would be found in the three-dimensional variable, see figure 25 below for an example of code. In addition, a visualisation of the variable was made to enable understanding and overview while coding, see figure 26 below.

```
begin P_Process4_Line1 arriving
    move into Q_in_Process4_Line1

    //Set up
    wait for VI_Process4_Line1_SetUp(PreviousSetUp, VR_Summary(AI_ID, AI_COMPID, SET_UP_PROCESS4) min

    //Produce; ordersize divided by speed
    wait for AI_ORDERSIZE/VR_Summary(AI_ID, AI_COMPID, SPEED_PROCESS4_LINE1) min

    //Dwell time
    wait for VR_Summary(AI_ID, AI_COMPID, DWELL_TIME) min

    send to P_chooseLine_Process5
end
```

Figure 25: Simplified example of coding using the three-dimensional variable, in code called VR_Summary

5.4 Additional usages

In addition to the previously mentioned initial parts and add-ons to the framework the file keeping the entire framework applied on the case also contained output sheets. These were created through letting the model write, to a text file, utilisation of all concerned lines, buffer levels of all included buffers and what components had gone to what line at every snap, and then reading these text files to MS Excel. Graphs were then created from this data to visualise utilisation and buffer levels and lists to enable following of components through the entire system. These graphs, and lists, were then used for validation but also for experimenting through saving copies of the framework before changing the input data and run the next experiment.

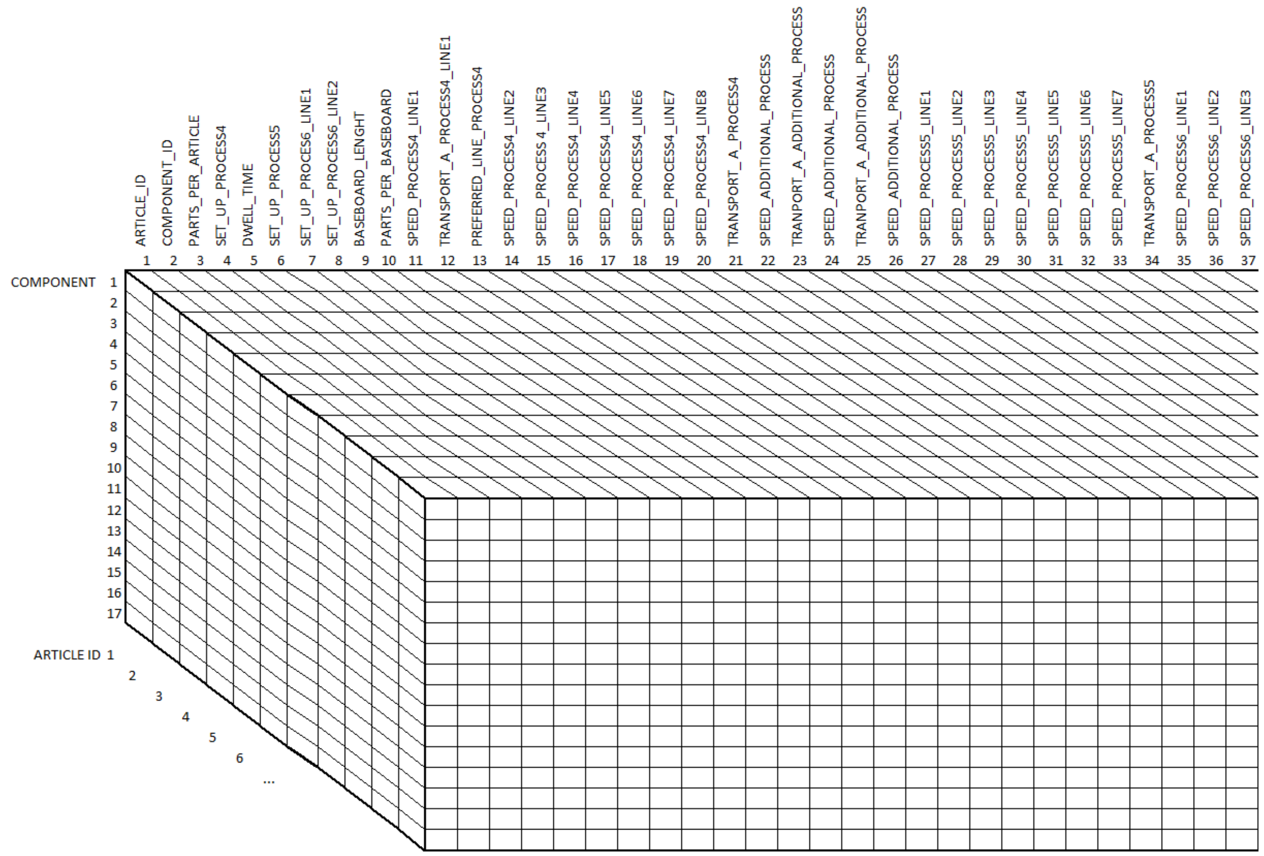


Figure 26: Visualisation of three-dimensional variable

6 Analysis

The following sections will cover the analysis of this study. First, the purpose and methodology will be discussed. Then, the set up and structure of the final framework will be analysed followed by advantages and disadvantages. Finally, suggestions for further development will be given.

6.1 Purpose

The purpose of this study was to get a deeper understanding of the working process when working with DES projects. It was also to find areas within this process that had potential to be facilitated and then make an attempt to find ways to facilitate these, in a general manner, with the aim to reduce project lead time. This purpose was a result of wanting to answer the problem formulation, which was based upon the statement that although complexity of manufacturing systems is increasing there is a need to reduce project lead time for DES projects. However, project lead time is a wide concept depending on several different factors. Measuring project lead time is possible for a single project but to compare this measurement with other projects and define the reason for lead times being either similar or very different is a lot more difficult. Aspects such as experience, learnings from related projects, complexity of project, upcoming challenges within the project, tools available etc. affects the lead times, leading to similarities or dissimilarities. To be able to compare two different project lead times all these aspects must be the same for the two projects, a situation very difficult to establish.

Included in the purpose of this study was an aim to reduce project lead time. However, following above discussed problems with lead time, no evaluation of if the created framework actually shortens the project lead time of DES projects can be made. The different parts of the framework were created and structured as an attempt to facilitate the project process of DES projects and hoped to also reduce project lead time. Though, whenever stated to actually do so, this is only based upon the authors' beliefs and such statements are to be regarded as quite biased.

6.2 Methodology

The chosen methodology for this project included a literature study and a single case study. The literature study was focused around the first two bullets of the purpose; understanding the DES project process and what underlying steps have the potential to be facilitated. It also aimed to find inspiration for the third bullet. This resulted in a framework which was then tried and evaluated in a case study.

The literature study was made in a quite unstructured manner and chosen key words to search for were very wide-spread. A more structured study focusing on more specific subjects could potentially have generated different findings, leading to a different structure of the three parts within the framework. Given the time frame of this project the literature study was also kept quite short and a prolonged study within the same or similar subjects would probably have led to more in-depth conclusions.

After having set up the framework, it was tried upon an embedded, single case study. No other evaluation of the framework was made, resulting in a structure probably very affected by the nature of the case. The iterative process defined also contributed to this, making the case study not a test of the framework, but a way to develop it further from the initial set up. Other aspects from the case study that probably impacted the final result is the geographical situation and the software restriction. Since the study was performed at a site in Eastern Europe and most time was spent at the central office in south Sweden a lot of struggle was experienced with communication and misunderstandings. Applying the framework on a similar case but with the possibility to work at the site would most likely have generated a somewhat different outcome. The second aspect, restriction regarding software, is important to mention since many authors from the literature study have stated the importance of choosing an appropriate software. However, in the case study AutoMod was already used as the simulation software at The Company and choosing software has for that reason not been incorporated at all in the framework. In addition, if the restriction had been set to another software this would probably also have affected the final outcome.

6.3 Set up of framework

Following the literature study it was concluded that the first five steps of the DES project process had the biggest potentials of being facilitated. Many publications were found regarding facilitation of these steps but most of them gave either a very in-depth description of a specific step, or a general study covering all steps within the project process. Following this discovery, an urge to create a framework contributing to a combined facilitation of all these five steps, in an easy-to-use, structured manner, arose. The decision of choosing these first steps were also based upon the purpose of contributing to a general facilitation of simulation projects. The last steps of the DES project process, verification and validation, experimenting, analysing and presenting results, were deemed very specific, differently structured depending on the type of project and hard to generally facilitate. They were therefore excluded from the framework.

A second focus of setting up the initial framework was to keep it simple. Many suggestions found when researching literature on how to work with complex manufacturing systems included a lot of different software and integration of methods. The end result coming from methods like these are probably superior to the framework set up in this study, but learning and using such methods requires time, engagement and technical competence not always present. Given this, many of these suggestions were excluded from the study as they were considered too complex.

The final version of the framework consisted of a three-part method, including all five above mentioned steps, more thoroughly discussed below. Figure 27 below shows an updated version of the schematic view of the framework, previously shown in section 4 Framework.

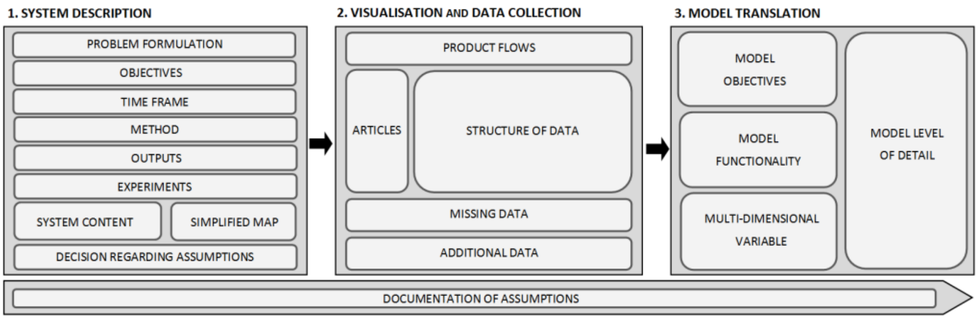


Figure 27: Schematic overview of the final framework

6.3.1 System description

The first part, *system description*, was decided to be included in the framework to help defining a clear problem formulation and scope of the project. This since it was found in literature being critical for reducing the lead time of simulation projects. However, in the case study, this part was filled in by the modellers themselves and done too early into the project, before a clear understanding of the system and the flows existed. When the modellers' understanding of the system, and its complexity, grew it was realised that the initial concept to be tried as a future state was not doable in practice. As a result, both scope and model content had to be changed halfway through the project. A big reason for this late change of scope was lack of involvement and understanding of simulation from stakeholders and project team members at The Site. It is believed that spending more time together with stakeholders on *System description* in the beginning of the project is of great importance to reach the goal of reducing the lead time. In this case, *system description* ended up instead being updated in the end of the project, mainly to enable a hand over, and was not of the use expected.

Looking back on the case study, one matter not handled very well was setting a time frame for the case together with stakeholders. It is believed that this should be added to the *system description* and deadlines for each part of the framework, as well as experimenting and delivering results, should be decided together at the beginning of the project. This would engage both parts and reduce the risk of modellers or stakeholders having different expectations on deliveries.

Documenting assumptions was frequently mentioned as crucial in the literature study. Documentation of and communication about assumptions could be dealt with in numerous different ways and it is believed important to discuss how to deal with assumptions already from the start. In the case study this was not done thoroughly enough and assumptions ended up being a separate document created later into the project, unconnected to the framework. The idea was that a separate file would enable communication about assumptions by sending the assumptions document back and forth to The Site for approval and signatures as new assumptions were added to the model. However, most assumptions made during the case study was verbally approved during meetings and the assumptions document was updated by the modellers. Having it separated from the file containing the rest of the framework made this updating more time consuming than needed and it was added later in the project. Since how to deal with assumptions is believed something that should be decided at the beginning of the project it is in the authors' beliefs that this decision belong under system description. However, since assumptions should be continuously updated throughout the project it does not belong to any specific step in the framework but should instead be documented according to previously mentioned decision.

6.3.2 Visualisation and data collection

The second part, *visualisation and data collection*, was decided to be included in the framework to enable understanding of the different flows and to give a clear view of what data is needed and missing. The idea is that this second part should be started as soon as possible, but not until all parts of *system description* is agreed upon. This since starting it earlier will increase the risk of having to re-do work if the scope, objectives, method etc. were to be changed. However, it is still to be expected that matters, for example unavailable data, might come up during the data collection which could force changes of the decisions made in *system description*.

The initial sheet was created before the authors had knowledge about The Site, and was therefore expected to turn out differently when applying it on the case. After applying it on the case, the part of understanding the flow was successful and one could easily follow a specific component from its first to its last process. The sheet also facilitated an overview of what data was needed and enabled understanding of the system by assisting quick comparison of the different operations by storing data next to each other. However, the sheer size of the final sheet made the overview of cells marked as "missing data" more challenging than initially expected. In addition to that, it was discovered that data were seldom accurate the first time received and documentation of information and questions regarding this was needed. This was in no way facilitated by the *visualisation and data collection* sheet and an additional sheet was therefore created, called *missing data*. This was used to keep track of what data that had been requested for, to whom and when it was sent and also what data that needed to be confirmed, corrected or updated. The authors felt this sheet was of great use and believe it would have been very beneficial having this as a part of *visualisation and data collection* from start. It is believed useful to avoid asking the same questions over and over again, not receive answers asked for or requesting the same data multiple times.

As mentioned before, the *setup sheet* was not added to the initial framework from start. This, since setups can be handled quite differently depending on the type of manufacturing system and it is in the authors' beliefs it would be hard to find a general approach for structuring setup times. However, once it was clear how set ups were handled at The Site they were added to the framework. Given the density of the set ups, adding them was quite a challenging procedure and linking components to their related set up in a way that would also enable easy coding took some time. Keeping in mind how different set ups can be in different manufacturing systems, the way set ups were dealt with in this project might not always be applicable. Although, it is still recommended to keep set ups, as well as other types of data that might not fit in the visualisation structure, and how to deal with them, in mind when starting this second part of the framework.

6.3.3 **Model translation**

The third part, *model translation*, was decided to be included in the framework to give guidance when specifying the design of the model and to facilitate the actual model building. It should not be started until it is clear all data and all information stored in the *visualisation and data collection* sheet is available, i.e. category A data. This to avoid structuring a model that then has to be re-structured at a later stage because it was discovered needed data was not available or information about the system was misinterpreted or altered.

During the case study, *model translation* was considered of great use and is believed to really have shortened the time spent coding. Thoroughly going through, discuss and make decisions regarding the first two parts, *model objectives* and *model functionality*, resulted in many important aspects of modelling being set already before building started. This reduced the risk of encountering unexpected questions further on and having to go back, re-discuss and change decisions. The last part of model translation, *model level of detail,* was also found particularly helpful. Although it had to be initially set up and frequently updated, which was of course time consuming, it is believed to have reduced the total time spent building the model and de-bugging the code. It is also deemed very useful when handing over the project and/or if the model is to be re-used at a later stage.

During the case study a visual representation of a three dimensional variable used to store all data from the *visualisation and data collection* sheet was created. The three dimensional variable was needed because data was dependant on both an article ID and a component ID for every entity in the model. To store data in such way enabled easier coding and shorter code but was sometimes hard to overview, which is why the visual representation was well needed. If working with systems where components also contains sub-components it is believed that more dimensions can be added to still enable storing of all data in one place. Creating a visual representation of a multi-dimensional variable consisting of more than three dimensions is of course difficult. However, during the case study it was found that it was mainly the different types of data stored and what variable was needed to get that data that was of the most use. For example, finding values under DWELL_TIME or SPEED_PROCESS4_LINE1. For multi-dimensional variables of higher dimensions than three this could still be facilitated by a visualisation of only this axis. Although found useful during the case study of this project, this way of storing data might not always be applicable depending on the type of software chosen for model building or the type of manufacturing system.

6.4 Advantages and disadvantages

When looking at previous published literature about the five steps chosen to be in focus of this study they are, as mentioned before, focused on either the entire project process of DES or focused quite specifically on one area. The problem with this is that the broader ones, focusing on the entire project process, do not give specific guidelines on how to work with the different steps. The more specific ones, however, explain in-depth how to work with a certain step but do not connect this work with the steps before and after. Instead, it is up to the reader to choose one published method for defining scope, one for conceptual modelling, one for data collection etc. It is to be expected that time and effort will be lost in the transitions between the steps if the methods are unconnected, extending the project lead time. The framework published in this study was an attempt to do something in between. It aimed to facilitate the first five, quite general, steps of the DES project process and to do this more specifically than previously published general literature. At the same time, it aimed to not be too specific on one area but to instead tie the five steps together through one framework, also facilitating the transitions between the steps.

One other matter found advantageous with the framework is that it was created based upon a real life case. In literature a lot of case studies can be found but those articles most often only concern the study and its results and quite seldom make an attempt to draw conclusions about how to work with DES in a more general manner. Articles that are actually doing so seem to be very theoretical and based upon previously published literature instead of actual real life work. This framework is a result of trying to do both, initially based upon literature and then tried in a case study. It is, however, worth to mention that a lot of previously published literature is written by people with a lot of experience within the field. Publications about more general facilitation of the DES project process are most likely also based upon the publishers' previous experience and not only on previously published literature. This experience is something lacking when creating the framework in this study.

One issue mentioned frequently in this report and very important in the case study is handover. Since the project done within the case is very likely to be continued or at least built upon, but not by the authors of this report, it was very important that the framework also facilitated a handover of the entire project. All documentation, naming conventions, commenting of code etc. were done with this future hand over in mind and have probably affected the final framework. If there is no future handover in mind documentation and other aspects of the project can probably be carried out in a less extensive manner.

When reaching the model building stage of the case study the initial idea was that the framework would have served its purpose since the upcoming stages were not to be facilitated. It was, however, found very useful to continue using the same Excel file for saving all outputs of the model and display the results. Not only did this enable experimenting but it also made verification and validation, as well as documentation of both these steps, a lot easier. Once the experimentation stage was started copies of the framework were made where input data could be changed for every experiment before the model was run. All results from the different experiments were, through this way of working, saved and easily accessible. When it was time to build a report for the case study coping the different graphs into the report was quickly done since all results were on the same format, allowing the creation of a neat looking report.

The main disadvantage of the structure of the framework created is that it requires some MS Excel knowledge and some experience in writing Visual Basic code to allowing writing to and reading from text files. In the case study this was not a concern because macros were already written and in place to support this needed functionality. However, coding such macro from scratch is assumed to be quite time consuming and might extend the project lead time for the first projects it is used for.

6.5 Suggestions for further development

Although updating the framework frequently during the case study, some problems occurred that are believed able to be facilitated by the framework, but no way of doing so was incorporated during the study. The main thing partly missing, that could be future developed, is keeping track of data accuracy. When looking at data collection during the case study all eleven bullets listed by Balci, Ormsby, Carr III and Saadi (2000) was experienced as troublesome and, at the end of the study, data is still believed somewhat inaccurate. Incorporation of a system to mark or keep track of data quality is believed to have the potential to greatly improve the framework.

It is also believed that the approach chosen to visualise flow might not be sufficient for flows where components can go through lines in multiple different orders. In the case study most components produced, although using different lines and/or skipping lines, went through production in approximately the same order. Exceptions to this were so few that special routing could be covered by colour coding. It is believed that in manufacturing systems where this is not the case further work is needed on the *visualisation and data collection sheet*.

Following the limitations of the methodology, already stated, it is also believed that further studying and testing of the framework could additionally improve it. There are many subjects from the literature study that could have been studied a lot more in-depth. For example, when it comes to data collection a lot have been published about integration of systems and automated data collection, or plug and play inoperability, which is considered to have big potential to really improve the use of the *visualisation and data collection* sheet. Another example is to, within a company, work further with the *model translation* part to set up standards used for all simulation projects. To test the framework on more case studies, with a wider spread of manufacturing system types, also have the potential to improve it and make it more general as it is believed to have been very influenced by the nature of the case study.

7 Conclusions

The purpose of this study was to get an understanding of how the process of working with DES projects could be facilitated. It was discovered in the literature study that the main areas that had the biggest potentials were the first five steps of the project process; defining problem formulation and scope, conceptualising the system, collecting data, specifying model design and building model. An additional discovery was that throughout all five steps, documentation was constantly mentioned.

Following the literature study it was concluded that a lot of work have been done within all five steps, but most work published is either very in-depth within a specific step or very general covering the entire project process. An attempt was made to set up a general framework aimed at assisting documentation and enabling a combined facilitation of above mentioned five steps. As a result, a three-part method, made in MS Excel, is concluded to have the potential to facilitate all five steps. Figure 28 below shows a schematic view of the final framework.

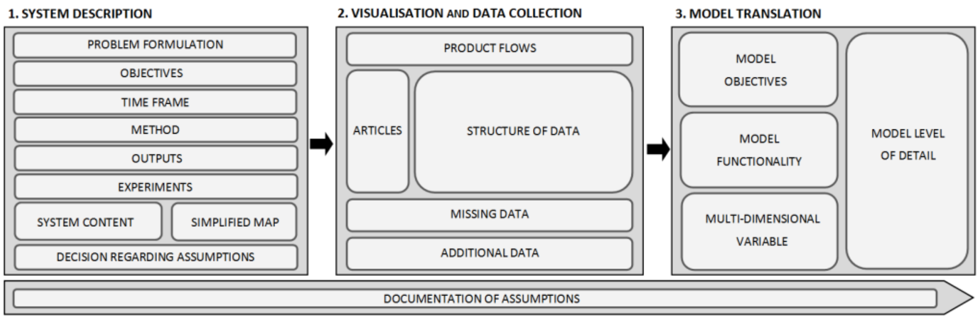


Figure 28: Schematic overview of the final framework

The first part of the framework, *system description*, should contain problem formulation, project objectives, project time plan, output KPI’s, proposed method, summary of experiments and system content. All these areas should be defined together with all stakeholders of the project before proceeding with the project. In addition, a decision should be made together with stakeholders regarding documentation and handling of assumptions. Assumptions should then be continuously updated throughout the project, and signed off, according to this made decision.

Once completed, the second part of the framework, *visualisation and data collection*, should be set up to obtain an understanding of the system and its flows and to define needed data. At this point it should also be discussed and decided how to handle data that might not fit into this structure, such as set ups. The *visualisation and data collection* sheet should then be used to gather all data in one place and keep track of missing data and data accuracy. In addition, it is recommended to document information regarding missing data and how it is to be gathered.

The third, and final part, *model translation*, should be started once all needed data is categorised as A data, i.e. available. Model objectives and model functionality should be discussed, decided and documented before the actual model building starts. Also before starting coding, the model content should be discussed and defined and a related naming convention, to be used throughout the coding, should be set. The model content should be constantly updated if changed while building the model. If working with a system producing articles consisting of several components, and sub-components, a multi-dimensional variable could be set up to store data.

8 References

Aghaie, A., & Popplewell, K. (1997). Simulation for TQM - the unused tool? *The TQM Magazine, 9*(2), 111-116.

Balci, O. (2010, October). Golden rules of verification, validation, testing and certification of modeling and simulation applications. *SCS M&S Magazine*(4).

Balci, O. (2011). How to successfully conduct large-scale modeling and simulation projects. *Proceedings of the 2011 Winter Simulation Conference.* Phoenix.

Balci, O., Ormsby, W. F., Carr III, J. T., & Saadi, S. D. (2000). Planning for verification, validation, and accreditation of modeling and simulation applications. *Proceedings of the 2000 Winter Simulation Conference.* Orlando.

Banks, J. (1998). *Handbook of Simulation: principles, methodology, advances, applications and practice.* John Wiley & Sons.

Banks, J., Carson II, J. S., Nelson, B. L., & Nicol, D. M. (2005). *Discrete-Event System Simulation.* Pearson/Prentice Hall.

Barra Montevechi, J. A., Leal, F., Ferreira de Pinho, A., Florencio da Silva Costa, R., de Olivieira, M., & Faustino da Silva, A. (2010). Conceptual Modeling in Simulation Projects by Mean Adapted IDEF: An Application In a Brazilian Tach Company. *Proceedings of the 2010 Winter Simulation Conference.* Baltimore.

Cetinkaya, D., Verbraeck, A., & Seck, M. (2010). Towards a component based conceptual modeling language for discrete event simulation. *Proceedings of the 24th Annual European Simulation and Modelling Conference (ESM 2010)* (pp. 67-74). Hasselt, Belgium: EUROSIS-ETI.

Chwif, L., Barretto, M. R., & Paul, R. J. (2000). On simulation model complexity. *Proceeding of the 2000 Winter Simulation Conference.*

Cooper, H. M. (1985). A Taxonomy of Literature Review. *Annual Meeting of the American Educational Research Association.* Washington DC: National Institute of Education (ED).

Davenport, T. D., & Prusak, L. (1998). *Working knowledge: Organizations manage what they know, Part 247.* Boston: Harvard Business School Press.

de Vin, L. J. (2012). Credability of Simulation Results - A Philosophical Perspective on Virtual Manufacturing. *Proceedings of the 13th Mechatronics Forum International Conference* (pp. 784-791). Linz: TRAUNER Verlag.

Donatelli, A. J., & Harris, G. A. (2004). Combining Value Stream Mapping and Discrete Event Simulation.

Erikshammar, J., Weizhuo, L., Stehn, L., & Olofsson, T. (2013). Discrete Event Simulation Enhanced Value Stream Mapping: An Industrialized Construction Case Study. *Lean Construction Journal*, 47-65.

Ferrin, D. M., Muthler, D., & Miller, M. J. (2002). Six Sigma and simulation, so what's the correlation? *Proceeding og the 2002 Winter Simulation Conference.* San Diego.

Fowler, J. M., & Rose, O. (2004). Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems. *SIMULATION, 80*(9), 469-476.

Hatami, S. (1990). Data requirements for analysis of manufacturing systems using computer simulation. *Proceedings of the 1990 Winter Simulation Conferance.* New Orleans.

Hernandez-Matias, J. C., Rios, J., Perez-Garzia, J., & Vizan, A. (2008). An integrated modeling framework to support manufacturing system diagnosis for continius improvemnet. *Robotics and computerintegrated manufacturing*.

Khaswaka, Z. N., & Irani, S. A. (2001). Value Network Mapping (VNM): Visualisation and Analysis of multiple flows in Value Stream Maps. *Proceedings of the Lean management Solutions Conference.*

Law, A. M., & McComas, M. G. (1991). Secrets of Successful Simulation Studies. *Proceedings of the 1991 Winter Simulation Conference.* Phoenix.

Massey, T., & Wang, Q. (2007). Modelling & Simulation of Complex Manufacturing Systems. *World Congress on Engineering, 2.* London.

McDonald, T., Van Aken, E. M., & Rentes, A. F. (2002). Utilising Simulation to Enhance Value Stream Mapping: A Manufacturing Case Application. *International Journal of Logistics Research and Applications, 5*(2), 213-232.

Oscarsson, J., & Urenda Moris, M. (2002). Documentation of discrete event simulation models for manufacturing system life cycle simulation. *Proceedings of the 2002 Winter Simulation Conference.* San Diego.

Perera, T., & Liyanage, K. (2000). Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. *Simulation Practice and Theory, 7*, 645-656.

Robertson, N., & Perera, T. (2002). Automated data collection for simulation? *Simulation Practice and Theory, 9*, 349-364.

Robinson, S. (2008). Conceptual modeling for simulation. *Journal of the Operational Research Society*, 278-290.

Robinson, S. (2008). Conceptual modelling for Simulation part II: A framework for Conceptual modelling. *The Journal of the Operational Research Society*.

Robinson, S. (2014). *Simulation: The Practice of Model Development and Use.* Chichester: John Wiley & Sons Ltd.

Robinson, S. (2015). A tutorial on conceptual modeling for simulation. *Proceedings of the 2015 Winter Simulation Conference.* Huntington Beach.

Robinson, S., & Bhatia, V. (1995). Secrets of successful simulation projects. *Proceedings of the 1995 Winter Simulation Conferance.* Arlington.

Rother, M., & Shook, J. (1999). *Learning to See* (1.2 ed.). Brookline, Massachusetts, USA: The Lean Enterprise Institute.

Sargent, R. (2008). Verification and validation of simulation models. *Proceedings of the 2008 Winter Simulation Conference.* Miami.

Seebacher, G., Winkler, H., & Oberegger, B. (2015). In-plant logistics efficiency valuation using discrete event simulation. *International Journal of Simulation Modelling, 14*(1), 60-70.

Skoogh, A., & Johansson, B. (2008). A Methodology for Input Data Managemnet in Discrete Event Simulation Projects. *Proceedings of the 2008 Winter Simulation Conference.*

Smith, A. M., & Chan, F. T. (1993). Simulation Aids JIT Assembly Line Manufacture: A Case Study. *International Journal of Operations & Production Management*, *13*(4), 50-74.

Solding, P., & Gullander, P. (2009). Concepts for simulation based value stream mapping. *proceedings of the 2009 Winter Simulation Conference.* Austin.

Standridge, C. R., & Marvel, J. H. (2006). Why lean needs simulation. *Proceeding of the 2006 Winter Simulation Conference.* Monterey.

Welgama, P. S., & Mills, R. G. (1995). Use of simulation in the design of a JIT system. *International Journal of Operations & Production Management, 15*(9), 245-260.

Yin, R. K. (2014). *Case Study Research Design and Methods.* SAGE Publications.

Yu, H., Tweed, T., Al-Hussein, M., & Nasseri, R. (2009). Development of Lean Model for House Construction Using Value Stream Mapping. *Journal of Construction Engineering and Management*, *135*(8), 782-790.