SAVONIA

Using ERP as Data Source for Discrete-event Simulation Model

Antti Alonen

Master's Thesis in Industrial Management

___.___.______   ______________________________

Master’s Degree (UAS)

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study
Technology, Communication and Transport

Degree Programme
Master’s Degree Programme in Industrial Management

Author(s)
Antti Alonen

Title of Thesis
Using ERP as Data Source for Discrete-event Simulation Model

Date	15.5.2012	Pages/Appendices	70

Supervisor(s)
Jarmo Pyysalo

Client Organization/Partners
Savonia UAS, Junttan Oy

Abstract

The main goal for this thesis was to find out if ERP-systems could be used as a data source for the discrete-event simulation, DES, models. Secondary goals were finding out best practices, creating a practical demonstration and using information from the ERP and DES in the decision support system.

The theoretical part of the thesis includes basic information about discrete-event simulation and simulation projects. Current use and possible ways to connect DES-models and ERP-systems were studied from simulation literature and related articles.

In a case study, 3DCreate simulation software was used to build a DES- model of Junttan Oy’s assembly line. Junttan’s ERP-system was connected to a graphical user interface for the simulation model where input data was processed. The user interface also functioned as a decision support system for production planning.

This thesis resulted in a practical example on how to link ERP system and a DES model in industrial scale manufacturing environment. This allows using simulation models continuously as part of a decision support system. Additionally, a graphical user interface was developed for input data processing and it also functions as a decision support tool for production planning.

Keywords
discrete-event simulation, DES, ERP, IFS, 3DCreate

ACKNOWLEDGMENTS

I would like to thank DigiBranch –project for the opportunity to improve my skills and Savonia's current simulation practices. This project has helped me to understand the complexity of the underlying issues slowing the process of integrating DES-models and manufacturing information systems.

One of the goals for DigiBranch –project was increased cooperation between local industry and Savonia UAS. I would like to thank Junttan Oy for the excellent cooperation during the simulation project. Junttan's production personnel Juha Hakkarainen, Juho Rentola and Juha Petjala were invaluable for this project. Without their help and knowledge of the manufacturing environment this thesis and the link between IFS and DES-models would never had been realized.

I would also like to thank my family and friends for their support during the long evenings I tried to convert my thoughts to an understandable form. Last, but not least, I'd like to thank my supervisor for this thesis, Jarmo Pyysalo, for his support and feedback.

GLOSSARY AND ABBREVIATIONS

DES	Discrete Event Simulation
ERP	Enterprise Resource Planning
IFS	Industrial And Financial Systems (an ERP software provider)
Input data	Data that DES -model requires to operate
IAL	Information Access Layer (information layer for IFS system)
SQL	Structured Query Language
Work Order	a production order that is used to deliver product information in the ERP system
SME	Small to Medium-sized enterprises
DSS	Decision Support System
CMSD	Core Manufacturing Simulation Data
SISO	Simulation Interoperability Standards Organization
RDI	Research, Development and Innovation
R&D	Research & Development
UAS	University of Applied Sciences
JIT	Just-In-Time
Takt Time	The desired time between units of production output
PDM	Product Data Management
CDBS	Central database system
WIP	Work In Progress

CONTENTS

1 INTRODUCTION

Metal industry employs a significant amount of labor in the North-Savo area. One of the strengths of Finnish machine shops is their capability for fast, cost-effective and agile short-run production. Some machine shops in the area are already capable of this but many SMEs do not have time or resources to develop their production processes. Maintaining the competitiveness of the company requires efficient operation from both the factory floor and the connected supply chain. New tools to maintain efficient operations with scant resources are needed but finding the correct ones for the company requires time and research that not all companies have. (Hietikko, 2008)

Production demands of the anchor tenants are delivered real-time to the supply chain. To keep in pace with the demands, local machine shops are in transition towards a networked operational model where modular short-run production requires constant production planning. Forecasting demand is difficult and fluctuating customer demands require fast operational decisions. Production operation efficiency requires managers to be able to predict and prepare for the possible problems in the manufacturing environment.

Manufacturing systems' complexity means that managers can't rely on their product and process knowledge alone and need additional tools to help in the decision making. An example of a complex planning problem that is difficult to solve is a dynamic bottleneck in the production. Dynamic bottlenecks change location depending on the customer orders, available resources and materials. (Heilala, et al., 2010)

Production planners and managers make their decisions based on the available information. They need to balance production while taking into account existing orders, materials and available resources. At the same time they need to provide information on the lead times and possibilities to the sales department. Management decisions are usually based on working experience, data from information systems, calculations and spreadsheets to help them process the information.

Manufacturing Decision Support Systems, DSS, help managers and planners to make production planning decisions. DSS systems have database management capabilities to collect and visualize the information to a form that is more easily comprehended. Managers use the information provided by the DSS system the same way they use any other information but it will provide them necessary information faster. (Shim, et al., 2002; Heilala, et al., 2010)

ERP-systems have been traditionally used by large companies because managing and sharing information on large-scale requires an efficient information system. The need to become more flexible, efficient and to satisfy the needs of supply chain requirements has led to the widespread adaptation of ERP-systems to SME-sector as well. While ERP-systems aim to satisfy the integration needs of the companies and include more than just manufacturing information, they have strong roots in manufacturing materials planning. ERP offers much information that is useful for production planning and offers managers access to real-time operating information. (Davenport, 1998; Nah, 2002)

Using ERP-systems for production planning has some limitations that have roots in the history of the system. One of the problems is that ERP-systems are unable to handle stochastic situations and the uncertainties that happen in the real life manufacturing systems. (Moon & Phatak, 2005)

Planning and scheduling manufacturing processes that are not fully automated means that the working times are rarely constant. Varying working times and the stochastic nature of the manufacturing process has to be taken into account when making production plans.

Discrete-event simulation programs have developed alongside computer technology to user-friendly tools that allow fast and accurate modeling of manufacturing processes. Discrete-event simulation allows taking into account stochastic variables such as varying working times and other uncertainties using random distributions. Using discrete-event simulation models to assist in production planning requires an accurate model and that the input data is up-to-date. Since DES-model requires a lot of input data, it must be easily updated using reliable, up-to-date data and the input process must be automated as far as possible.

At the moment the full potential of discrete-event simulation tools is not being utilized by the companies in Northern Savo. Local industry has a common opinion about DES-tools that they are suitable for one-time, special case situations such as planning of a new manufacturing line. This is based on the relatively poor knowledge of the simulation possibilities and the fact that the usual simulation cases they are familiar with, have been for one-time purposes only. There is no knowledge of how simulation could be used as a tool for production planning even though production simulation is one of the main interest areas.

Experts in the RDI-department of mechanical engineering at Savonia University of Applied Sciences have noted increased customer interest in utilizing simulation models made for layout planning and other ”one-time” –scenarios as a continuous tool for production management and planning. Using DES-models for such functionality has traditionally been prevented by a slow and tedious process of inputting the initial data.

During the last decade there has been an increasing interest in connecting discrete-event simulation models to manufacturing information systems. There have been numerous articles about the problems and possibilities of such a connection. There are also many standards and methods for harmonizing the required data for easier interface with the simulation models and simulation-enhanced decision support systems. (Pyysalo, 2010; Kivikunnas & Heilala, 2011)

Using ERP-systems as a data source for DES-models and enhancing ERP functionality with simulation models has been proven possible. There have been numerous articles about the concept and even some practical approaches have been published. (Moon & Phatak, 2005; Johansson, et al., 2007)

However, simulation enhanced ERP-systems and using ERP as data source for simulation initial data is still not a common occurrence. Only a few articles have been published about real-life, industrial scale solutions with practical documentation.

Savonia’s mechanical engineering RDI –department was interested in improving the usefulness and operational life of DES-models by using ERP as a data source. Savonia started to research the possibilities for this purpose in the DigiBranch project in 2009.

1.1 Goals and scope of the study

This thesis was part of a simulation subproject in DigiBranch-project. DigiBranch is an applied research project in the RDI-department of Mechanical Engineering at Savonia University of Applied Sciences. Because of the scale of DigiBranch -project, the thesis was limited to following topics:

- Researching the possibility to gather input data automatically from Junttan’s ERP-system (IFS) and deliver that information to a simulation model.

- Researching and demonstrating how discrete-event simulation model can be used with a decision support system to help production planning.

Research work using ERP-systems as a data source was important to Savonia because current simulation practices are not practical for building simulation models for continuous use. The reason for the study was to find a way to improve current simulation practices with the over-the-counter simulation programs, instead of competing against DES-enhanced production planning systems such as Delfoi Planner, Simul8-planner or Simio RPS. Because of that, this thesis did not take into account the commercial products already available for advanced scheduling or production planning.

As one of the results, DigiBranch project will introduce a new off-the-shelf simulation program, 3DCreate, to Savonia. 3DCreate will be used to create a simulation model of Junttan's piling machine assembly line. It might also be necessary to create user interface for the simulation model to help to process and handle the initial data for the simulation model.

The author of this thesis worked first as a project engineer responsible for the simulation sub-project and later as a project manager for the DigiBranch -project.

Confidential information related to the case study has been removed from this thesis.

1.2 Research methods and thesis structure

The theoretical part of this thesis is based on seminal works of discrete-event simulation, (Law, 2007; Banks, et al., 2005) and a number of recent publications in the field. The practical part of the thesis was completed in the years 2009-2011. Some of the publications used in the theoretical part appeared after the case study had already been completed, meaning not all of the methods presented were utilized during the project.

The first part of the thesis includes an introduction and the goal of the study. It provides necessary background information about Savonia UAS, its previous discrete-event simulation practices, Junttan Oy and DigiBranch-project.

It also provides enough theoretical information to understand the principles on which the thesis is based upon. It includes an introduction to discrete-event simulation and how it can be used in industry. The different phases of simulation project are presented in detail. A short introduction to ERP-systems, 3DCreate simulation software and SISO CMSD-standard are also included.

Introduction of the practical side of the thesis follows after the theoretical information of study subjects. It provides information on the simulation model, user-interface and transfer methods and interfaces.

Results and conclusions and future recommendations are presented at the end of this thesis.

1.3 Savonia University of Applied Sciences

Savonia University of Applied Sciences (later Savonia UAS) is one of the largest Applied Science Universities in Finland. Savonia's headquarters are located in Kuopio, North-Savo. It offers six different fields of study and has studies in both Bachelor's and Master's degree. It has over 6000 students and a staff of 600. Savonia has campuses in three different cities: Kuopio, Varkaus and Iisalmi.

Savonia UAS has an active research and development department that offers high-quality services to the local industry. Savonia's Research, Development and Innovation department is organized by three competence networks: “Welfare Products and Services”, “Energy, Environment and Safety” and “Integrated Product Development”. Simulation services are part of the Mechanical Engineering RDI Unit, which is part of the Integrated Product Development competence network.

Simulation in Savonia UAS

Savonia University of Applied Science has been using discrete-event simulation, later DES, software in education and promoting their use to the local industry since 2000. The use of DES in Savonia started with WaLT-SIM –project (2000-2004) and later continued in Savonia's business services unit in Varkaus. In 2009 all simulation activities were concentrated in Kuopio with the start of simulation research as a part of DigiBranch -project.

The first DES software used by Savonia was Dassault Delmia's QUEST and it was used extensively between the years 2000-2009. Numerous simulation models were made with different versions of QUEST. During this time the problems with the simulation software and its unintuitive user interface complicated the use of DES models both in RDI-services and education.

The main customer base of Savonia's simulation modeling services has been the local metal industry and companies providing services for it. Most of the manufacturing systems have been assembly-oriented. Simulation models have also been made for various companies in other industries, including models to the health care sector by Savonia, Varkaus, Industrial Services unit. Simulation has proved to be a good tool for comparing different options and finding out the problems of the existing plans before they are implemented. Several models have also been used for marketing purposes.

A single most used reason for simulation has been using simulation to assist in layout planning. In many cases the interest for simulation models has come from the need to be able to produce "what-if" –scenarios to help to predict how different situations affect the production capabilities of the system. This includes analyzing different production mixes and predicting what will happen if critical resources fail for some reason.

Most of the simulation cases Savonia has worked with have been one-time "offline" simulations. Many such cases have been for customers that have had no data collecting systems that has resulted in tedious data collection periods. From that viewpoint it's not surprising, that the dominant opinion in local industry is that DES-models are mainly usable in development projects, such as the layout-planning of new production lines.

Collecting and handling large amount of initial data for a simulation model, has been a major obstacle in many projects. Savonia's client companies have been mainly SMEs which haven't had MES, ERP or any kind of professional data collection systems, for various reasons. Some companies are using the "old and proven" ways in their production, and data collection systems have not been part of the company culture. Many companies think that the cost of a MES-system is too high compared to the value gained from it. The lack of such a system often results in complicated and slow gathering of data.

Different ways to overcome this obstacle have been tried in the past - some of the projects included an external interface for harmonizing, inputting and processing the initial data. Other commonly used means for input data handling were also used, such as reading data from external file sources (excel files, .csv & etc.)

DigiBranch

Savo Consortium for Education, Sakky, is building a learning environment to Kylmämäki, Kuopio. The project is called “Metallialan Oppimistehdas ja digitaalisen tuotannon tutkimusyksikkö” and it was scheduled for years 2009-2011. Funding of the project is provided by the Finnish Government, The European Social Fund (ESF), The city of Kuopio, Sakky and Savonia UAS. Company partners Junttan Oy, Komas Oy and J-Metallikaluste Oy were also involved in the project.

There is a significant concentration of metal industry in the area of Northern Savo, main suppliers, system providers and multiple component suppliers. The region is an ideal learning environment, since it offers real industrial surroundings for students and teachers. The companies in the area are also interested in cooperation with the learning institutes, because it ensures that graduating students will be familiar with their production systems and learn necessary skills.

The digital production research unit, called DigiBranch, is part of the learning environment project coordinated by Sakky. DigiBranch is managed by the Mechanical Engineering RDI-unit of Savonia UAS. The purpose of DigiBranch -project is to form a research unit that will provide significant addition to the technical capabilities, knowhow and productivity of the involved partners. (Hietikko & Suhonen, 2010)

DigiBranch concentrates on deepening the cooperation between involved educational research institutes, companies and their networks. It has several focus points which are:

- Simulation
- Virtual production
- Lifecycle Management
- Manufacturability
- Prototyping
- Structural Engineering

Discrete-event simulation was selected as the most important focus point of DigiBranch because of the possibilities it offers for improving the competitiveness of the industry. The limitations of Savonia's current DES modeling practices were already well known from the previous simulation projects. The DigiBranch –project offers an opportunity to find solutions to current problems and to increase the use of simulation in Northern Savo.

The main goals of the simulation project were specified as:

- To select a modern simulation software that is easier to use than Quest, and demonstrate its capabilities by creating a simulation model of a large production line.
- To study the use of discrete-event simulation models in everyday production management as enhancing the decision support systems functionality.
- Linking the ERP-system (IFS) to the simulation model for up-to-date and reliable input data.

1.4 Junttan Oy

Junttan Oy, founded in 1976, is a world-leader in hydraulic piling machines. It specializes in the designing, manufacturing and marketing of hydraulic piling equipment. The main products of Junttan are pile driving rigs, multipurpose piling and drilling rigs, deep stabilization machines, hydraulic impact hammers, rotary heads and power packs. (Junttan Oy, 2012)

Junttan's way of manufacturing products had traditionally been a classic workshop –model with a single work cell oriented production. One team manufactured one product, e.g. a pile driving machine, in a same work cell and location from start to finish. The work cell crew had a significant impact on finished products meaning every product could be slightly different from each other, depending on which crew had been working on it. The strong role of work-crews, and their individual way of working, made it possible that even though the pile driving machines looked identical on the outside, they were assembled in a slightly different way. An example of this is wiring – the employees used different routes for electrical wiring which caused problems for the maintenance later in the product life cycle.

In 2006 there was a change in ownership of the company, which started the modernization of production. Junttan set an aim to significantly increase the production capacity resulting in the change of production method. Junttan concentrated on its core competence, assembly of the piling machines, and outsourced the fabrication process of most parts. This made the change from work cell oriented production to assembly line production possible.

In 2008 Junttan introduced an ERP-system (IFS) and built a new factory in Kylmämäki, Kuopio. The new factory layout was also planned with future expansions in mind. The production was transferred to new facilities in 2009. (Tähtinen, 2007)

With the introduction of the ERP-system and the new assembly line a lot of changes were required. Product structures and designing had to be harmonized to allow assembling of the products in the assembly line.

Junttan's production is mainly assembling of products. The assembly of the upper carriage of a pile driving machine was designed during 2007-2008. It was documented and standardized to ensure that the products are always assembled in the same order. (Tähtinen, 2007)

Pile-driving machines were divided to two different categories in respect of their complexity. Standard, modular volume products, are assembled on the assembly line. Heavily customer-specified products are assembled in a work cell oriented manner, separate from the assembly line. The aim is to reduce the amount of customer-specified products by providing a sufficient amount of different modules that the customers can choose from.

Because of the swift change in the way production is done Junttan still has many development projects under way to enhance the efficiency of production. The sheer amount of change, in such a short time, means that not all systems are yet used at their full capacity.

The new production system and increased production capacity means that there is an increased need for a decision support system that would help production planners and managers to make fast and correct decisions. ERP- and PDM-systems make it possible to receive up-to-date and reliable input data. At the moment Junttan's production decisions are based on long working experience, knowhow of the products and production line, data provided by the information systems and calculations collected in spreadsheets.

Because the information is collected from many sources, and requires a lot of experience to handle, it is challenging for responsible personnel to provide accurate information to the sales department. The sales department works under pressure in order to provide customers accurate estimations of delivery dates. Currently production planning is cumbersome to use and it's not easy to compare different options for production planning.

A decision support system is needed to streamline the decision making process by automatically collecting necessary real-time information from different systems.
The main need for decision support is in production mixes and different options in production plans. It is essential for production planners to know if certain orders can be produced in the assembly line or if they need to be produced separately from the volume production.

FIGURE 1. Junttan's hydraulic piling machine (Junttan Oy, 2012)

2 BACKGROUND INFORMATION

This chapter provides the theoretical background information that is necessary for understanding the underlying issues and the reason for this thesis. Information is provided on discrete-event simulation, ERP -systems, manufacturing information systems, existing ERP-DES connections and SISO CMSD –standard.

2.1 Discrete-event simulation, DES

“Simulation is the imitation of the operation of a real-world process or system over time. It involves the generation of an artificial history of a system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system.” (Banks, et al., 2005)

Discrete-event simulation is a collection of events that happen in chronological order and change the system’s state. The state of the system (or part of it) is changed instantly when an event happens. Discrete-event simulation models are used to study how the system works during the period of observation. (Banks, et al., 2005; Law, 2007)

A discrete-event simulation model can be created in different ways depending on the available resources and objectives of the model:

- General programming language (C++, C#, Visual Basic, Java & etc.)
- Simulation specific programming language (Arena, Simscript, etc.)
- Off-the-shelf, tailored discrete-event simulation programs, such as
    - 3DCreate, Automod, ProModel, Quest, Taylor/ED, ProcessModel, etc.

Using off-the-shelf programs offers many benefits comparing to other options in a case of regular manufacturing simulation case. The benefits include a shorter time required to build a model, greater formability of a model and reduced amount of mistakes resulting from writing the code. Off-the-shelf programs have many necessary features that are well planned and have evolved with the development of the software. There are simulation programs that are built specifically for certain simulation purposes and many programs offer modules for adding specific functionalities, such as health care packages, logistics, conveyor systems & etc. (Law, 2007; Banks, et al., 2005; White & Ingalls, 2009)

However, using off-the-shelf simulation program does not remove the need of writing the model and case-specific functionalities with programming languages. Simulation programs have a lot of features built-in the software and thus they reduce the amount of coding needed. Most simulation cases are more complex than what the simulation program user interface allows, requiring a simulation practician to write some of the model behavior manually. For this reason, practically all simulation programs offer internal programming language used for that purpose. Some of these programming languages are general programming languages like C, Python & etc., some simulation specific programming languages and some scripting languages from between the previously mentioned.

Most of the Savonia’s simulation cases have required writing a significant portion of the simulation model behavior by hand. Modeling human resources is a weak point in most simulation programs and most of Savonia’s simulation customers are from local metal industry where work processes involve a lot of manual work phases done by humans.

Simulation models are used by observing the behavior of system entities that are the objects of interest during a specified time window. Simulation models are not solved and they do not automatically provide solutions to specific problems. Instead they are used by “running” the model for a certain amount of time and then provide information of the outcome of the model using predefined input parameters. The model and states of the system variables can then be analyzed to draw conclusions about the system. (Banks, et al., 2005)

Some DES-programs, such as Quest, 3DCreate and AutoMod have built-in 3D-engines and simulation models are built in the 3D-environment. In other programs, the simulation model is first built in a 2D-environment and it can then be visualized in 3D-video mode by introducing different geometries to components. This is useful depending on the purpose of the model. 3D-visualization can be highly attractive if the purpose of the model is to find out the big picture or visualize the workings of the production system for people who are not very familiar with the system.

The term ”simulation model” is used in this thesis to describe a discrete-event simulation model that is built in a 3D-environment with off-the-shelf software and represents a manufacturing system. It is built using a set of assumptions on how the system works, and it takes into account relationships, and interactions between system elements, such as material flow.

The purpose of the simulation model defines what entities (objects of interest in the model) and variables will be tracked during the simulation run and what parameters are used to define the operation condition of the model. Examples of simulation parameters could be the order list variables such as lot size, work cell related variables such as machine speeds and resource related variables such as shifts, breaks and so on. (White & Ingalls, 2009; Law, 2007; Banks, et al., 2005)

2.2 Classifications of simulation models

Simulation models can be classified by their characteristics in three dimensions. A model can be static or dynamic, deterministic or stochastic and discrete or continuous. (Law, 2007, p. 6; Banks, et al., 2005)

Static simulation model represents the working of a system in a single point of time.

Dynamic simulation models represent the system and its behavior during a defined time window.

Deterministic simulation models do not take into account any stochastic variables and the end result of simulation can be calculated from the input parameters. The only thing affecting speed of receiving end result is the complexity of the model and the amount of processing power available to perform the simulation.

Stochastic models take randomness into account in a way or another and it can have a significant effect on the system. Usually randomness and stochastic variables are taken into account with the help of statistic distributions. Examples of random variables represented with distributions are product demand, machine breakdowns & etc. Because of the stochastic nature, the results of the model are always estimations on how the system works and this must be taken into consideration when analyzing the results.

In **Discrete** models state variables change state regardless of each other. An often used example of a discrete system is a bank (Banks, et al., 2005). Examples of state variables in a model representing bank could be customers. Customer -variables changes state only when a customer arrives, completes a task or is removed from the model.

In **Continuous** simulation models, state variables change state continuously during the simulation time. An example of continuous simulation model could be flying an airplane, where the state variable (flying altitude) would change continuously.

Using the foregoing classification the discrete-event simulation model built in the simulation project and this thesis can be classified by its characteristics as dynamic and stochastic.

2.3 Why use discrete-event simulation?

Discrete-event simulation (DES) is one of the most used modeling techniques since the computer aided simulations started to become common in the 1950's. The development of DES has been going hand in hand with the development of computers and information technology. Compared to the early stages of DES simulation models, modern simulation programs have developed significantly in their features and modeling capacities. Perhaps the most significant change has been the way programs are used – modern user interfaces have dramatically improved the ease of use and the development is still going on. Ease of use is important because it is lowering the threshold of using DES -models in the future. (Robinson, 2005; Banks, et al., 2005)

Using discrete-event simulation offers numerous benefits that have been well documented by simulation authors of the field. The following short list of benefits has been collected from many literature sources and condensed to include the benefits of typical simulation cases Savonia usually does. (Banks, et al., 2005; Law, 2007, pp. 76-77)

Benefits of simulation:

- The principles of discrete-event simulations are easy to understand.
- One of the most important features of simulation models – controlling time - allows the observation of a system during a certain simulation period. Simulation model time can be slowed down or sped up depending on the needs of the user. Weeks, months and even years of events can be processed in a matter of minutes.
- Simulation is safe. We can observe a system in action without touching the actual system. This allows observing the current system behavior, altering it with “what-if” –situations and testing out the effect of new production decision.

Discrete-event simulation models are commonly used in the industry for the following purposes:

- Capacity calculations.
- Analyzing throughput and lead times.
- Layout-planning.
- Balancing production.
- Supporting investment decisions and as risk-management tool.
- Identifying bottlenecks and testing out control techniques.

Simulation models provide a wealth of information as a result of simulation runs. In fact, the amount of information produced is considered as a weak point of simulation models, as it can be challenging to find out the results that respond to the research problem.

Results related to items of interest can be collected from the simulation run in the same accuracy as that the model is built. One of the benefits of the off-the-shelf simulation programs is that they usually automatically collect results of simulation run. Outlining and selecting information that needs to be collected as results is basic functionality in most simulation programs. Programs have different ways to show the collected simulation run results to user, and practically all programs have made it easy to save that information in common text-based files that can be opened as spreadsheet. Many simulation programs also offer ways to export the results directly to database-based statistical software such as Minitab®.

2.4 Simulation project steps

Building a simulation model is only one step in a simulation project that does not usually start by building a model. Law (Law, 2007, p. 66) argues that it is important to understand that model building is only part of the simulation project and other steps are as important for successful model building.

Simulation project is usually divided in to several steps to make it easier to understand and control. Simulation authors have many different ways to divide the simulation project to steps according to their viewpoint on the matter. (Banks, et al., 2005, p. 15; Law, 2007, p. 67; Pooch & Wall, 1993, p. 6)

Building a simulation model is considered to be as much art as science (Banks, et al., 2005) meaning there are many different ways to build a model. Because the actual building of simulation model is only one step in the project, but is in turn related to other steps, it means that the order and definition of these steps can vary. However, there are certain basic steps that are essential to simulation project that will always be completed during a simulation project.

Simulation project flow chart (Figure 2) is based mainly on the simulation steps defined by Banks & al (Banks, et al., 2005, p. 15) and Law (Law, 2007, p. 67). The flow chart has been generalized and it describes in a rough level the steps included in a typical manufacturing simulation project carried out by Savonia Mechanical Engineering RDI –unit. Steps are described in a necessary detail to serve as background information for this thesis. Short descriptions of the steps are:

- Problem definition and setting of objectives.
- Collecting initial data and building of the conceptual model.
- Model building.
- Model verification and validation.
- Simulation runs.
- Analysis and documenting.

Simulation project flow chart shows six steps that are divided into three different phases depending on the timeline of the project.

- Initial phase includes steps one and two, and takes place before building of the simulation model starts.
- Phase two includes steps three and four which are the actual model building steps.
- Phase three is at the end of the project, after the simulation model is built.

Some of the steps can occur partly or wholly at the same time with other steps depending on the simulation team’s size and project schedule. Possible overlapping of the steps has been illustrated in Figure 3.

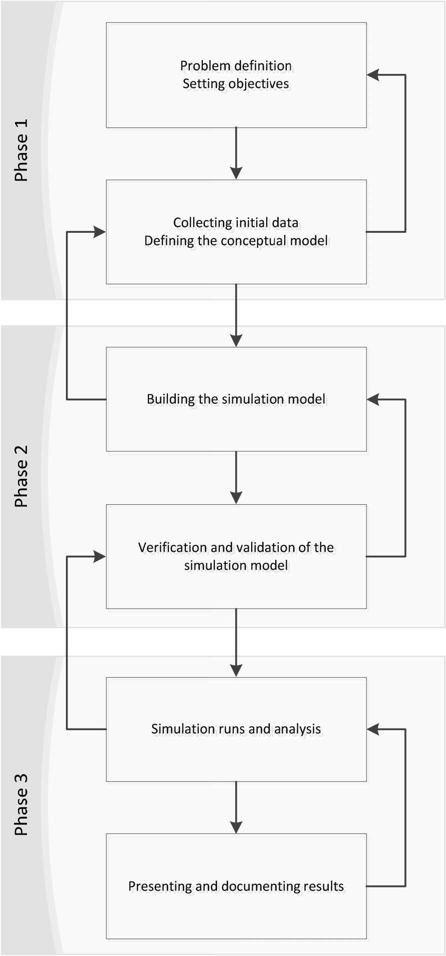


FIGURE 2. The phases and steps of a general simulation project (based on Banks & al., 2005 and Law, 2007).

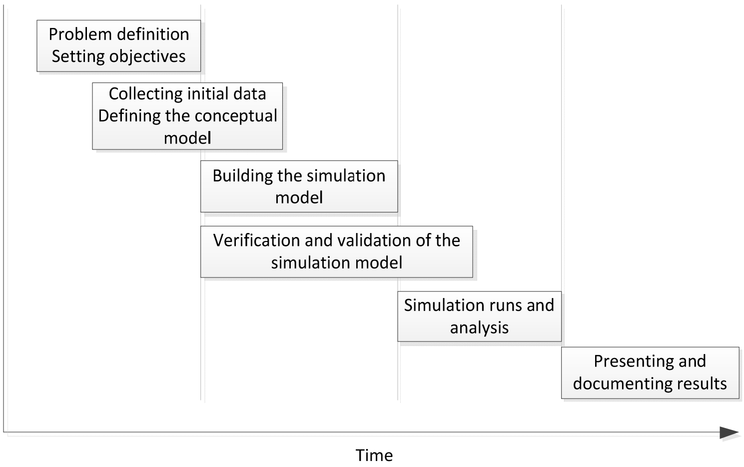


FIGURE 3. Overlapping of the simulation project steps (based on Banks & et al., 2005 and Law, 2007).

The following pages give more detailed descriptions of the simulation steps presented in the flow chart, Figure 2.

Problem definition and setting of objectives

Simulation models should always be built for a specific purpose such as a need to solve a certain problem. (Banks, et al., 2005, p. 14; Sargent, 2011)

Problem definition can be one of the most difficult things in a simulation project. The problem must be defined accurately enough so that it can be used to prepare objectives for the project. Setting a clear objective for the simulation model is important because it will reflect on the model structure and state what results are expected of the model.

It is not uncommon for customers to add additional objectives for simulation model during the project. This is usually a result of poor knowledge about simulation possibilities. If the additional objectives differ from the original objectives it can result in a redefinition of the whole simulation project and its objectives.

When defining the research problem and the simulation model's purpose it is also important to consider the other steps in the simulation project, available resources and project timetable. (Banks, et al., 2005; Law, 2007)

Collecting input data and conceptual model definition

Input data is needed in the model building phase to create a simulation model of the system. Input data is also needed in the simulation runs, to set the simulation variables and produce simulation results. Collecting input data has a crucial role in a simulation project and can be thought as a bottleneck in a project (Pooch & Wall, 1993, pp. 20-21). The needed accuracy of the input data depends on the project's objectives and accuracy can vary during the simulation model's building phase. (Banks, et al., 2005, p. 16)

Inaccurate input data is one of the most common reasons why simulation projects fail. Defining who is responsible for collecting input data, at what project stage it must be collected and how to validate that the collected data is correct is as important as defining what data is needed for the model. Collecting the input data can consume a large part of the time available for the simulation project. (Banks, et al., 2005, p. 16; Bengtsson, et al., 2009; Moon & Phatak, 2005; Robertson & Perera, 2002)

A conceptual simulation model is needed in order to find out what input data is needed for the simulation model, as well as describing how the system works. Since defining the conceptual model can take a long time it is enough that it's defined to a level that shows the required data before starting data collection. Initial data is an essential part of this thesis and more detailed description of the required data can be found from chapter 2.6, "Simulation input data".

Building a simulation model

Building a simulation model can be divided into two parts. The first part is finishing the conceptual model where the actual simulation model is built based upon. Conceptual model is a collection of assumptions on how the system works. Building a conceptual model usually means generalizing the workings of a system where possible. Generalization should be done in a way that does not endanger the accuracy of the simulation model.

After completing the conceptual model it is "transferred" to a computer model. The conceptual model is often done with the help of the process flow charts and some simulation models allow direct transfer of these charts to the simulation program. However, the conceptual model includes a lot of information that can't be included in the process flow charts. After transferring and building a simulation model it should represent the real system and provide similar results.

Model verification and validation

Model verification and validation means ensuring that simulation model works the way it was intended and that both the input data that is fed to the model and the results simulation model provides are valid. Usually verification and validation is started while building the simulation model. (Robertson & Perera, 2002; Sargent, 2011)
Large simulation projects may also use independent verification and validation services provided by a third party. (Sargent, 2011)

Validation has to take into account the purpose of the model. Models are usually made for a specific purpose and can be valid only for that purpose. A change in model specification could mean that the approach is no longer valid for the purpose.

Several authors point out that it is important for simulation end users to have confidence in the model. To achieve that “face validity”, a simulation model should be reviewed by end users and subject-matter experts. If they agree that the model works correctly and provides correct results, it has face validity. Face validity is easier to achieve when simulation end users and subject-matter experts are part of the project team that is building the simulation model. This ensures that they know how the simulation model is built and gives them confidence in the model. (Law, 2009; Banks, et al., 2005, p. 362)

Simulation runs and analysis

Simulation runs include selecting proper input parameters to the simulation model and running the simulation model multiple times. Because simulation models do not automatically produce an optimum result, the simulation run has to be run enough times to receive satisfactory results. If the simulation model has stochastic variables, several simulation run iterations and a proper statistical analysis are required before making conclusions of the results. (Banks, et al., 2005, pp. 383-423)

Several simulation programs offer additional software packages made for optimization such as OptQuest® that makes running several simulation runs easier. These packages allow the user to define ranges for the input parameters that the optimization tool can use to run the simulation model several times with different input parameters, compare the results, and try to provide an optimum result.

While simulation model builders can draw conclusions of the results and model behavior, they are usually not the experts of the manufacturing system in question. Simulation end users or subject-matter experts are responsible for analyzing of the simulation run results.

Presenting and reporting

Presenting and reporting is the last step in the simulation project. It is important to create complete documentation of the simulation model and how it was built in case the model will be used in the future. Model building often requires writing customized behavior logic and it has to be properly documented so that other people understand the code and reasons for it.

Results that the simulation model should provide are defined when the simulation model's purpose and simulation objectives are defined. In the case of Savonia's simulation models, it is Savonia's responsibility to provide customers with simulation results, to ensure that the customer understands model accuracy, and all things that affect it.

Since most simulation cases are unique in a way or another, it is possible that something unexpected turns out during the project that requires considering previous simulation project steps again. A common step where an "unexpected" turn of event happens is after the model objectives are defined, and a when conceptual model has been developed to a stage that allows collecting initial data. If initial data is too tedious, or even impossible to collect, the simulation model objectives should be reconsidered.

As previously mentioned, practically all simulation models made by Savonia's Mechanical Engineering RDI-unit have been made for a "one-time purpose", meaning that the purpose of the simulation model ceases to exist shortly after the model is completed. In those cases simulation project steps proceeded once according to a previous simulation project flow chart. The challenge of feeding input data to the model has been a major reason in creating simulation models for "one-time purpose". If the simulation model is going to be used only once for a specific purpose, such as layout planning, the collecting and entering the input data, even if slow and tedious, is not a problem.

However, if the simulation model purpose is to work as a tool to support production decisions, input data needs to be constantly collected, updated and fed to the model. Even if some of the data needs to be collected only once, e.g. manufacturing system layout and machines, there is still a lot of information that would need to be updated every time a simulation model is run. Simulation input data update needs are described in more detail in chapter "2.6 Simulation input data".

2.5 3DCreate simulation program

Visual Components Ltd offers a variety of 3D-modeling and simulation programs. 3DCreate is the premium package of the Visual Components' software family. It offers a component-based approach to 3D-simulation of complete factory layouts and manufacturing environments.

The operational principle of the software is a component-based approach. It allows creation of components that can be customized to create the needed functions and behaviors. Created components are saved to the component library and can be reused on many simulation models depending on the interfaces and functionalities built on them.

The simulation time mechanism of the software is not purely discrete, as some components allow “fixed-time” style functionalities. Examples of such components are conveyors where the functionalities can be polled with certain intervals to accomplish required behaviors. With those exceptions, the software is based on discrete events.

3DCreate has built-in COM and Python API interfaces for software developers. Easily accessed interfaces that allow direct manipulation of simulation software, and its features, was one of the reasons why 3DCreate was selected as simulation software for the project. These functionalities allow building of the external user interfaces for the simulation models, and they were used to create a data link between the simulation model and user interface. 3DCreate uses a widely known general programming language, Python, as the internal simulation language.

However, 3DCreate is not the only simulation software that has features allowing software developers the access to all of the program features via general programming languages such as C#.

2.6 Simulation input data

In this thesis, terms “input data” and “initial data” are used to represent the information and necessary input data that a DES-model requires to operate properly. GIGO, an acronym usually used with information technology and comes from the words “Garbage In – Garbage Out” is something that holds true in simulation data as well. Simulation input data is critical for the success of the model, so it is important that the data is reliable and valid. (Sargent, 2011)

Input data can be categorized in many different ways. Categorization could be defined, for example, by its functionality, what part of manufacturing process it comes from, or how fast it becomes obsolete. CMSD-standard, for example, uses functionality (Layout data, process data & etc.) to categorize different types of data.

If the purpose of the DES-model is to be used continuously as part of decision support system, then it’s essential to categorize data also by its expiration date. There is a lot of information that needs to be input only rarely, while some of it needs to be updated every time a simulation model is run. For a typical simulation project Savonia is commissioned for, the input data could be categorized by the required data update rate into three categories:

1. Static data that needs to be input to the model only once.
    - E.g. layout of the manufacturing environment, static machines, processes and their general working logics.
2. General data that doesn’t require update for every simulation run, but that must contain update option for different production runs.
    - E.g. working shifts/overtimes, resources, subcontractor capacities & etc.
    - Dynamic machines that can switch purpose or location, material flow.
3. Production run specific data that needs to be updated for every simulation run.
    - E.g. Order list, machine/setup-times for different product classes, material flows & etc.

The following lists different general initial data more accurately according to its function:

- Products
    - Part structures from all parts that are used in a simulation model. This includes sub-assemblies, pre-assemblies, subcontracting parts & etc. Because this is essential manufacturing information for any company, information related to products is often easily accessible via manufacturing information system such as ERP or PDM-software.
- Work cells and machines
    - Working logic for work cells and/or machines and related information that needs to be configured as simulation parameters.
        - Working times.
        - Rules and predecessor constraints (e.g. stage 1 needs to be completed completely before stage 2 begins).
        - Required resources (such as humans, cranes, forklifts).
        - Resource -related priorities (e.g. work cell 1 is a priority work cell that always gets required resources even at the

        cost of other work cells), how different amount of resources affect production.
        - Setup-times, e.g. for changing tools or setting up a welding jig.
- Resources
    - Resource capacity limits.
    - Resource restrictions.
        - Physical restrictions, such as lifting limits for cranes.
        - Employee know-how matrices: are all employees capable of doing every stage or are there different resources for different jobs.
- Processes
    - Is it possible to complete the same work phase on different locations / work cells or does it always require the same work cell.
    - Process priorities – are some processes more important than others.
        - Can some processes borrow resources from other processes for time-critical work phases.
- Material flow
    - Material flow rules: how the material flow is moving between work cells.
    - How does the material flow move between different work phases, does it need external resources such as cranes, agvs, forklifts or humans. Is there a limited amount of resources available for the material flow and does using those resources affect working processes.
        - E.g. using a crane requires processes 2 and 3 to stop for safety reasons.
- Load information
    - What kind of load is the manufacturing system under (production mix and order lists)?
- Warehouse information
    - Warehouse and buffer capacities.
    - Buffer rules: e.g. cooling times (product needs to cool down in a buffer for 1 hour before proceeding.
    - Stacking rules: e.g. certain products can’t be stacked on top of others & etc.
- Shifts and working times

  - Shifts, work times, vacations, absences (e.g. average absence rates for different worker groups & etc.).

- Visualization
  - Geometric information for products, resources, machines and accessories.
  - Is the simulation model going to be visually simple and model building concentrated on output information or does the model require more detailed production geometries?

2.7 Data collection methods for DES -models

The amount of input data required for simulation model depends on the model size, complexity, accuracy and level of detail. Naturally, more complex and detailed models require more information. Data amount also varies depending on the simulation purpose: are we feeding data to a model that already exists or are we trying to build the whole simulation model based on the initial data.

Different methods to collect simulation input data have been discussed by various authors during the last decade. Robertson and Perera (2002) defined four different methods to collect and input data for the simulation model as the following:

1. Manual input of the input data to the simulation model. This method is often used with off-line, one-time simulation models.
2. Manual input of the data to a computer application, where the simulation model can automatically retrieve necessary information
3. Gathering information automatically from manufacturing information systems to an intermediary database where it's stored. The simulation model will automatically retrieve necessary information from the intermediary database.
4. Automatic collection and input of required initial data directly from a manufacturing information system to simulation model

The simulation model can also require information that is not directly available in any manufacturing information systems. There is also some automatically collectable information that the simulation user may want to change for different production run results. This requires that data can be changed directly in the simulation model or that there is a program that acts as simulation user interface and allows the modification of the necessary data. Modifying input data directly in a simulation model can be difficult because simulation programs are not usually created for that purpose. Creating input data handling function to a simulation program may require writing a lot of additional logic code.

None of the previous methods provide a way for both the easy data modification and automatic updates. Intermediary database allows external data storage, but input data is not easily modifiable without user interface. Computer application allows easy modification of input data, but large amount of data is difficult to handle without a database. Combination of the methods two and three, building an external user interface or data processing software with a database, is a proven and an increasingly common solution for handling the input data.

External user interface and data processing software can be used for data processing purposes such as data modification, cleaning and creating statistical distributions. User interface can also offer multiple ways of automatically harmonizing the input data and exports to multiple formats. This approach has been used by several authors to create CMSD-specified input data. A middleware solution called Generic Data Management Tool (GDM-Tool) was used to collect, modify and harmonize the input data and generates CMSD-specified XML-files. (Skoogh, et al., 2010; Boulonne, et al., 2010; Bengtsson, et al., 2009)

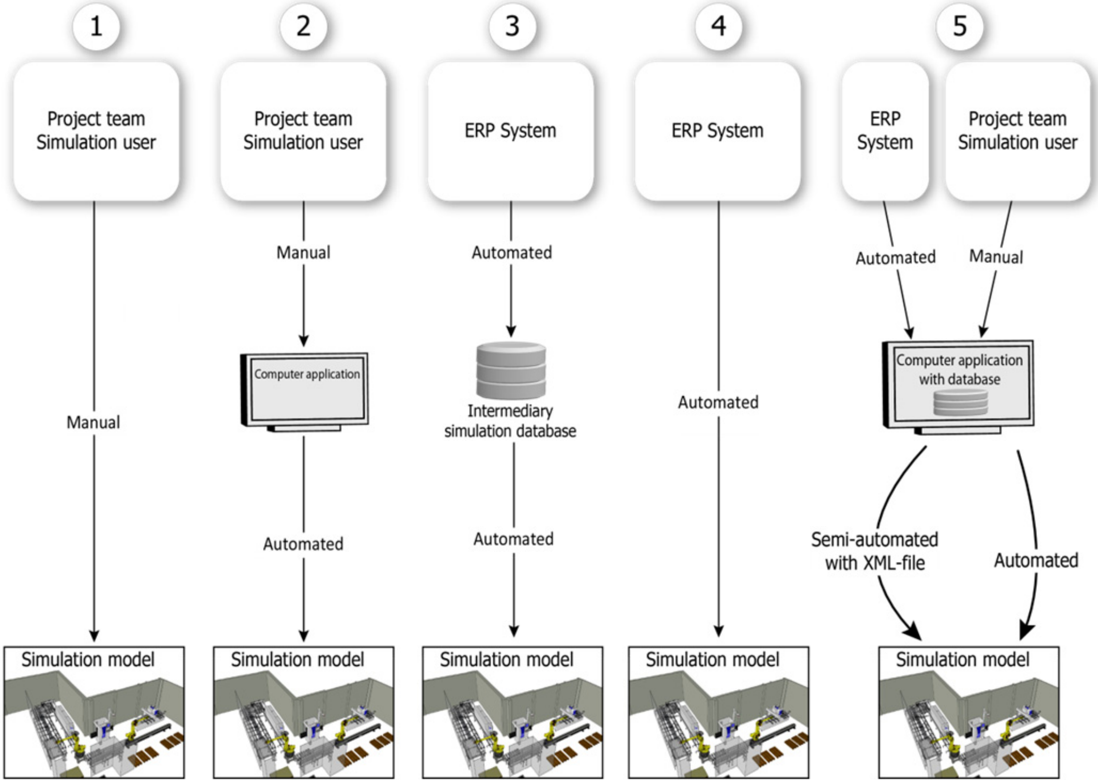


FIGURE 4. Possible data collection methods (modified from Robertson & Perera, 2002)

Figure 4 shows a modified version of methods proposed by Robertson & Perera. It has been modified by combining methods two and three to make up a fifth method.

Part of the input data can be collected from ERP -systems to a user interface that stores the data in an intermediary database. It allows modifying the automatically collected data or adding completely new information from sources not available for automated data collection.

The amount of initial data can be huge and not all data needs to be updated every time the simulation model is run. To avoid unnecessary data updates, users should be offered the option of updating certain data, instead of automatically updating everything. Simulation model can request and collect required input data automatically from the user interface. Input data can also be exported to external input data files, that simulation models can then use as data source.

External input files can be useful in several ways. If input data is transferred via file, simulation model does not need to be directly connected to the user interface. The same file can be transferred via email to recipients and used simultaneously in several simulation models. If the data file uses standardized way to represent information it could be used with different simulation programs.

Manufacturing information systems, such as ERP -systems, make automatic data collection possible. Missing neutral data structures pose a challenge. At the moment there is no widely used standard for specifying what input data is used with simulation models and in what form the data is represented. This makes interoperability between programs challenging and information transfer difficult. (Bengtsson, et al., 2009)

A widely accepted, standardized way to represent input data could make the use of ERP -systems as DES -model data sources more common. The need has been recognized and currently there are many different projects aiming to define and harmonize required initial data for simulation models.

Recent literature survey lists existing and incoming standards and data transfer efforts between manufacturing information systems and simulation models. The list includes the following efforts: ISA-S95/IEC 62264, MIMOSA/ISO13374, SISO CSMD, MTCONNECT, AutomationML, OAGi, SDX, SISO COTS, ISO 10303-239 PLCS. (Kivikunnas & Heilala, 2011)

Several of these standards use UML and XML as presentation modules. XML has several advantages as a data format and these are discussed in more detail in the next chapter.

CMSD-standard is a promising neutral data transfer format for discrete-event simulation models. It is open source, uses XML-representation and there are already several case-examples describing its use with DES -models. (Johansson, et al., 2007; Boulonne, et al., 2010)

2.8 SISO CMSD Standard

CMSD (Core Manufacturing Simulation Data) is a standard published by SISO (Simulation Interoperability Standards Organization).

CMSD aims to harmonize data exchange between manufacturing simulation applications (DES models) and manufacturing systems applications (MES, ERP & etc.) by providing neutral data structures. CMSD will eventually define interfaces for the whole manufacturing lifecycle and expand to include supply-chains as well. (Leong, et al., 2006)

CMSD information model is presented with two different methods: Unified Modeling Language (UML) and schema language for XML representation. UML definition is published as SISO standard product, SISO-STD-008-2010.
(SISO-STD-0xx-2011, 21.3.2011)

UML is a standardized, general purpose modeling language used in the field of object-oriented software engineering. UML standard was created and is managed by Object Management Group. UML diagrams are used for describing interrelationships between entities in the manufacturing field and XML files are a way to store and exchange that information.

XML is a simple, universal format for representing structured information. It is one of the most widely-used formats for sharing structured data today. It has number of advantages over other formats, the main advantage probably being its wide use in the industry today. The structured representation makes it very readable by both human and machines. One of the main reasons why XML is so widely adapted is that it's completely free of charge and can be used by anyone.

An **XML Schema** is a language for expressing constraints about XML documents. Among other things it provides list of elements and attributes in a vocabulary. XML specification is published by W3C (World Wide Web Consortium).
(W3C, 2011)

In their CMSD specification test implementation, Johansson & al (Johansson, et al., 2007) list issues that should be taken into account when using CMSD:

- Parsing xml-documents in DES software can be very memory intensive if the data file is large.
- CMSD was designed as information model for job shop models, but can be modified easily enough to take into account the needs of flow shop modeling
- Understanding CMSD is necessary before building an implementation model
- Anyone with sufficient knowledge of DES-software, XML and CMSD is able to write a script for CMSD implementation. It also provides non-

simulation experts the opportunity to understand the data structure and automatic data inputs from other systems.

CMSD manufacturing data categories are specified as organization, calendar, resources, skill, setup, operation definition, maintenance definition, part, bill-of-materials, inventory, process plan, work, schedule, revision, probability distribution and reference. (Leong, et al., 2006)

CMSD XML-files allow manufacturing information exchange between manufacturing information systems, manufacturing applications and DES-models. An example of manufacturing information exchange is illustrated in Figure 5.

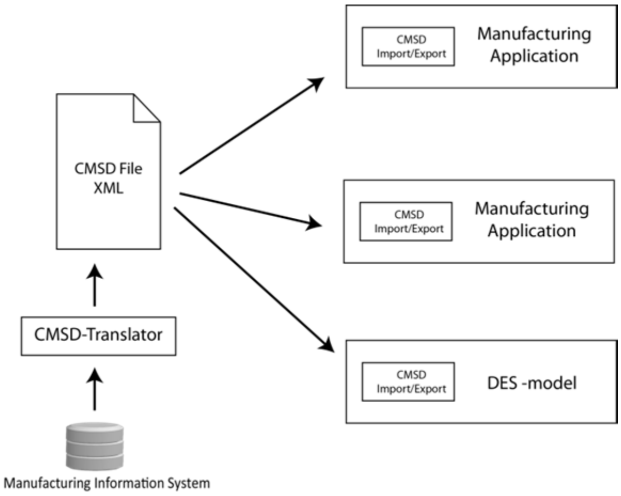


FIGURE 5. CMSD Manufacturing Data Exchange Example (based on Leong et al.,2006 and Bengtsson, et al.,2009).

2.9 ERP-systems and IFS

ERP -system (Enterprise Resource Planning) consists of an ERP software, information technology infrastructure that the software needs to operate and integration of related business processes. A successfully implemented ERP -system can handle all essential functions of business processes and operations efficiently.

ERP -systems have been around since the early 1990’s, and were originally designed to solve the problem of incoherent data systems. ERP was designed to work as data storage by integrating all information to one place. The system was built around a database that would collect and share data from different processes. (Davenport, 1998)

ERP is not just a common central database. The difference between ERP and central database system, CDBS, is that in CDBS data is delivered to central database that then shares the data around. In ERP systems, data is delivered to central database but the different processes can also share data directly among themselves making the information transfer more efficient. (Nah, 2002)

ERP-system foundations can be traced back to manufacturing MRP and MRP II-systems. MRP (Material Resources Planning) systems were designed to manage production system resources and inventory levels. They work as registers for different manufacturing events by collecting information about different production events and delivering that information to required recipients. Ensuring that enough material is available for production and that products are available for customer delivery is one of the most important functions of MRP systems. MRP automatically schedules material requests to meet the production criteria and issue notifications to production planners if there is not going to be enough materials. (Nah, 2002; Leon, 1999)

ERP is a continuation of this development and includes all MRP and MRP II-system features and also includes business processes. ERP -software comprises different modules allowing companies to choose what processes they want to implement their ERP -system on. Modules are made for specific processes and purposes. Some common modules are manufacturing, inventory, maintenance, financial controls & etc. (Davenport, 1998; Nah, 2002)

ERP -systems are not just software suites designed for all business processes of the company but also a way to operate businesses. Successful implementation of an ERP system forces companies to adopt “best practices” that the ERP -system has been designed to help manage. Customer specific customizations are common and possible but usually very expensive. Customizing an ERP -system can also cause problems with communications to other systems such as ERP -systems of other supply chain members. (Davenport, 1998)

ERP -systems are not without weak points. One of them is that data is usually not directly entered in to ERP -system but instead collected from other manufacturing systems. (Moon & Phatak, 2005) Even though many ERP -systems nowadays have built-in MES-modules and are able to receive information from manufacturing environment, it is usually handled by a third party software.

From the scheduling point of view ERP’s weaknesses are in its’ founding principles. It is built upon MRP and MRP2 –systems that were originally created to work as event registers to help in the material resource planning. ERP -systems usually calculate production with unlimited capacity and static resources. This limits the effectiveness of different production schedules and options because most manufacturing systems do not have static resources or unlimited capacity. Real manufacturing systems usually have stochastic variables and capacity is limited with several constraints. Because discrete-event simulation is well suited for taking stochastic variables into account, it would improve the scheduling capabilities of an ERP -system. (Moon & Phatak, 2005; Heilala, et al., 2010)

Production scheduling is an important part of the manufacturing process and there are APS-modules (Advanced Planning and Scheduling) available for most ERP-systems. APS-modules are usually deterministic in nature and based on assumption that working times are constant and that there are no stochastic or unexpected events in the production.

IFS (Industrial and Financial Systems) is a system developer and offers and ERP-system called ”IFS Applications”. As most ERP -systems, it is divided in many different modules that the customers can choose to implement on their manufacturing system. These modules are shown in Figure 6.

”IFS Manufacturing” –module allows production planning, execution, control and analyzation of manufacturing systems. Junttan uses basic IFS manufacturing module.

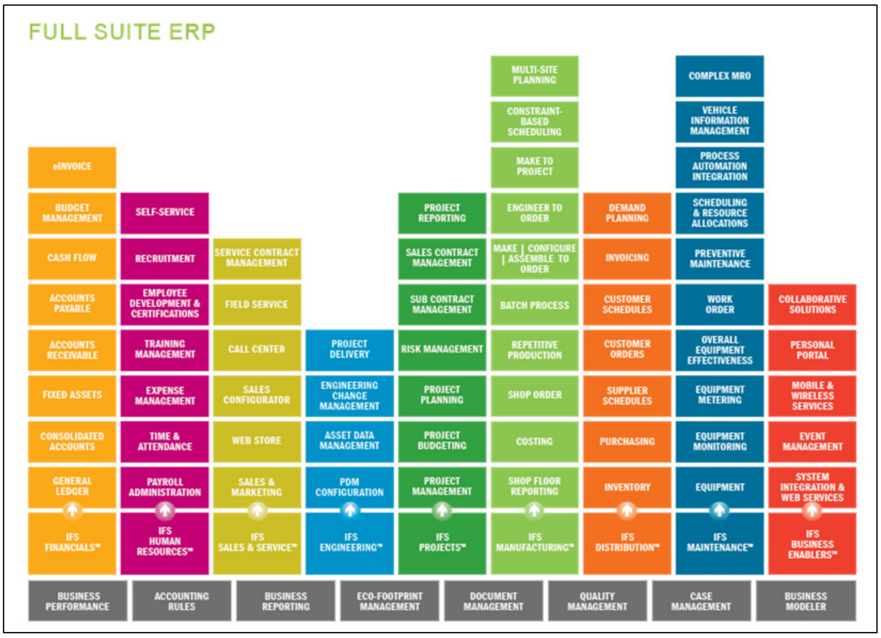


FIGURE 6. IFS modules (IFS, 2011).

2.10 ERP-systems and discrete-even simulation models

Using ERP-systems as data sources for discrete-event simulation models is not a new idea and there are several commercial decision support products that use DES to enhance their scheduling properties.

Programs for discrete-event simulation have been developing in rapid phase during the last 20 years and the development speed doesn't seem to be slowing down. Programs have evolved from hard-to-use, expensive programs made for large companies to moderately priced, easy-to-learn and off-the-shelf programs. DES -programs are currently used mainly for one-time specific purpose, for example as part of layout planning, but there have been many indications and predictions that their use will be expanded to production planning purposes. (Robinson, 2005)

At the same time ERP-systems have become general to a point that middle-size companies are unable to operate without them and even SMEs have been rapidly implementing them.

There are several commercial products using discrete-event simulation available for scheduling and manufacturing decision support. These tools are usually classified as APS products and are developed to fix the deterministic approach usual APS-products and modules have. Examples of such products are Simio® RPS (Risk-Based planning and scheduling), Simul8-Planner and Delfoi Planner®.

Automated data collection for discrete-event simulation models has been a topic of interest since manufacturing information systems became widely adapted. Using DES -models to complement ERP -systems appeared on several publications during the last decade. There have been some articles describing the practical side of connecting ERP –systems to DES -models but the practice is far from commonplace and examples involving industry-scale manufacturing systems are scarce. (Robertson & Perera, 2002; Moon & Phatak, 2005; Johansson, et al., 2007; Heilala, et al., 2010)

Moon & Phatak (2005) describe linking SAP R/3 ERP and PDA (Product Data Acquisition) system to Arena simulation program. PDA system is used to update current shop floor status to the model. They use simplified example called Ides (International demonstration and education system) included in SAP R/3 as their manufacturing system.

Input data harmonization is an essential part of the connection between manufacturing information systems and DES -models. Input data should be transferred in a format that allows for easy implementation of data reading component to simulation model. Johansson & al (2007) have published a test implementation of using a CMSD-specified XML-file as data transfer method between ERP -system and DES -model.

As a conclusion of the literature research, it can be noted that connecting ERP-system to DES -model has been proven possible. There are several ways to accomplish the connection that allow easy updating of input data whenever necessary. However, there is lack of widely accepted neutral data interfaces and formats although there are some standards, such as CMSD, that are gaining popularity. As a result, using of ERP as data source for DES -models is not a common practice.

3 CASE JUNTTAN OY

Junttan Oy was selected as a pilot-case in the simulation project because Junttan's manufacturing environment had recently undergone many changes. In addition to introduction of the new assembly line, the way of manufacturing has changed from job-shop, work cell oriented system to production line-based approach and Junttan had implemented an ERP-system, IFS, in 2008. (Kärkkäinen, 2007)

Starting point for the simulation model's building phase was a M.Sc. thesis prepared for Junttan Oy by Anni Tähtinen in 2008, where processes of an upper car of a pile driving machine were designed, timed and phased for the new assembly line. (Tähtinen, 2007)

Junttan needed a tool to help their production planning and it was important that the tool uses information directly from ERP. Collecting information from the ERP should be easy and as automated as possible. Studying the possibilities and best practices of ERP-DES link was the main goal of the simulation project and this thesis.

The project's goal was to create a practical demonstration defining how DES -model can be used to help production planning and scheduling in Junttan's manufacturing environment. During the project, all possible obstacles that would hinder the use of ERP as data source should be documented, in order to ensure that a production ready decision support tool could be built in the future. The sheer amount of simulation input data, its availability, being up-to-date and the laborious process of feeding it to the simulation model was considered to be the main research problem.

Junttan was not using a separate manufacturing executing system (MES) or any other product for production planning. Unreliable and inaccurate data from the factory floor was considered to be the main risk in the project.

Other risks included suitability of the simulation software. The software was not considered a serious risk because DES -simulation with simulation program is not needed for the decision support system tool. The link to DES-model could be accomplished later with different a simulation software if software compatibility becomes an issue.

The case –project was divided into four steps:

1. Defining the necessary input data.

2. Creating a dynamic DES-model of the manufacturing system.
3. Figuring out the data structures of the ERP-system, collecting necessary input data and designing data interface for automatic information collection.
4. Building the user interface and connecting it to ERP-system and DES-model.

During the project, manufacturing information proved to be insufficient. This meant that using results from the DES-model was no longer considered important for the scheduling tool. The focus was changed to developing a user interface that could also work as a rough production planning and scheduling tool. Data interface for automatic data retrieval from IFS and automatic data transfer to DES -model were built and demonstrated but simulation results were not used in the tool. A more detailed description of choices and results are explained in the results and conclusions sections.

3.1 System components

The operating principle of the system is illustrated in Figure 7. The system is made of three components and information transfer between them.

- Simulation model (3DCreate)
- User interface (DSS-Tool)
- ERP-data source (IFS)

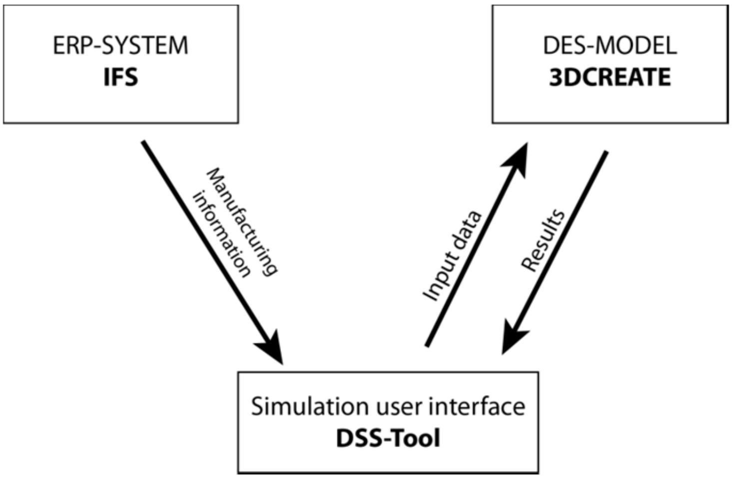


FIGURE 7. The operating principle of the system.

The selected data collection method was a combination of methods two and three by Robertson & Perera (2002), described in more detail in “chapter 2.7, data collection methods for DES-models”. The data is collected from IFS to decision support system tool, DSS-Tool, that also works as user interface for the simulation model. DSS-Tool has an internal database which allows processing large amounts of data. DSS-Tool works as a rough, deterministic production planning and scheduling tool, provides order lead times and allows testing out different production mixes. It does not take into account any stochastic variables, as that was specified to be accomplished by the link to the simulation model. The lack of accurate and detailed input data meant that DES-model was not used with production planning.

In the future, the user will be presented a choice to send the production information to the simulation model which will then return the results back to the program. This leaves the user a choice to either decide that a rough deterministic view is enough or use the simulation model and take into account stochastic variables for more accurate results.

Because of the input data deficiencies, the link to simulation was completed only in demonstration capacity. The data transfer between the user interface and DES-model was completed only in one direction. Results are not returned from the simulation model to the program. The DSS-Tool can be used as source for simulation input data but simulation model results do not enhance the results of the DSS-Tool.

Brief descriptions of different system components and their functions:

- User interface (decision support tool)
    - Loads input data from the ERP-system (IFS).
        - Inventory levels, shop orders (both future and current state of the production).
    - Information that is not available from the ERP-system can be input manually.
        - Resource specific information such as number of resources, shifts & etc.
    - Allows input data modification.
    - Order forecasts can be added to the production mix.
    - Allows rough production planning and selecting production mix for the manufacturing system.
        - Calculates delivery times based on constant work times in work phases, material requisitions, material delivery dates, shifts & etc.
        - Does not take into account stochastic variables or probabilities.
    - Allows opening a data link to simulation model.

    - Open data link makes it possible for simulation model to receive input information directly from the program.
- Simulation model
    - Collects input data from the user interface when needed.
    - Runs simulation model with the input data and parameters coming from user interface.
    - Collects results data during the simulation run.
    - Writes results to external file or sends them back to user interface.
- IFS
    - Works as a data source for user interface / decision support tool

3.2 Product structure and assembly line description

In this chapter product structure and production processes are described in general level.

Junttan had developed a new modular and mass-customizable pile-driving machine model. The new model was introduced to production during the simulation project. It provided good input data to the simulation model because all of the related information was available in electronic form. Compared to some older products, this made it easier to collect and send information to the simulation model. Downside on using the new model was that the factory's flow information about it was scarce.

The following product structure description includes only parts that are visually represented in the simulation model. Only a small fraction of the actual components and work phases are visually represented because visual representations are not necessary for the simulation model operations. Rest of the components and work phases are not visible to the simulation user even though they still use the simulation model's resources and time.

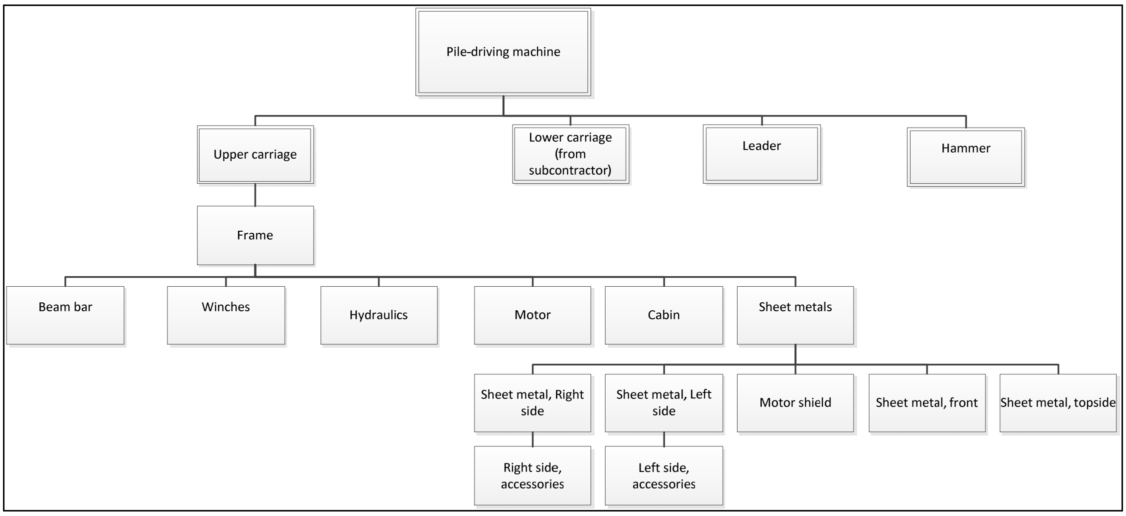


FIGURE 8. Generalized product structure for pile driving machine.

A pile-driving machine can be divided in to four different sections from the assembly viewpoint. These sections are: Upper Carriage, lower carriage, leader and hammer. Many of the parts, such as frame for upper carriage, come directly from subcontractor as JIT-deliveries.

Most components for the pile driving machine require pre-assembling before the actual assembly work phase. Upper carriage modules are pre-assembled beside the assembly work cells. Pre-assembly is completed one phase-step ahead of the assembly.

Assembly line can be thought of consisting of three different segments. The first part, “Upper carriage assembly line” is a phase-timed upper-carriage assembly line for regular module machines. “C-machines” – are upper carriage assembly work cells meant for heavily customized products or prototypes that are not suitable for the assembly line. There are three work cells for C-machines and upper-carriages are assembled in the same work cells from start to finish. Both the assembly-line and “C-machine work cells” provide fully assembled upper carriages to the end of the line.

Assembled upper carriages are lifted to the marriage work cell, where upper carriage is attached to the lower carriage. The machine then moves forward on its own power source. The end of the line includes assembling leader and hammer to the machine. After that, the machine goes to the inspection outside the facility. Final inspection and packaging follow the testing phase.

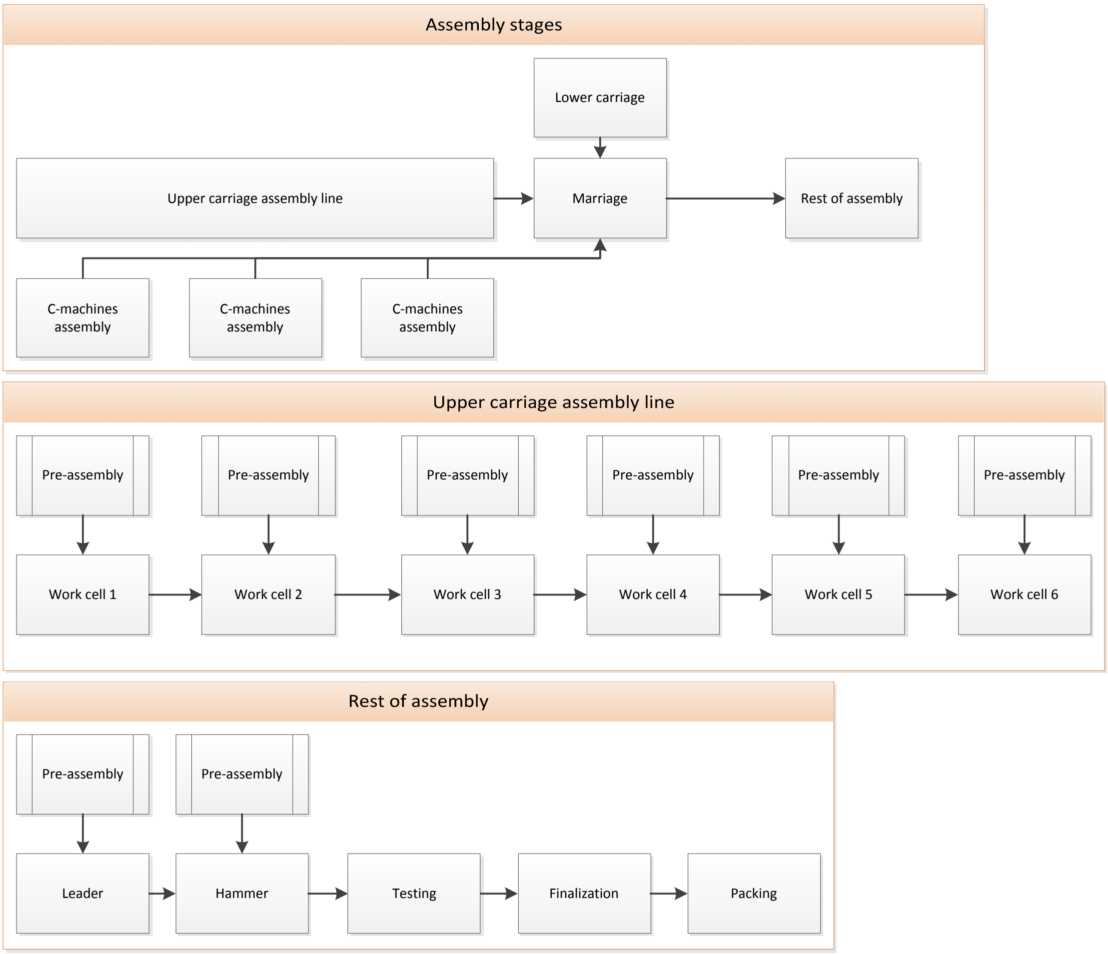


FIGURE 9. Pile-driving machine assembly line.

Upper carriage frame is delivered by the subcontractor as JIT-delivery when needed. First six assembly line work cells handle pre-assembling and assembling the modules to the upper carriage. Lower carriage comes partly assembled from subcontractor as JIT-delivery to marriage-stage.

Leader is pre-assembled from several different parts and assembled in its own work cells. Because pre-assembling leader is the longest work phase in the process, there are several pre-assemblies going on at the same time.

Hammers are pre-assembled and assembled on the next work cell that is the last assembly work cell in the manufacturing system. Hammers are also sold separately as maintenance parts and there can be several pre-assembly and assembly works going on the same time, depending on the number of orders.

A more detailed description of different work phases can be found in the following section.

3.3 Creating the 3D-Simulation model

Discrete-event simulation model was built with a 3DCreate simulation program. 3DCreate was selected as the simulation program because of its easy user interface and technical features that were considered beneficial for the project. Some of these features were natural 3D-environment, good data interfaces for developers (COM API) and direct Solidworks compliance.

Junttan’s development pace required that both the simulation model and the decision support tool were designed to be flexible towards future changes. Building a flexible simulation model and trying to anticipate future needs is more difficult and slower than creating one with static requirements from an existing system. The simulation model has to be able to process different products that require different amount of resources without having to reconfigure the model manually.

The simulation model was verified and validated to a reasonable level during the model building phase by the simulation project team. The project team included subject-matter experts from Junttan’s production planning department and simulation practitioners from Savonia. The results of the simulation run could not be compared to results from a real system because there was no such data available yet.

3DCreate has two basic types of components that can both be customized to include required behaviors:

- **Static** components are created in the simulation layout before the simulation run. Examples of static components could be work cells, machines and other resources.
- **Dynamic** components are created during the simulation run, and destroyed during the run or at the end of the simulation run when simulation world is reset. Products and other parts in the material flow are examples of dynamic components.

The difference between the two is that static components are not destroyed during simulation runs. Dynamic components are created during simulation run and must be always positioned in components having “Container” –property or they are instantly destroyed. Otherwise the functions and behaviors between dynamic and static components can be identical.

3DCreate COM API -interface allows good compatibility with .Net –programming environment and fluent real-time connectivity between the simulation program and user interface. The disadvantages of the program are that it’s slightly unstable with large and complex simulation models. 3DCreate also has very limited components for resource modeling requiring extensive programming for any behaviors that are more complex than simple pick-and-place logics. The lack of preprogrammed resource modeling options affected the inclusion of human resources on the model.

3DCreate clearly needs a better logic for controlling human resources and error handling/tracing. Many of the error conditions can be avoided with safe coding practices but the program has a lot of instable elements that can cause instant crashes without any warning to users. Visual Components is continuously developing the program and many of the error instances encountered have already been fixed before this thesis was published. However, these deficiencies and instant crashes slowed down the model building step in the simulation project.

The direct Solidworks compliance of the software is an add-on-feature and did not work as well as anticipated. Junttan provided the project team with part structures of the module-machine as Solidworks parts. Because Solidworks parts were directly from Junttan’s PDM-system they were very detailed and large (> 1 GB) models that required lots of computing power. Using production scale 3D-models from the PDM-system was not expected to work fluently as the geometries are large and impractical to be used with DES-models.

One part of the simulation program introduction was finding out the limits of the simulation programs when dealing with heavy graphics because detailed graphics are often used with 3D-simulation models that have marketing uses. Using Solidworks models with little modifications would shorten the time required for simulation model building. This in mind, the project’s starting period concentrated on finding out a good solution to how much production geometries would need to be changed, or “lightened”, to work with the simulation program.

Heavy geometries were lightened with Solidworks and then imported to the 3DCreate. Even lightened geometries were significantly more complicated than the primitive geometric features of 3DCreate and many of them caused memory problems. Some memory problems can be explained by the fact that 3DCreate is a 32-bit program which limits the use of memory. However, there are no safeguards to prevent program from crashing from excessive memory use which meant that the maximum level of detail was found on trial-and-error basis.

The problem with the geometries was solved using the same method that has been used previously with detailed simulation models. The most important simulation components were built with two sets of geometries, one set for simulation runs and one for visualization purposes. During the simulation runs, the visual representation has little value and heavy geometries slow the model down considerably. Simple, blockified geometries allow speedy simulation runs and result collecting. For visualization purposes the geometries can be switched to realistic and detailed graphics that, although not practical in the real production runs work well for visualization purposes. This was easily accomplished with 3DCreate's built-in functions that allow parametric geometry changes.

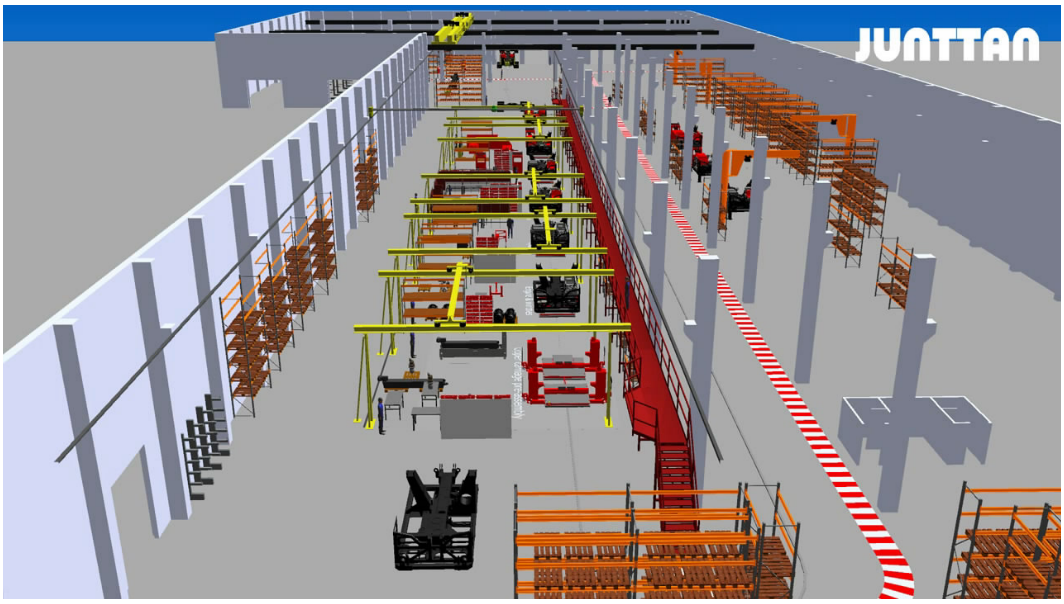


FIGURE 10. Simulation model of the assembly line.

The data link between simulation model and user interface must be initiated by the simulation user from the user interface once 3DCreate is opened and simulation model loaded. This can be automated in the future by starting the simulation program automatically with a correct model from the user interface.

When simulation model is started, it sends a handshake -signal to user interface to indicate that it's ready to receive information. The signal also works as an order to send the basic order list information to the simulation model.

Based on the basic information received from the user interface, simulation program can create event lists for the following future event types:

- Work lists for the assembly work cells including estimated starting times (for both the assembly line work cells and c-machine work cells).
- Required starting times for the preassembly work cells.
- Shift schedules for the active work cells.
- List of available resources and resource-related restrictions.

Preassembly work cells are defined as independent, single work cells because they can have unique shift, load and resource arrangements. Individual work cells for C-machine assemblies use same preassembly work cells as assembly line.

Simulation clock is run until the first event in assembly or pre-assembly event lists is reached. The model reads the general order information from the event list and asks user interface for more detailed information. Detailed information is then attached to the part component and can be accessed during the simulation run from the visible geometry of the product. Preassembly information is sent directly to the correct pre-assembly work cell.

Simulation model uses Junttan's future, modular product and related files for geometries. Simulation model receives information about work phases and the geometry files that are used to visualize them. This allows changing products and geometry files and attaching them to different work phases without any changes to the simulation model.

Creating a manufacturing system simulation model requires simplifications on those parts of the system that are not considered essential. 3D-simulation models require visual simplifications as well as behavioral ones because complex geometries are not necessary for the simulation user to get a sense of the manufacturing system. Only geometry-files that are required to visualize important work phases are used. Most work phases are invisible to the user even if they require simulation resources and time. Simulation user may see visual indications about them, for example, with the use of resources for certain work phases.

Components have work related information attached to them as a parameter that assembly work cells read when component enters the work cell. In preassembly work cells, the work list is attached to the components when they arrive to the work cell. After completing the work phases work cells write results information to the part. Results information includes timestamps on part arrival, resource needs and completion times. Results can be collected and used to find bottlenecks and product throughput times at the end of the simulation run.

Assembly line has been setup up to work in takt time. All assembly work cells have to perform their operations in time defined by the takt time. Even if there are delays or problems in the work cell, the upper carriage will move to the next station. If the problems are critical, such as a critical component with long delivery time is missing, upper carriage can be moved aside from the assembly line. Small problems and delays can be fixed in the next work cell. Simulation model follows a generalized approach that marks the work phase as failed and continues with the next work phase. Reason for this approach is that serious problems with the workstations are rare and taking them into account would be complicated and provide no useful results.

Upper carriage frame comes from the subcontractor and is delivered directly to the beginning of the assembly line. Cleaning and inspecting frame is the first work phase before frame enters assembly line. Frame is lifted to the first assembling work cell by crane once inspection is completed.

Most work cell operations/work phases are not visible to the simulation user. They are all included as a dynamic list that will be delivered to the work cell in question. Dynamic list includes all requirements for the process in question and allows easy upgrading of product structures and work phases.

Simulation model needs virtual processes in addition to the real work processes to successfully emulate the real system. Virtual processes are usually only required in the simulation model although they can be used to emulate some manufacturing environment behaviors. Virtual processes that are used only for visual purposes take no simulation time. Examples of such virtual processes are all visual changes to component geometries. Some virtual processes, such as resource requests for crane take both simulation time and resource use. If resource in question is reserved for something else, the process has to wait until resource is available.

First work cell receives upper carriage frame from the cleaning and inspection and it is placed upon hydraulic runners. Runners are needed because the work cell includes some components that are attached underneath the frame. The assembly work cell has preassembled components waiting for the assembly process. Preassemblies have been completed in preassembly work cells located at the side of the assembly work cell. Preassemblies are completed one phase/takt time before assembly so that the components required for assembly are always available. After all the assembly processes have been completed in the work cell, an agv-vehicle is ordered and upper-carriage is lowered upon it. Only visible work phases in the work cell are attaching the reversion gear and a beam that is installed inside the frame. Agv-vehicle moves the upper-carriage to the next work cell for the next takt time.

Engine and winches are attached in the work cell two. Winch count and type vary depending on the product. There are two preassembly work cells for winches and one for the engine. Preassembled components are waiting for the assembly on the floor.

Work cell three handles hydraulics unit assembly. In the real manufacturing system some of the hydraulics processes are completed outside the work cell. Assembly process was simplified to count all outside processes as one work cell stage. Outside processes use same resources and outside locations are not used for any other purposes.

Work cell four assembles most of the electrical wiring and electronics packages. The only visible work process is attaching the upper shield for the engine.

Work cell five assembles and attaches the cabin and a swing for the leader. Cabins are preassembled by the subcontractor and arrive complete except for the electronics. Preassembly processes for work cell five include connecting electrical wiring and electronic modules to the cabin.

Work cell six includes attaching the outer shell and related parts to the carriage. Visualized components include all sheet metal parts, counterbalance block and stabilizators. After completing work cell processes, upper-carriage is ready to be connected to the lower carriage in the marriage work cell. Upper-carriage is lifted from the agv and moved to the marriage work cell allowing agv to move back to the assembly line starting point.

Work cell seven handles the marriage between upper- and lower carriages. Lower carriage arrives from the subcontractor, and is preassembled beside work cell. Preassembly includes assembling the hydraulic components and other missing parts. Lower carriage moves on tracks, and after the marriage the carriage can drive forwards with its own tracks.

Individual assembling work cells for C-machines complete the assemblage of upper-carriage in the same work cell from start to finish. Preassemblies for leader and hammer that are needed for machines assembled in these work cells are timed according to their own schedules, separate from the regular product assembly line.

Work cell eight assembles and attaches the Leader, longest component in the pile driving machine. Preassembling leader is by far the longest process in the assembly line meaning there are several (1-4) preassembly work cells serving the assembling process. Visual parts for the assembling work cell are different modules for the leader and the head of the leader, called a cock.

Work cell nine assembles and attaches hammers to the leader. Hammers are also produced as service parts meaning their order list include more than just the assembly line products. There are several preassembly work cells serving the assembling.

Work cell ten is the testing site outside the factory. Pile-driving machine is tested for the first time for any irregularities.

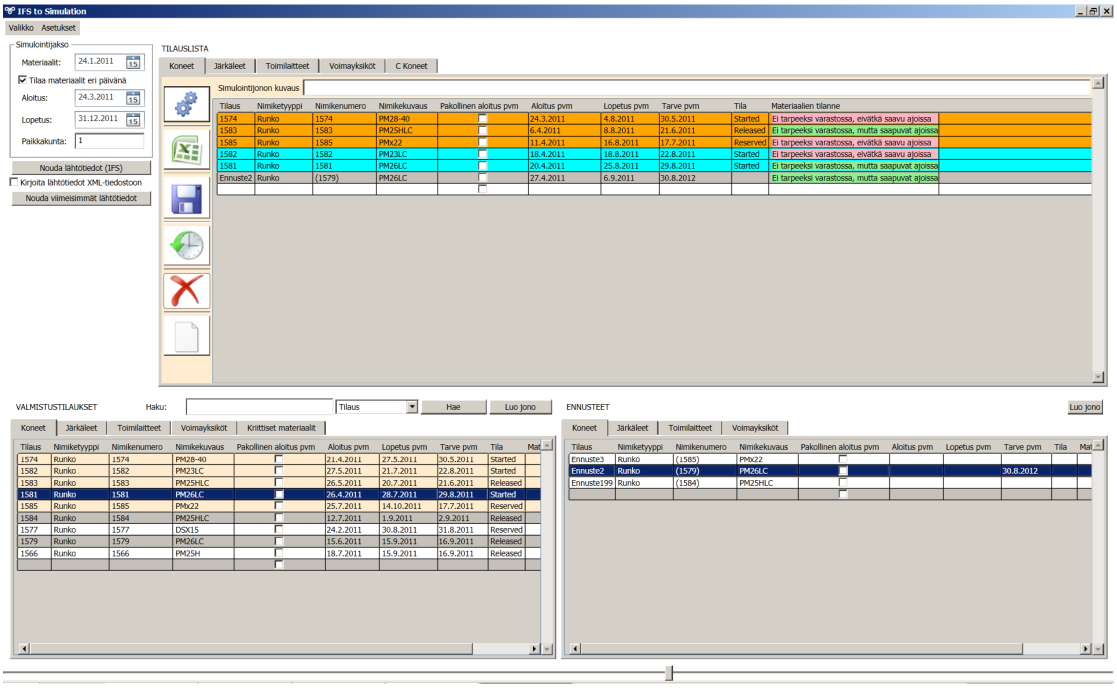
Work cell eleven includes product and test result inspection and **work cell twelve** is the final work cell. Final work cell includes disassembling and packing the pile-driving machine for the delivery.

3.4 User interface / Decision support system

User interface, UI, was originally designed for the simulation input data collection, modification and transfer. Simulation project focus changed during the project and user interface was developed to a decision support system tool, DSS-Tool. User interface's front page is shown in Figure 11.

Several people were involved in UI/DSS-Tool building during the different simulation project stages. Changing program objectives from simple user interface to decision support system tool did not affect data transfer interfaces but increased the complexity and required amount of functionalities. Program architecture for the DSS-Tool is illustrated in Figure 12.

A customized version of the DSS-Tool was completed as part of Bachelor of Engineering Thesis in the autumn 2012 by Pasi Heiskanen. DSS-Tool and the database structure are described in more detail in that thesis work.



Materiaalit: 24.1.2011
☑ Tilaa materiaalit eri päivänä
Aloitus: 24.3.2011
Lopetus: 31.12.2011
Paikkakunta: 1
☐ Kirjoita lähtötiedot XML-tiedostoon

Simulointijonon kuvaus:

Tilaus	Nimiketyyppi	Nimikenumero	Nimikekuvaus	Pakollinen aloitus pvm	Aloitus pvm	Lopetus pvm	Tarve pvm	Tila	Materiaalien tilanne
1574	Runko	1574	PM28-40	☐	24.3.2011	4.8.2011	30.5.2011	Started	Ei tarpeeksi varastossa, eivätkä saavu ajoissa
1583	Runko	1583	PM25HLC	☐	6.4.2011	8.8.2011	21.6.2011	Released	Ei tarpeeksi varastossa, mutta saapuvat ajoissa
1585	Runko	1585	PMx22	☐	11.4.2011	16.8.2011	17.7.2011	Reserved	Ei tarpeeksi varastossa, eivätkä saavu ajoissa
1582	Runko	1582	PM23LC	☐	18.4.2011	18.8.2011	22.8.2011	Started	Ei tarpeeksi varastossa, eivätkä saavu ajoissa
1581	Runko	1581	PM26LC	☐	20.4.2011	25.8.2011	29.8.2011	Started	Ei tarpeeksi varastossa, mutta saapuvat ajoissa
Ennuste2	Runko	(1579)	PM26LC	☐	27.4.2011	6.9.2011	30.8.2012		Ei tarpeeksi varastossa, mutta saapuvat ajoissa
				☐					

Haku: Tilaus

Tilaus	Nimiketyyppi	Nimikenumero	Nimikekuvaus	Pakollinen aloitus pvm	Aloitus pvm	Lopetus pvm	Tarve pvm	Tila	Mat
1574	Runko	1574	PM28-40	☐	21.4.2011	27.5.2011	30.5.2011	Started	
1582	Runko	1582	PM23LC	☐	27.5.2011	21.7.2011	22.8.2011	Started	
1583	Runko	1583	PM25HLC	☐	26.5.2011	20.7.2011	21.6.2011	Released	
1581	Runko	1581	PM26LC	☐	26.4.2011	28.7.2011	29.8.2011	Started	
1585	Runko	1585	PMx22	☐	25.7.2011	14.10.2011	17.7.2011	Reserved	
1584	Runko	1584	PM25HLC	☐	12.7.2011	1.9.2011	2.9.2011	Released	
1577	Runko	1577	DSX15	☐	24.2.2011	30.8.2011	31.8.2011	Reserved	
1579	Runko	1579	PM26LC	☐	15.6.2011	15.9.2011	16.9.2011	Released	
1566	Runko	1566	PM25H	☐	18.7.2011	15.9.2011	16.9.2011	Released	
				☐					

Tilaus	Nimiketyyppi	Nimikenumero	Nimikekuvaus	Pakollinen aloitus pvm	Aloitus pvm	Lopetus pvm	Tarve pvm	Tila	Mat
Ennuste3	Runko	(1585)	PMx22	☐					
Ennuste2	Runko	(1579)	PM26LC	☐			30.8.2012		
Ennuste199	Runko	(1584)	PM25HLC	☐					
				☐					

FIGURE 11. Front page of the user interface (Heiskanen, 2012).

Information flow was designed to work only from IFS to user interface. ERP-system is a critical system for a company and limiting the data transfer to work only in one way was seen as an easy way to reduce risks. Data connection back to IFS was not considered essential since program was designed to help in the scheduling and decision making, not actually making the decisions. Later versions can include the option to send the production mix back to ERP automatically.

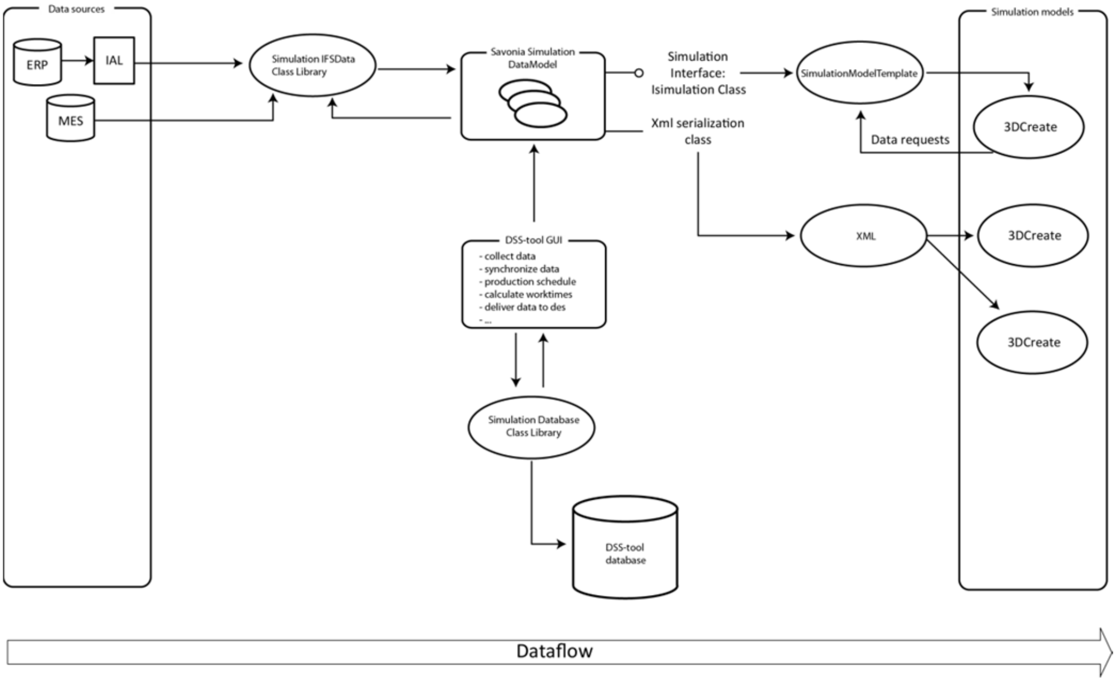


FIGURE 12. DSS-Tool program architecture.

DSS-Tool provides rough production planning information for the assembly line and fulfills the current needs that Junttan has for the production scheduling. The program is deterministic and does not take into account any stochastic variables. It calculates the customer's order throughput times by taking into account process times, shifts, inventory levels and material requisition times. It also shows if there are enough materials for the order to be completed. If there are not enough parts in the inventory, it calculates if the primary supplier can deliver the needed parts in time.

All information needed for a simulation model is not usually available in the ERP-system. Reasons for this can vary – information might not be automatically or manually collected or it can be collected to different information systems. Junttan is not currently using any MES-system or module that would collect information directly from the factory floor. It also uses several different information systems for various aspects of manufacturing environment.

DSS-Tool allows input and modification of the following input data:

- Human resources/workers.
    - Available resources can be assigned for work cells and processes individually, or for work cell groups.
- Shifts and working times.

    - Shifts and working times can be changed for specific work cells or resource classes.
    - Different parts of assembly line can work in different shifts. Assembly line and C-machine assemblies are an example of different shifts.
- Order lists for different product classes.
    - Four basic order types were defined as: pile-driving machines, hammers, power packs and accessories.
    - DSS-Tool collects the current order the order list from the IFS. After rearranging the order list throughput times must be calculated again.
- Simulation run period.
    - Start- and end time for simulation period.
        - Defines the orders that are collected from IFS.
        - Defines the time window for material requisitions and inventory levels collection.
        - Defines the time window for shift information collection.
- Virtual processes.
    - Includes processes that are needed for the simulation model to perform correctly. Some of the virtual processes are connected to real processes that define when virtual process is used. Examples of virtual processes are visualization changes, resource requests & etc.
- Production forecasts.
    - DSS-Tool allows creating forecasts for different products by using existing products and product structures as template. Product characteristics such as process times and required resources can be modified after creating a template from an existing product.
- Inventory levels.
    - Inventory levels are collected from the IFS using only primary vendor's delivery times. Program allows user to manually change delivery time to indicate use of secondary vendor(s).
    - Program also takes into account inventory levels on different moments of time. IFS schedules material requisitions for the current production during MRP-runs. Using that information allows taking into account all material requisitions related to current orders.

DSS-Tool does not need to update all the information from IFS every time simulation is run. Part of the information can be used several times in different simulation runs and only part needs to be refreshed every time. This was the main reason why data collection from IFS was not created as an automatic feature. The user can refresh the data manually by pressing a button.

DSS-Tool uses Microsoft SQL Express relational database for data storage. It allows quick data-related operations despite large amounts of data. Using database for data management also makes program development easier.

Using information from the ERP-system ensures that data is up to date. DSS-Tool allows production planners to test different production mixes and customer orders with the existing production. Current state of the production combined with forecasted products give production personnel better grasp on how the production would be affected with new orders. Different production scenarios can be tried out by changing production resources and material availability information.

DSS-Tool shows the created order list and all related restrictions visually. Different colors are used to indicate production possibilities and shortages. If the order is not completed in simulation time, the reason is shown to the user. If there is not enough time to complete production, the background for the order is shown red. If there is part shortage, order background is orange unless the part can be supplied in time. The resulted production schedule can be exported to Microsoft Excel for visualization purposes as shown in Figure 13. Assemblies and subassemblies are indicated with different colors.

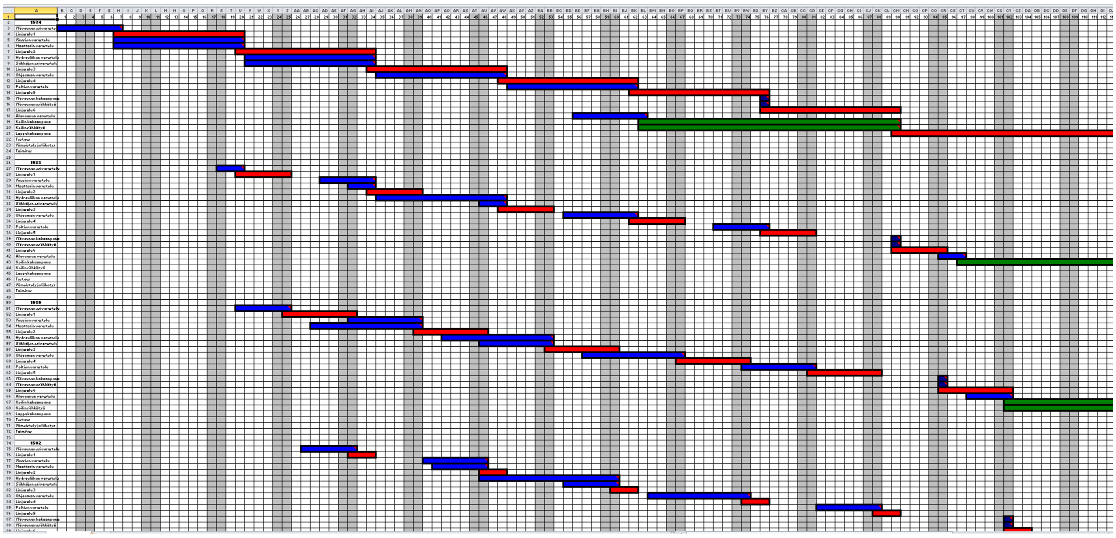


FIGURE 13. Production schedule can be exported to excel.

Because data transfer is one way from IFS to DSS-Tool, users need to input simulation results manually to IFS when suitable scheduling solutions are found. The DSS-

Tool allows users to save the order list to program database for later use or verification purposes. Changes to the information received from IFS are also saved and can be printed out.

3.5 Data transfer interface between DSS-Tool and simulation model

Data transfer between DSS-Tool and simulation model was created with two different methods. The first method is a direct connection between DSS-Tool and simulation model and uses 3DCreate COM/API –interface. The second method uses external XML-files.

Direct connection was chosen as primary data transfer method because of the large amount of input data. The data interface in the DSS-Tool has been abstracted to individual module to allow future modifications without any impact on the functionalities of the DSS-Tool. Keeping data interface module separate makes it easier to use different simulation models with the same user interface, or use different user interface with the simulation model.

Using direct connection method does not put any restrictions to input data amount as necessary data can be transferred to simulation model only when needed. External data files as simulation input data have been used previously in Savonia. For some models the amount of input data can be very large, causing peaks in memory use of the simulation models. In addition to Savonia’s own experiences, available examples about using CMSD-specified input files as data transfer method pointed out that large amount of initial data must be taken into account. (Johansson, et al., 2007)

Second data transfer method uses external XML-files. The initial version uses simple, custom made data structure with a minimum amount of attributes and identification tags. There are good examples on how to use CMSD-specified XML-files but because there was not enough time to get familiar with the CMSD-standard it was selected as a data format that will be used in the future.

Off-the-shelf simulation programs usually allow building of procedures to read external files into the model. This was one the main reasons why external files were chosen as secondary data transfer method. XML was selected because it is an open-source and very common data format that is also easy to read without any programs.

Simulation model is directly connected to data interface that delivers the information to the DSS-Tool. Data interface uses 3DCreate COM-API “Component Listener” function to listen directly to a data transfer component in the simulation model. Simulation model uses predefined commands to request simulation input data from the DSS-tool.

Benefits of the direct connection are automatic data transfer, no external files and it requires no additional actions from the user. Using external files allows using the input data separately from the user interface. Files are also easy to share via email and the same input data can be used with several different simulation models at the same time.

3.6 Data transfer interface between DSS-Tool and IFS

IFS data interfaces possibilities were researched during project meetings and with the help of IFS online documents. Some project meetings also included specialists from IFS. Information can be collected from IFS using three different methods:

a) Direct SQL -queries to database views and tables in IFS
b) Using IAL-objects, SQL queries to IAL-objects in IFS
c) External files exported from IFS

Research into pros and cons of the three methods resulted in the following information.

Direct SQL-queries to data view and tables in IFS is a well working and traditional way of doing the data collection.

- The problem with the method is that information is located in many different views and tables. IFS database structure has over 5000 tables and views to choose from.
- To be able to access all the different data storage location requires user account rights equivalent of an administrator or manually assigning the account rights to all data locations.
- Granting out administrator rights to the program would not be a good security practice. Assigning access separately to all locations would require a lot of work from administrators and any updates or changes in the system would be complicated.

External files would be an even more difficult solution to implement and would most likely require customization from the software provider. The main difficulty of this method is pretty much the same as with direct data queries – the fact that data is spread on so many different locations and there is no built-in way to export that data easily. It would require customization from the software provider.

Using IAL-objects was chosen as the best solution from the three options. IAL-objects are custom objects within Information Access Layer (IAL) that provide easy data access to IFS tables and views. It was also a method supported by the Junttan's IFS administrators. The main reason for selecting this method is explained in more detail in the following explanation of IFS's IAL objects.

3.7 Using IAL-object in IFS

IAL (Information Access Layer) is a good way to collect data from the IFS database. IAL-objects offer an easy and secure way to retrieve data and can be easily controlled and implemented.

IAL-objects work in the same way as normal database views allowing the possibility to collect data from different tables, views and API-commands.
The main difference between IAL-objects and database tables and views is the management of user rights. Instead of giving user access to all the different data views and tables, administrators create IAL-object that has access to those data sources. IAL-layer has all the necessary rights to access information within IFS so users who have access to IAL-object, have automatically access to all information that is accessed via it.

Easy management of user rights does not mean that using IAL-objects is without challenge. Creating an IAL-object in IFS has the same challenge as creating a direct SQL-query. There are lots of tables and views that can be used as data sources and persons creating the object need to know where to get the information they need. Same data can be found from several different sources (tables, views) and some of the sources depend on the language settings. Language settings use the "core data" and change it according to selected languages. It's important to know the location of the "core data" where the data is retained in the original form.

The working logic of Information Access Layer and IAL-objects is illustrated in Figure 14.

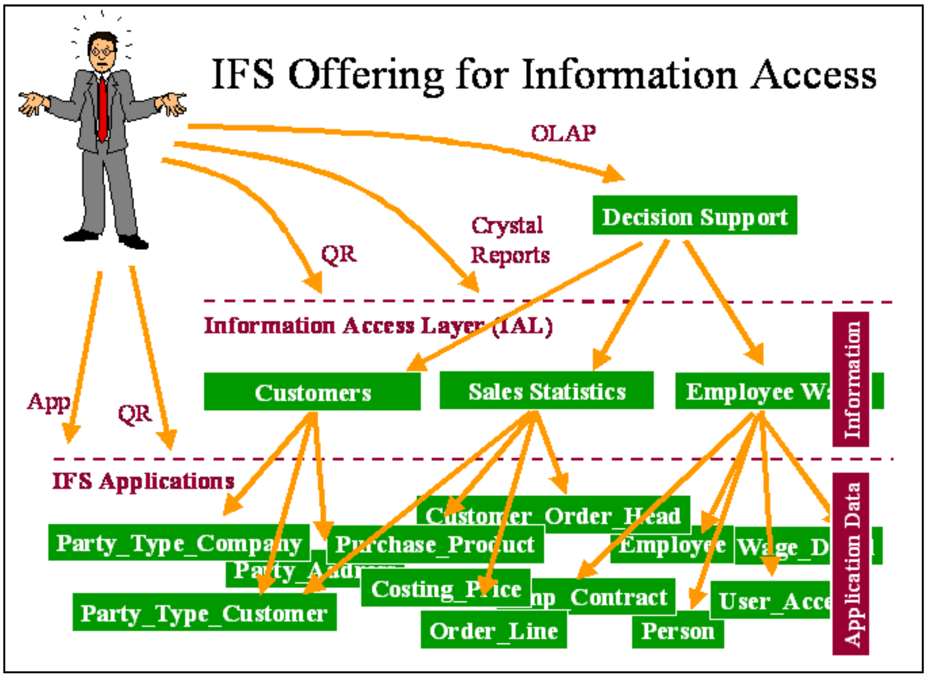


FIGURE 14. IAL Layer and IAL-objects (IFS, 2011).

Using IAL-objects makes it possible to change the source of the data in the IAL-objects later without making any changes to the DSS-Tool. SQL-queries that collect data from IAL-objects are created in the DSS-Tool and there is no need to modify them unless the IAL-object field names change. The ability to easily change data location is useful in case better sources are found for the data.

Granting access rights to IAL-objects is an easy task. The only rights that have to be granted are the connection rights to IFS and accessing the IAL-object in question. When IAL-objects are created, no user group has access to them automatically.

According to IFS online help documentation, IFS offers two types of IAL-objects (IFS)

1. "Live data" IAL-objects, which means the object acts as a linking service pointing to the correct data. No actual data is stored in the IAL-object meaning the data is always up-to-date. The problem with live data method is that more specific queries are not possible. IAL-object points to all records of that particular data and it must be narrowed down with SQL-queries to the object.
2. Table-based IAL-object. A new data-table is created for the IAL-object and it is refreshed in certain periods. This allows data to be stored to the IAL-object and in case of large and often used software, lessens the load on the IFS-system. It also allows using more specific queries narrowing down the data so

the amount of data in the table can be limited. This makes SQL-queries to the table lighter as the collection process is already done with the system.

DSS-Tool uses “live data” IAL-objects for two reasons:

- DSS-Tool is not necessarily used every day so it is important that the tool or any related procedures related to it do not place unnecessary burdens to IFS system.
- The data for the DSS-Tool should be as recent as possible. This requirement combined with the first reason makes “live data” type object a better choice.

IAL-objects were defined as general as possible to allow including different products from different factories/sites without having to redefine the objects. A more detailed description of IAL-objects can be found from IFS online help documentation.

“IFS IAL-objektien käyttö”, a short introduction to using IAL-objects was created during the project. It includes step-by-step documentation on how to create IAL-objects with examples of all IAL-objects used in the DSS-Tool.

4 RESULTS

This thesis and the simulation project resulted in a working demonstration on how to use IFS ERP-system as data source for Decision Support System and discrete-event simulation models in a manufacturing industry environment. Savonia can use the research results in the future to offer more valuable DES-models to customers.

The 3DCreate simulation software was used to build a dynamic 3D-Simulation model that allows effortless changes to the assembly line products and related processes. Also, changes in the simulation model layout are possible without reprogramming the whole model.

During the simulation project, methods for using 3D-geometries directly from 3D-Cad programs such as Solidworks were studied, but the results were not encouraging even though the simulation program has native Solidworks support. Ready 3D-Geometries make the graphical design of the model building easier, but still require a lot of work until they are light enough for simulation model purposes.

Lack of accurate input data caused the original focus of the simulation project to be changed during the project. The precision of the assembly line information was too vague for the simulation model to be any more accurate than deterministic calculations. Because of this, IFS data was used for deterministic production planning calculations instead of utilizing simulation results. Simulation model collects typical manufacturing information, such as throughput times, work cell and resource utilization during the simulation runs but because of the poor input data accuracy, DSS-Tool is not using the simulation results.

Simulation run results can be exported from the simulation model to external file. Due to the dynamic nature of the simulation user interface, the same framework can be used later on different simulation models that allow testing the connection back from the simulation model to the DSS-Tool.

User interface for the simulation model was created with Microsoft .Net/C# programming environment. Simulation user interface evolved in a decision support system, DSS-Tool, which includes all the necessary functions for modifying simulation model input data. It can also be used as a rough production scheduling and planning tool with or without the simulation model. DSS-Tool was built around relational database and parts of the program were abstracted to allow its use on other similar manufacturing systems with minimal changes to the program logic. DSS-Tool was built as a “proof of concept” presentation that was not intended for actual production use. A customized version of DSS-Tool was made as part of Bachelor’s Thesis. Documentation of the program structure and a user manual were also produced.

A connection between User Interface (DSS-Tool) and ERP-system was created with the help of IFS Information Access Layer and IAL-objects. IAL-objects allow collecting data from numerous tables and views to single object that can be accessed easily. Documentation on how to create and use IAL-objects with detailed examples was created to ease the use of IAL-objects in the future.

DSS-Tool includes two different methods to send input data to simulation model. The first method was “direct link” through 3DCreate COM API-interface. Direct link interface was abstracted in a way that allows changes to DSS-Tool or simulation model without affecting other side of the connection.

The second method was exporting the initial data to XML-files. CMSD-specification was researched and selected as a future format for the XML-file. Simulation user has an option to choose either one of the data transfer methods from the DSS-Tool. Both methods have their pros and cons that have to be weighed depending on the end use.

Input data can be sent to simulation model by either method but no data is returned to the DSS-Tool. Working data interfaces allow future development when data collection from the factory floor becomes more accurate. More accurate data collection from the factory floor requires customization to Junttan’s IFS -system because work phases were reported as total time of the work cell instead of individual work phases.

The DSS-Tool works as a deterministic decision support tool and provides Junttan the information they need for production planning. Junttan was satisfied with the customized version of the DSS-Tool although the information accuracy was not as good as what the DES-model could have provided.

5 CONCLUSIONS

The main goal for this thesis was to study how to use ERP-systems as a data source for discrete-event simulation model input data. Collecting and entering input data for DES-models is a slow and laborious process that could be improved with a data link to the manufacturing information systems. Even though automatic data collection for simulation models could extend the operational life of DES-models significantly, practical, industry scale examples about ERP to DES connections are scarce.

Additional goals were introducing new DES-program (3DCreate), finding out its limits and creating a decision support tool for Junttan's assembly line. Both goals were successfully completed, although the simulation model is not used to enhance the production planning capabilities.

Connecting manufacturing information systems such as ERP to DES models is actively studied area in the simulation field. During the project, several articles and conference proceedings were published about the matter, including a VTT report (Heilala, et al., 2010) that is used in this thesis. The main difference between those articles is the used data interface. Many of the recently published articles use CMSD-based XML-files instead of direct data link to the DES-model.

During the project, perhaps the most common problem with simulation models surfaced – inaccuracy of the input data. Data from the factory floor is not collected in enough detail to provide the simulation model with accurate input data. Currently the work phases are collected at the work cell level meaning specific working times in preassemblies or assemblies are not available. Junttan has made separate time-measures for rough knowledge of the assembling work phases but there was not enough information to create statistical distributions for the simulation model.

The way Junttan currently uses IFS does not allow collecting detailed working times for work phases unless additional features are built to IFS. Collecting more accurate work phase information would also require customization of the IFS-system. Junttan continues developing the manufacturing processes and the factory floor information is expected to get more accurate in the future. The possibility of using a simulation model in the future was ensured by building two interfaces for data transfer between the DSS-Tool and simulation model.

The lack of proper input data means discrete-event simulation does not provide any additional benefits that could not be attained by deterministic calculations of the work phases. When the problem surfaced, the project team concentrated on developing the simulation user interface to a rough production planning tool to support decision making. This did not require additional work because the necessary information was already collected to user interface via the link to the ERP-system.

The end result was a DSS-Tool that can work both as a deterministic, rough production planning tool and a simulation user interface. The accuracy of production planning is directly proportional to the accuracy of the input data. DSS-Tool was also developed further to include forecasted products to the order mix. The process of testing out forecast products in the production in IFS had been a tedious process in the past. DSS-Tool allows testing out the forecasts with the same information IFS has and finding out if they should be input into IFS.

XML-file export/import function used in the DSS-Tool is not currently based on the CMSD-standard. The main reason for this was the limited resources and time available for the project. CMSD-standard would be a logical choice for simulation input data files because it has been developed specifically for that purpose and there are several examples on its use. DSS-Tool's XML-file export/import function can be modified to CMSD-specifications with little effort if the program development continues.

There are many different end goals for the input data harmonization efforts. Many researchers concentrate on trying to automating the whole model building process based on the input data while the practical need is to input data to existing models.

Savonia has been creating discrete-event simulation models for local industry for the past twelve years. All simulation projects have involved spending a lot of time in the factory floor figuring out how production works. There is a lot of "silent information" on the factory floor that is not input in any manufacturing information systems. The only way to access it is to discuss the specifics of manufacturing environments with the people involved in the actual production.

Creating DES-models in this kind of environment will never be accomplished automatically just based on the information available on different information systems. The situation is also not likely to change fast because it is part of the current operating culture in this industry. Building up automatic simulation models could be possible on manufacturing systems that are very automatic by nature but even in those cases the benefit from actually building the model from the input data is not that beneficiary. The main problem in the input data management of DES-models is to feed current information to the model once it has been built.

Using discrete-event simulation models has a potential to enhance the decision making process and allows more realistic production planning and scheduling. There are several commercial products available that use DES-models for such purpose and the thesis proved that it is also possible to use off-the-shelf simulation program to create similar decision support tools.

During the project, there was a lot of interest towards DSS-Tool in the local industry, especially in the local SMEs. Problem is that local SMEs do not usually use IFS that is priced for larger companies. For the future development, there should be a data interface to the common ERP-system used by the SME companies in the area. LemonSoft ERP is an increasingly common ERP-system in the area and Savonia already has research partners using it on real production. Because many of the problems were documented during the simulation project, a data link on the ERP-side would not be very complicated and could be a logical step in the DSS-Tool development. The main problem will remain the same as with the IFS – knowing the correct location of the data in the ERP database and finding out the best solution to collect it.

As a result of this thesis and the simulation project one may conclude that ERP-system is an excellent and proven data source for the simulation model but the following crucial issues should be taken into account:

a) It is most likely that some of the information for the simulation model comes outside the ERP-system as many companies still use several different systems for different functions. Some information may not be readily available in any information system.
b) If a MES-system is used to collect more accurate data from the factory floor it should also be added as a data source to simulation model
c) Using information from ERP, MES or any other information system requires data editing because data is stored in different formats. Some of this process can be automated but those parts that cannot, require user interface that is easy to use. Simulation models do not usually offer good tools for building customized user interfaces.

REFERENCES

Banks, J., Carson II, J. S., Nelson, B. L. & Nicol, D. M., 2005. *Discrete-event system simulation, fourth edition.* s.l.:Prentice Hall.

Bengtsson, N. et al., 2009. *Input data management methodology for discrete event simulation.* s.l., Proceedings of the 2009 Winter Simulation Conference.

Boulonne, A., Johansson, B., Skoogh, A. & Aufenanger, M., 2010. *Simulation Data Architecture For Sustainable Development.* s.l., Proceedings of the 2010 Winter Simulation Conference.

Davenport, T. H., 1998. Putting the Enterprise into the Enterprise System. *Harvard Business Review,* Issue July-August.

Heilala, J., Montonen, J., Järvinen, P. & Kivikunnas, S., 2010. *Decision Support Using Simulation for Customer-Driven Manufacturing System Design and Operations Planning,* s.l.: VTT.

Heilala, J. et al., 2010. *Developing simulation-based decision support systems for customer-driven manufacturing operation planning.* s.l., Proceedings of the 2010 Winter Simulation Conference.

Heiskanen, P. J., 2012. *Tuotannon kuormituksen simulointijärjestelmä,* s.l.: Savonia University of Applied Sciences.

Hietikko, E., 2008. *DigiBranch projektisuunnitelma,* Kuopio: Savonia ammattikorkeakoulu.

Hietikko, E. & Suhonen, A., 2010. *DigiBranch-hankkeen väliraportti - Digital Tools for Product Development,* s.l.: Savonia.

IFS, 2011. *IFS Applications - Products.* [Online]
Available at: http://www.ifsworld.com/en-na/product/
[Accessed 2 20 2011].

IFS, 2011. *IFS Applications 7.5 Online Documentation.* [Online]
[Accessed 2011].

Johansson, M. et al., 2007. *A test implementation of the core manufacturing simulation data specification.* s.l., Proceedings of the 2007 Winter Simulation Conference.

Junttan Oy, 2012. *Junttan Products.* [Online]
Available at: http://www.junttan.com/products
[Accessed 22 4 2012].

Kärkkäinen, K., 2007. *Suunnittelu- ja toimeenpanojärjestelmän määrittely ja valinta Junttan Oy:lle,* s.l.: Lappeenrannan Teknillinen Yliopisto.

Kivikunnas, S. & Heilala, J., 2011. *Tuotantosimuloinnin tietointegraatio, standardikatsaus,* s.l.: VTT.

Law, A. M., 2007. *Simulation Modeling and Analysis, Fourth edition.* s.l.:McGraw-Hill.

Law, A. M., 2009. *How to build valid and credible simulation models.* s.l., Proceedings of the 2009 Winter Simulation Conference.

Leon, A., 1999. *Enterprise Resource Planning.* s.l.:The McGraw-Hill.

Leong, S., Lee, T. Y. & Riddick, F., 2006. *A Core Manufacturing Simulation Data Information Model for Manufacturing Applications.* s.l., Systems Interoperability Standards Organization.

Leong, S., Lee, T. Y. & Riddick, F., 2006. *A Core Manufacturing Simulation Data Information Model for Manufacturing Applications.* s.l., In Proceedings of the Systems Interoperability Standards Organization 2006 Fall Simulation Interoperability Workshop. Paper 06F-SIW-028.

Moon, Y. B. & Phatak, D., 2005. Enhancing ERP system's functionality with discrete event simulation. *Industrial Management & Data Systems,* 105(9), pp. 1206-1224.

Nah, F. F.-H., 2002. *Enterprise Resource Planning Solutions and Management.* s.l.:IRM Press.

Pooch, U. W. & Wall, J. A., 1993. *Discrete event simulation, a practical approach.* s.l.:CRC Press, Inc..

Pyysalo, J., 2010. *WMD4222 Agile Production: course notes.* Varkaus: Savonia University of Applied Sciences.

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

Robinson, S., 2005. Discrete-event simulation: from the pioneers to the present, what next?. *Journal of the Operational Reserach Society,* pp. 619-629.

Sargent, R. G., 2011. *Verification and validation of simulation models.* s.l., Proceedings of the 2011 Winter Simulation Conference.

Shim, J. P. et al., 2002. Part, present and future of decision support technolgy. *Decision Support Systems,* Issue 33, pp. 111-126.

SISO-STD-0xx-2011, 21.3.2011. *Standard for Core Manufacturing Simulation Data - XML Representation.* s.l.:s.n.

Skoogh, A., Michaloski, J. & Bengtsson, N., 2010. *Towards continuously updated simulation models: combining automated raw data collection and automated data processing.* s.l., Winter Simulation Conference.

Tähtinen, A., 2007. *Vaiheistetun kokoonpanon suunnitteleminen uuteen tehtaaseen,* s.l.: Tampereen Teknillinen Yliopisto.

W3C, 2011. *World Wide Web Consortium (W3C) XML Essentials.* [Online]
Available at: http://www.w3.org/standards/xml/core.html
[Accessed 14 10 2011].

White, P. K. & Ingalls, R. G., 2009. *Introduction to Simulation.* s.l., Proceedings of the 2009 Winter Simulation Conference.