

# Graduation Report

Improving ESoT's user Interface



7 June 2012  
Vanderlande Industries  
Nikolas Dandy



**GRADUATION REPORT**

**FONTYS UNIVERSITY OF APPLIED SCIENCES**

**HBO-ICT: English Stream**

<b>Data student:</b>	
Family name , initials:	Dandy, N.D
Student number:	2136282
project period: (from – till)	6 February 2012 – 22 June 2012
<b>Data company:</b>	
Name company/institution:	Vanderlande Industries
Department:	Technical Information Systems
Address:	Vanderlandelaan 2 5466RB, Veghel, Noord-Brabant, The Netherlands
<b>Company tutor:</b>	
Family name, initials:	Jochen Cobussen
Position:	Project Leader of Technical Information Systems
<b>University tutor:</b>	
Family name , initials:	Boots, P.B
<b>Final report:</b>	
Title:	Improving ESoT Model Browser
Date:	7 June 2012

Approved and signed by the company tutor:

Date:

Signature:



## Preface

My name is Nikolas Dandy, I am taking a bachelor degree program at Fontys University of Applied Sciences Eindhoven. As 4<sup>th</sup> year student Information & Communication Technology (ICT), student is required to organize an internship within a company. The internship is at least 95 working days which in my case is from February until June 2012. This graduation report reflects the process I went through during my graduation internship at Vanderlande Industries. It explains how I managed to accomplish the assignment.

This project has been initiated by Technical Information Systems (TIS) department. The assignment is to improve the component called ESoT Model Browser in ESoT application. My task is to develop a new ESoT Model Browser component that gives the right information and possibility to make a change. The new component must meet all the functional and non-functional requirements of the users.

This opportunity to work in one of the big companies in Netherlands is a priceless experience for me. I would like to express my deepest gratitude to Mr. Jochen Cobussen as Manager of this department for his guidance throughout my graduation internship period. Despite his busy schedule, he managed to spare some time to guide and teach me in getting acquaintance to the company environment and the assignment.

I would like also to say thank you to Mr. Peter Boots as school supervisor for his support and feedbacks about the structure, content and English grammar of my reports. With his support, it helps to ensure the quality of the reports.

Other persons I would like to say thank you for helping me is Mr. Peter Kamps, Mr. Kevin Pluk and Mr. Peter van den Brule for their technical advices about this new component.

I won't forget this amazing experience!

June 7, 2012

Nikolas Dandy



## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>3</b>
<b>2</b>	<b>The Company.....</b>	<b>4</b>
<b>2.1</b>	<b>Leading in baggage handling systems .....</b>	<b>4</b>
<b>2.2</b>	<b>Integrator of fully automated systems for parcel and postal sorting centers .....</b>	<b>4</b>
<b>2.3</b>	<b>Automated logistics systems for warehouses and distribution centers.....</b>	<b>4</b>
<b>2.4</b>	<b>Organization Structure .....</b>	<b>5</b>
	2.4.1 Project Structure .....	6
<b>3</b>	<b>The Project.....</b>	<b>7</b>
<b>3.1</b>	<b>ESoT .....</b>	<b>7</b>
<b>3.2</b>	<b>Current ESoT Model Browser .....</b>	<b>9</b>
<b>3.3</b>	<b>Project Description.....</b>	<b>10</b>
<b>3.4</b>	<b>Methodology .....</b>	<b>12</b>
<b>3.5</b>	<b>Project Tools and Environment .....</b>	<b>12</b>
<b>4</b>	<b>Project Phasing .....</b>	<b>13</b>
<b>4.1</b>	<b>Phase 1: Initiative phase.....</b>	<b>13</b>
<b>4.2</b>	<b>Phase 2: Definition and Design phase .....</b>	<b>13</b>
	4.2.1 The First version .....	13
	4.2.2 The Second version .....	14
	4.2.3 The Final and Initial version .....	15
<b>4.3</b>	<b>Phase 3: Implementation phase.....</b>	<b>16</b>
<b>4.4</b>	<b>Phase 4: Delivery phase .....</b>	<b>17</b>
<b>5</b>	<b>The New Component.....</b>	<b>18</b>
<b>5.1</b>	<b>The New ESoT Model Browser.....</b>	<b>18</b>
<b>5.2</b>	<b>Event Message.....</b>	<b>19</b>
	5.2.1 Tree view selected message.....	19
	5.2.2 Combo box selected message.....	19
	5.2.3 Combo box value changed message.....	20
<b>5.3</b>	<b>Submit the Values .....</b>	<b>21</b>
<b>5.4</b>	<b>Problems and Solutions .....</b>	<b>22</b>
<b>6</b>	<b>Result.....</b>	<b>23</b>
	<b>Conclusion and Recommendation .....</b>	<b>24</b>
	<b>Evaluation.....</b>	<b>25</b>
	<b>Referance .....</b>	<b>26</b>
	<b>Appendix I: Project Plan .....</b>	<b>27</b>
	<b>Appendix II: Final Version Requirement Document.....</b>	<b>39</b>
	<b>Appendix III: Initial Design Requirement Document.....</b>	<b>41</b>



## Summary

This report describes the progress and result of graduation project at Vanderlande Industries, Department Technical Information Systems. This project was carried out by Nikolas Dandy, a fourth-year ICT student from Fontys University of Applied Sciences, Eindhoven.

Vanderlande Industries has a department called Technical Information Systems. This department has jobs to develop software that are used in the company in order to improve the quality of Vanderlande's products. One of the software is Engineering Suite of Tools (ESoT). The job of this software is to support layouters with designing a material handling system using AutoCAD. ESoT consists of several components and there is one component that handles misbehavior of the system, and it is called ESoT Model Browser. The current version of ESoT Model Browser is dissatisfying for the reason that it gives abstract information, and does not allow users to make changes.

Therefore, the main goal of this project is to develop the new ESoT Model Browser that has good understandability for an expert user in advance for normal users, and allows users to make changes. The data that is given is based on XML libraries which store all the rules of the system.

In order to plan and track this project, the methodology that is used is agile methodology. This project has several cycles, and every cycle includes initiating, defining, designing, realization, testing, and transferring. The initiating, defining, and designing phases are done simultaneously where the project with its requirements is initiated, the functionalities are defined, and several versions of the prototype are designed. These phases are followed by realization, testing, and ended in the transferring phase where all deliverables were delivered to all concerning parties. The designing phase is followed directly by the defining phase and went back to the designing again in an iterative loop until the design was over. In every iterative stage, there were some progress meetings, problem discussions, and progress demonstrations then the project plan and requirement documents are updated. The realization phase covers the implementation of the design and programming the new ESoT Model Browser. The testing phase covers the test of the component in order to check its quality. The last phase is the transfer phase. In this phase, the new ESoT Model Browser and all documentation were handed over to the company.

The research and analysis were gathered through company supervisor's guidance. All the results from the research and analysis were applied in this project and reflected in this report. The ESoT and the ESoT Model Browser were built in C++ using Visual Studio 2010. ESoT consists of several modules that communicate to AutoCAD by using XML which is also used by ESoT Model Browser.

The assignment was successfully accomplished, even though there were several adjustments from the plan. The company was pleased with the result. The knowledge and experiences obtained during the project and development process make the project valuable. Not only educational experiences, but also organizational culture and project management.



## Glossary

Agile	Is a methodology of developing a software.
AutoCAD <sup>[1]</sup>	is a software application for computer-aided design (CAD) and drafting.
ESoT	Engineering Suite of Tools - an application that developed by Technical Information Systems department in Vanderlande Industries.
ESoT Model Browser	Is a component inside ESoT that handle the violation constraints.
XML	eXtensible Markup Language



## 1 INTRODUCTION

This report describes all the activities and the final result that delivered while doing internship in Vanderlande Industries. The project is about redesign the ESoT Model Browser. ESoT Model browser is one of the components in ESoT (Engineering Suite of Tools), and ESoT is an application that is developed by Technical Information Systems department. The main goal of ESoT is to helps user in this case the Company's employees to design a material handling system, furthermore it can be implemented in real situation. ESoT is developed in C++ environment and has two tools that run as plug-in in AutoCAD(the Data Editor and Graphical Editor), also ESoT consists of several modules that communicate to AutoCAD by using XML.

Main features of ESoT are working although there are some features need to be improved or fixed and added new features. One of the features that need to be improved is ESoT Model Browser. ESoT Model Browser is the component that handles the misbehavior of a baggage system and should give accessible information. However the current situation the ESoT Model Browser is not give a sufficient information and not allow user to make changes. It makes user confuse instead, because the information is given in form of a complex programming equation that even hard for a programmer to understand. This is becoming the main reason why the company initiated this project. The company wants ESoT Model Browser to be able to give right information that easy to understand and user can change a parameter into a valid value.

In the Chapter 2, it describes about detail of the company and the organization structure.

Chapter 3 describes about the background, current situation, objectives, methodology and tools that are used of the project.

Chapter 4 explains the phases of the project and the approach of problems, also the detail about the research that had been done during this project.

Chapter 5 describes about detail of new ESoT Model Browser and the improvements that are implemented and how they work also the problem solving of the project.

Chapter 6 describes the result that the intern delivered during this internship

Followed by, conclusion and recommendation of this project and company, and evaluation of the intern during this period of internship at Vanderlande Industries.

Last but not least, the appendixes of the documents that was needed and important during or as result of the internship. These documents are: Project plan and requirement documents.



## 2 THE COMPANY

Vanderlande Industries is dedicated to improving its customers' business processes and competitive position by providing automated material handling systems and services.

Focus is to improve Vanderlande Industries customers' logistics processes and increase their logistics performance today, tomorrow and throughout the entire life cycle.

Vanderlande Industries systems and associated services enable fast, reliable, labor-saving goods handling in distribution and parcel and postal sorting facilities, as well as for baggage handling at airports.

### 2.1 LEADING IN BAGGAGE HANDLING SYSTEMS

From check-in to aircraft hold, from arriving flight to reclaim carousel. On-time every time, at the lowest possible cost and above all, with the best performance, because every bag is Vanderlande Industries' business.

At Vanderlande Industries, they're dedicated to smooth handling of baggage, day in and day out. Through the busiest daily and seasonal peaks, enabling customer to address increasing passenger numbers and baggage volumes as well as changing traffic flows, security regulations and airport layouts. Making baggage handling an efficient part of your operations.

Vanderlande Industries designs, builds and services leading baggage handling systems for airports of all sizes. This belt, tub and/or track solutions combine operational effectiveness, short connection times and high convey-ability together with effective integral control of your baggage operation. Based on proven technology, in-depth business knowledge and industry best-practices, Vanderlande Industries deliver the highest availability, reliability and lowest costs per bag. That's what it's about.

### 2.2 INTEGRATOR OF FULLY AUTOMATED SYSTEMS FOR PARCEL AND POSTAL SORTING CENTERS

- The most cost-efficient solution
- The best handling of your customers' products
- High performance and reliability
- Maximum transparency of internal processes and product flows

Vanderlande Industries provides today's widest range of automated parcel and postal sorting solutions for hubs and depots of all sizes. Vanderlande Industries focus on customer's entire door-to-door process from arrival and unloading, sorting up to loading and departure.

As well as the necessary state-of-the-art sorting technology, Vanderlande Industries have a strong focus on providing real-time process and system intelligence using IT solutions. This enables customer to maximize the efficiency of customer's sorting process and continuously improve your operations.

### 2.3 AUTOMATED LOGISTICS SYSTEMS FOR WAREHOUSES AND DISTRIBUTION CENTERS

Vanderlande Industries is a leading supplier of integrated logistics systems for automation of your warehouse/distribution centre. An automated warehouse/distribution centre gives you:

- Order picking productivity up to 1,000 lines / man hour
- Order picking accuracy up to 99.99%



- Triple throughput per m2 in your warehouse
- 15 minutes learning curve of new operators

Vanderlande Industries focus on customer's entire goods flow from goods receiving, storage, order picking up to shipping and the related information flow. Vanderlande Industries' automated logistics systems for warehouses and distribution centers include:

- Warehouse Management Systems (WMS) / Warehouse Control Systems (WCS)
- Order picking / order fulfillment systems
- Automated storage and retrieval systems (AS/RS)
- Sorting systems
- Conveyor systems / internal transport systems

Vanderlande Industries is among the top suppliers of these solutions in the world, with a track record of automation of over 1,000 warehouses and distribution centers in recent years. These are found in a wide range of industries such as food, non-food, footwear, apparel, parts & components, automotive, pharmacy, 3PL and e-commerce.

## 2.4 ORGANIZATION STRUCTURE

The Technical Information Systems (TIS) Department performs software improvement and it is where the project is executed. This department consists of 13 people who are expert in software programming. In many cases, this involves also the development of new concepts and development of tools or systems. This figure below is the organization structure of Vanderlande Industries:

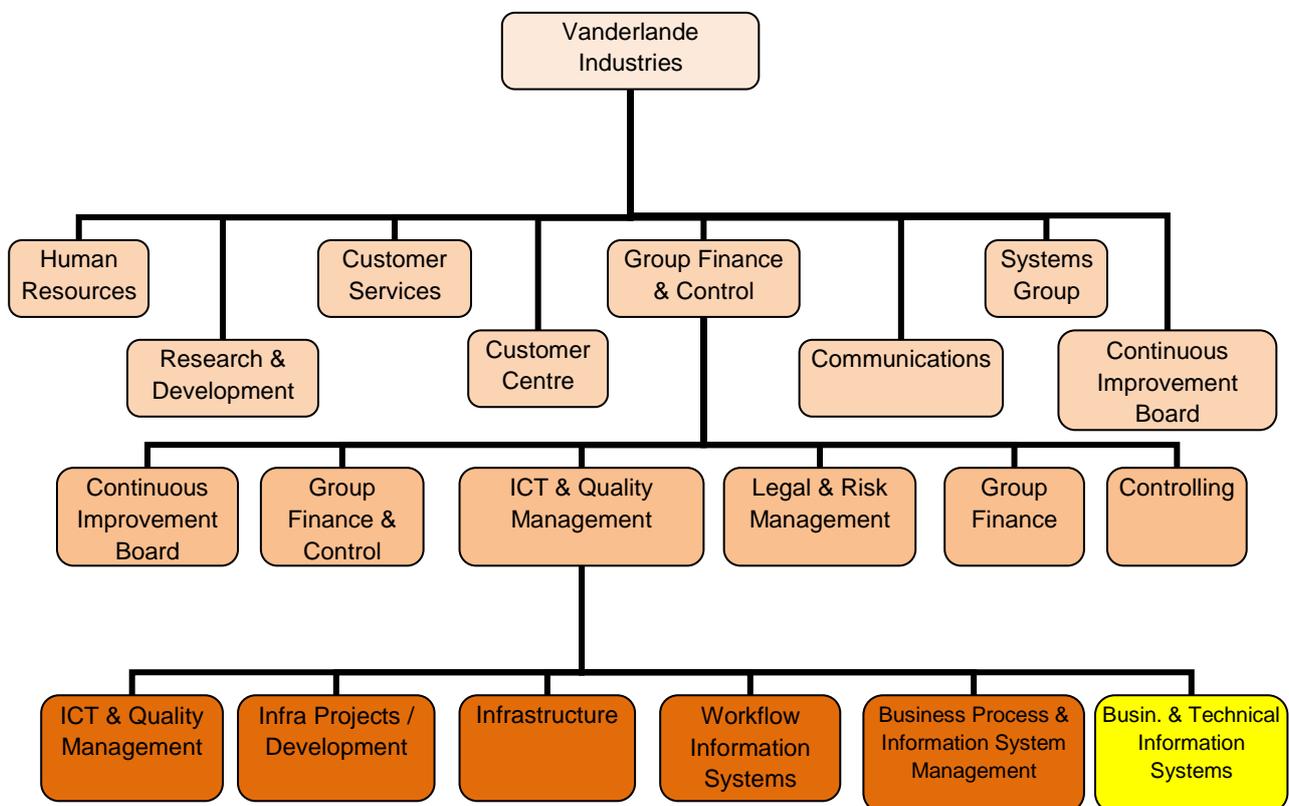


Figure 2.1: Organization structure.



### 2.4.1 Project Structure

This is the structure organization for this project. This organization was divided into 4 roles:

1. Company Manager: Supervise intern's works and liaises with the intern to ensure that the solution meets those needs in terms of quality, functionality and ease of use/maintenance.
2. Company Supervisors: Responsible for the quality of the products delivered by the Developer. The Company Supervisor has authority to commit or acquire resources on behalf of the Company.
3. School Supervisor: The School Mentor is to supervise and ensure project developer is working on the right track and delivers as written in the document plan (e.g. Project Plan).
4. Intern: Has the authority to run the project on a day-to-day basis within the guidance of company manager, supervisors and school supervisor, subject to the tolerances set within this document.

The details:

The Company	Vanderlande Industries
The Company Manager	Jochen Cobussen Senior IT Specialist Email: jochen.cobussen@vanderlande.com
The Company Supervisors	Peter Kamps Senior IT Architect Email: peter.kamps@vanderlande.com  Kevin Pluk IT Specialist II Email: kevin.pluk@vanderlande.com
The School Supervisor	Peter Boots Medewerker Fontys Hogeschool, Eindhoven Email: p.boots@fontys.nl
The Intern	Nikolas Dandy Student in ICT/Software Engineering Fontys Hogeschool, Eindhoven Email: nicolas_dandy@yahoo.com



### 3 THE PROJECT

This chapter describes the project. This chapter contains the detail about ESoT, ESoT Model Browser, the project description, methodology and the tools and environments that are used in this project.

#### 3.1 ESOT

Vanderlande Industries' baggage system has a lot of components such as check-in counter, conveyer-belt, baggage scanner and etc. Each of components has parameter (specification) for example a conveyer-belt has codeID, length, type of engine, type of material, speed and etc. The data need to be as accurate as possible to prevent mistake in ordering and building the component; and redesign the system that can increase the amount of cost and time. Therefore, the Company developed software to help doing this.

Engineering Suite of Tools or ESoT is software developed by Technical Information Systems (TIS) department. ESoT is developed using C++ language. ESoT allows its user to design and build a transport system within a 3d environment (AutoCAD). The software improves the engineering process and the quality of its deliverables.

The ESoT's user interface has two main parts which are also the plugs-in in AutoCAD; Data Editor and Graphical Editor (see: figure 3.1 the bottom area with dark green background). Data Editor is the part where the parameters, properties, quantities of a component are shown, and Graphical Editor is the part where the design of a component is shown. However, behind these two parts there is a part that called Engine. All the intelligent commands are executed by Engine and the result is sent to Data Editor and Graphical Editor as a XML form. Data and Graphical Editor will sent a Window Message to Engine if there is a event occur. Engine has input not only from Data Editor and Graphical but also Component Library and System Definition in XML format. These library files store all the data regarding the components within its specification. Basically, the library files are the input, then Engine is where the input is executed, and Data Editor and Graphical Editor are the output from the Engine.

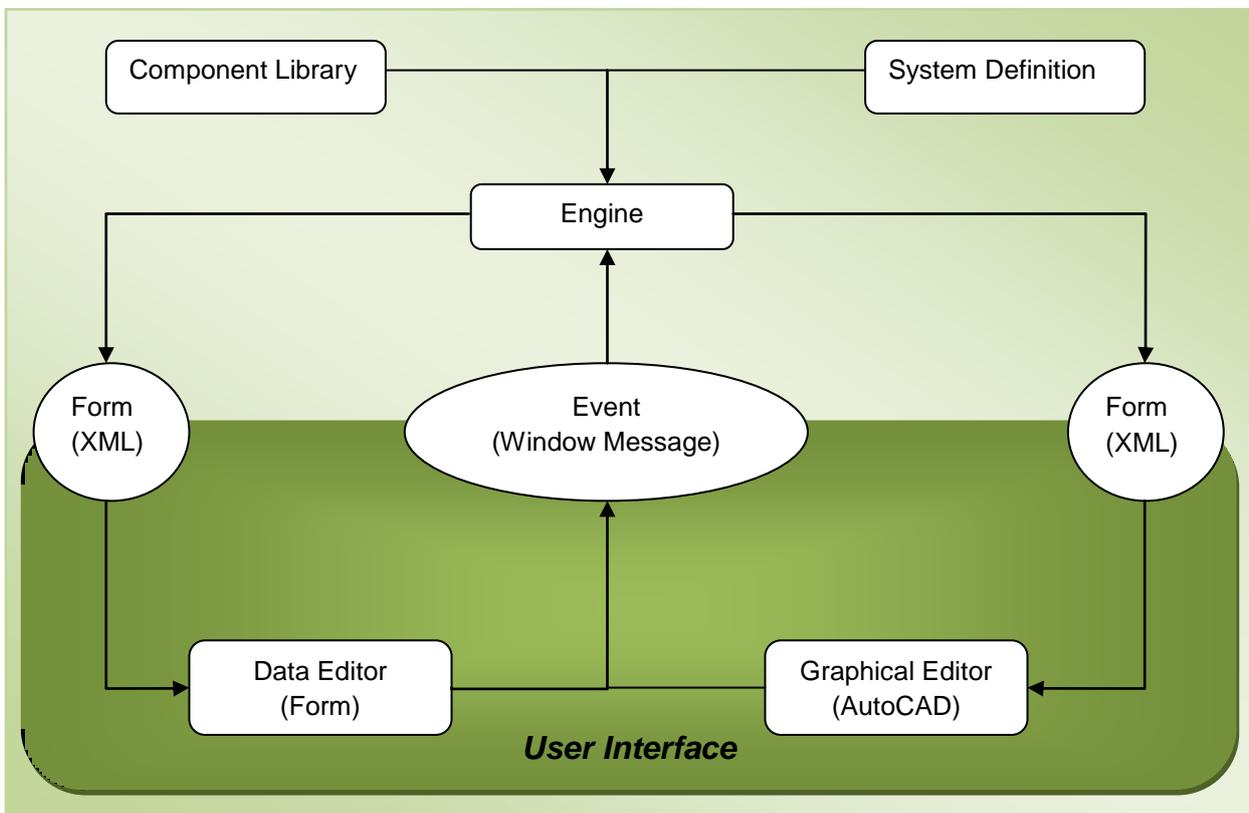


Figure 3.1: ESoT's internal Process flow

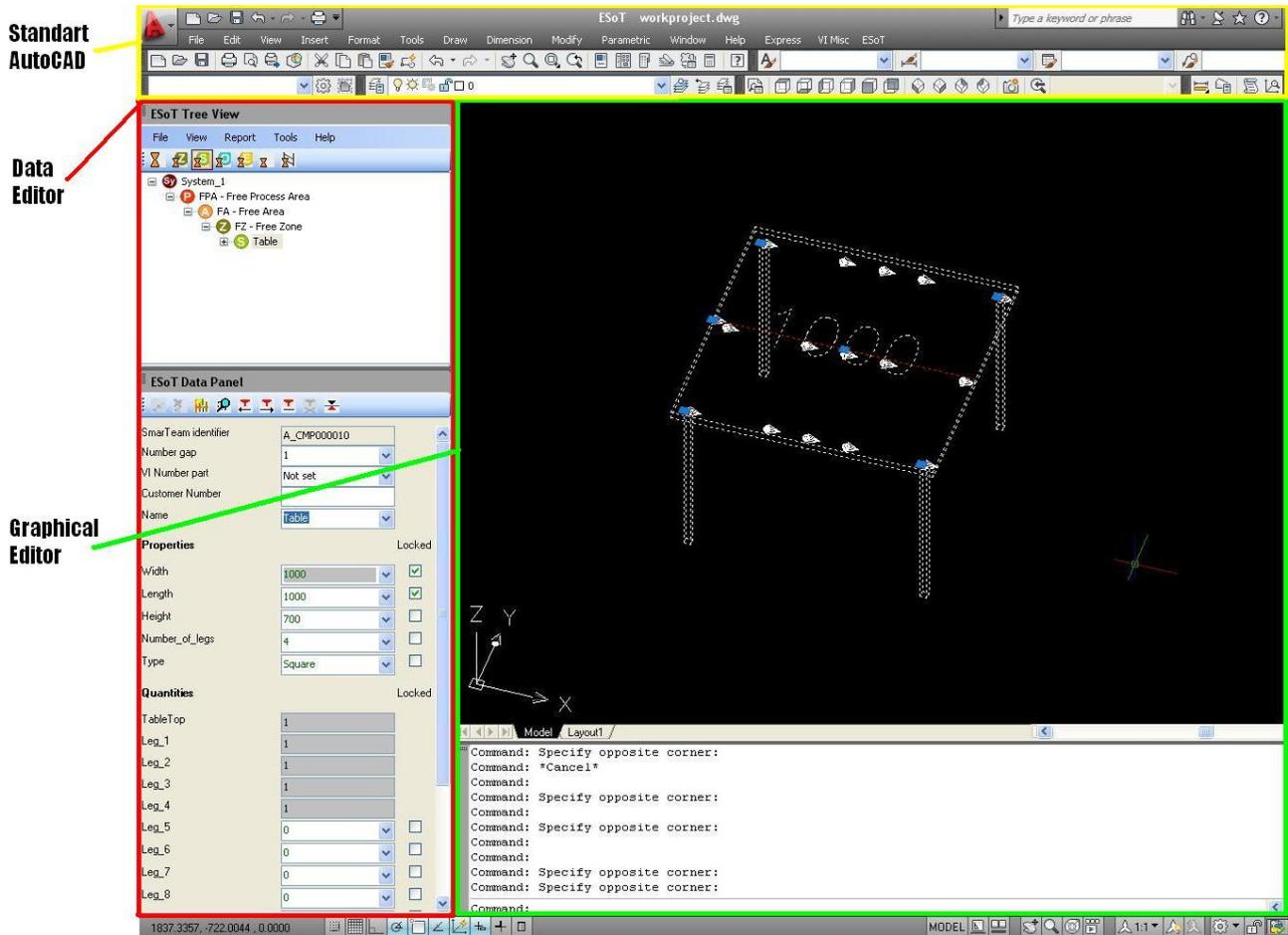


Figure 3.2: User Interface of ESoT

In figure 3.2 we can see at left side is the Data Editor and on right side is Graphical Editor. Data Editor is divided into two parts; tree view and data panel. The tree view displays level of component. The data panel displays name, value and is-locked of the selected-component's parameters. Inside the data panel user can change the value and is-locked of parameters. User cannot change the parameter's value that is locked.

List of parameters is divided into three parts:

- General attribute – common for all component,
- Properties – to read and set the component configurations,
- Quantities – to read and set the amount of subcomponents.



### 3.2 CURRENT ESOT MODEL BROWSER

In ESoT, there is a component called a *constraint*. A constraint is a rule that consist of one or more components. A brief explanation will use a table as example. To build a square table, first determines the size of the table by specify the width, height and length. Using the measurements, the amount of legs can be determined. A table has at least 4 legs to support it. In large table, it can have 6, 8, or 12 legs. All legs need to have same height in order to make a stable table. When one leg has different height than others, it violated a constraint. This will trigger ESoT Model Browser and tell a user that all legs should have same height. ESoT has hundreds of components and constraints, and these are made base on real situation. ESoT Model Browser should be able to handle all the violated constraints.

ESoT model browser will be triggered when a user put a value that violates one or more constraints. It shows all the violated parameters and constraints as a warning to a user. In this window, a user should be able to understand why the value is violated and should be able to change it into right value. However, the current situation is the ESoT Model Browser give information in form of a complex programming equation that even hard for a programmer to understand, also it is not possible to change the value of any parameter.

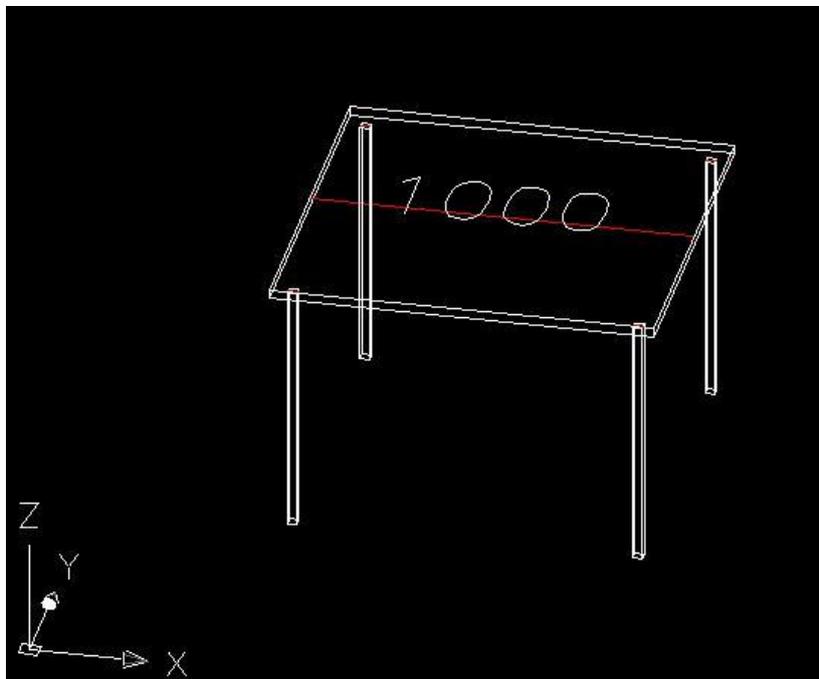


Figure 3.3: Picture of table in Graphical Editor.

Why the company needs improvement in ESoT Model Browser? In Figure 3.4, you may notice the window title is Constraint validation error, this was its name before ESoT Model Browser, the company supervisor suggest this name. ESoT Model Browser shows name, equation, properties and quantities of violated constraints. All of these are hard to understand for normal user. In the figure 3.4, the name is "Constraint Drive\_GTH[0].Drive\_bed", what is it mean? The equation text is "Rnd150\_std.Width = IF(Rnd150\_std,Quantity = 1 ; Width ; 1200)" this equation is base on Microsoft Excel equation syntax. Perhaps, for expert user it is understandable but most cases the equations are more than one line and it will take time to know what is wrong, beside that what is "Rnd150\_std"? In properties and quantities side only show current value and it is not possible to change it, this also apply to the is-it-locked checkboxes. Currently, this window only shows the information that hard to understand and user cannot do anything about it.

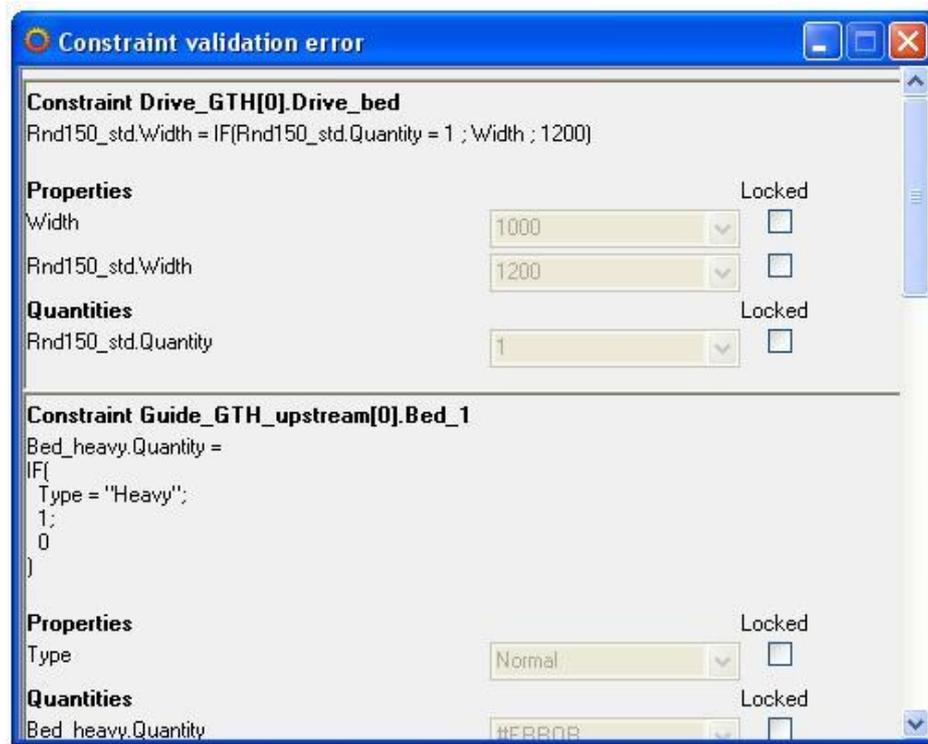


Figure 3.4: example of current version of ESoT Model Browser

### 3.3 PROJECT DESCRIPTION

The Technical Information Systems as developer of ESoT wants the ESoT Model Browser can be more reliable than what it is now. The main goal is to build the new ESoT Model Browser that displays the understandable message for expert user additionally for normal user, also allow user to make changes and submit them. The meetings had been held to discuss the detail of requirements of the new ESoT Model Browser, from these meetings the functional and non-functional requirements are acquired. The intern concluded all the requirements and generated the MoSCoW table (see table 3.1)

The company divides the project into two parts, the first part which also main part of this project is to build new ESoT Model Browser. This should contain all requirements in must have – **M** and should have – **S** (see table 3.1). This part is done by several iterations where the first iteration has the requirements in must have – **M**, and next iterations have the requirements in should have – **S**. The second part is to generate the equation text into human readable expression, this means the second part also contain all the requirements in could have – **C** (in this project, only requirement number 4).



No.	Functional requirements	M	S	C	W
1	Show the violated constraints.	V			
2	Allow user to change value of parameter.	V			
3	Allow user to change lock value of parameter.	V			
4	Generate a clarification message of a constraint.			V	
5	Has a function to display a specific parameter in Graphical Editor.		V		
6	Be able to submit or cancel the parameter's values that have been changed to Data Editor.	V			
7	Navigation feature: tracks the activities of user for example changing a value and selecting other parameter.		V		
8	Automatically update the interface when user change a value.		V		
9	Possibility to change more than one value without updating the interface.		V		
10	System capable to find the correct value.		V		
11	Show the non-violated constraints.		V		
12	Highlight the selected parameter in equation text.		V		
No.	Non-functional requirements	M	S	C	W
1	Has a better user interface that user is easy to notice the error.	V			
2	Every button has an icon.				V

Table 3.1: The MoSCoW table.

**Reference:**

**M** – Must have this function

In the *Must* area, this function will be implemented in the application.

**S** – Should have this function if at all possible

In the *Should* area, this function will also be implemented in the application, but are not necessary for delivery in the current delivery time plan.

**C** – Could have this function if it does not affect anything else

In the *Could* area, these functions are less critical and often seen as nice to have.

**W** – Will not have this function this time but would like in the future

In the *Will not* area, in the meantime these functions are not implemented in the application but maybe in the future.



### 3.4 METHODOLOGY

This project is using the agile methodology where the development takes several cycles. Every cycle include initiating, defining, designing, realization, testing and transferring. The methodology to develop the codes as well as the testing is described in the figure below:

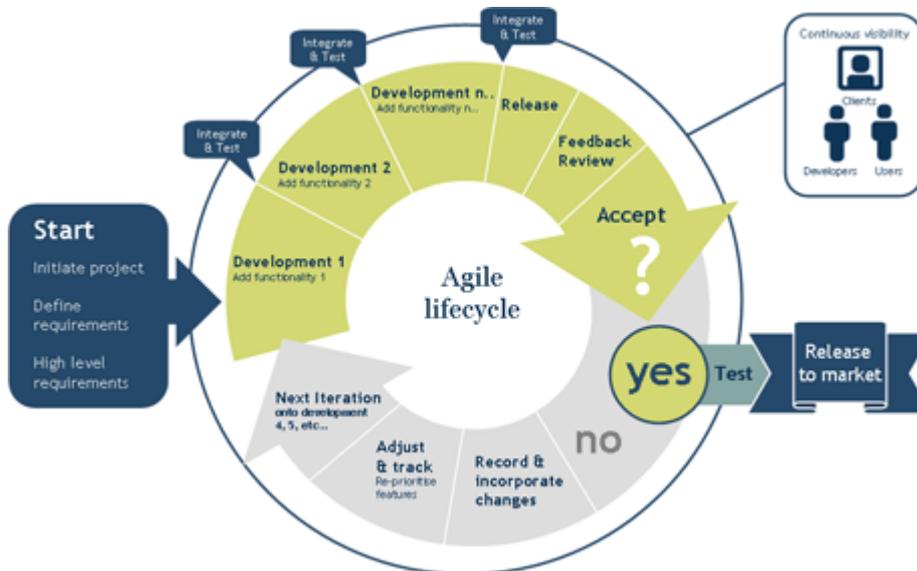


Figure 3.5: Agile Development Model taken from <http://www.talkwiseconsulting.com/agile-methodology.html>

In this project, the first cycle is the initial version of the first part with deliverable of the Must have area from MoSCoW table (see table 3.1). There will be several cycle to complete all the requirements in MoSCoW table. However, due to the lack of time this project only covers the first cycle. In the future, the company will decide whether to do the next cycle or not.

### 3.5 PROJECT TOOLS AND ENVIRONMENT

This sub-chapter describes and list the tools and environment that were used in this project.

List of tools:

- **Microsoft Visual Studio 2010** – the main tool to develop the ESoT Model Browser, and also for built the design of ESoT Model Browser.
- **Microsoft Office Visio 2007** – to built the flow diagram and first scratch of ESoT Model Browser.
- **VI XML Editor** – is a software that developed by Vanderlande industries. It is used to display detail of XML that is generated by the engine.
- **Notepad++** – to help editing the XML file to get the desire result.
- **DataeditorController** – a small application inside ESoT to test the XML file.
- **AutoCAD** – the main application to run the ESoT.

List of environments:

- **C++**
- **XML**
- **Excel equation formula** – this is used in developing the equation constraint inside XML library.



## 4 PROJECT PHASING

This chapter describes the idea taken to solve the problem and manage to finish the project as the company wanted. It divides into 4 phases. Each phase acted as a milestone and at the end of last phase a meeting with was held with the company and the school supervisor.

### 4.1 PHASE 1: INITIATIVE PHASE

The initiative phase is where the intern had to learn about ESoT and old ESoT model browser (the structure, how do they work, what are the inputs and the outputs, creates independent library component, etc.). Few times meeting are held to discuss more detail of this project with the company supervisor and sometimes also with company manager. Project plan was made base on this information. The project plan includes the detail of the project, and time planning (see appendix I). The project plan sent to the school and company supervisor to have it reviewed. Few updates and changes were made based on the school and company supervisor's feedbacks. Finally, the approved version was used as a guide to finish this project.

Result: Approved project plan and time management.

### 4.2 PHASE 2: DEFINITION AND DESIGN PHASE

The definition and design phase is where meetings were held to gather information about the functional and non-functional requirements with company manager and supervisor. The intern formulates all the requirements and sends it to the company supervisor. Lastly, some requirements were finalized and some new requirements were added by the company supervisor. The list of requirements document was made base on this information. A few changes and updates were made in the list of requirements document based on company manager's and supervisor's feedbacks, afterward they approved it.

Result: Approved list of requirement.

The design had been through three versions from the beginning until the final version. The sub-chapter below will explain more detail about the versions.

#### 4.2.1 The First version

This first version of the design prioritizes simplicity due to the complexity and abstruse of the current version. In the first version the GUI is divided into 3 parts: main part where the list of parameter and list of constraint are, toolbar where navigation buttons and footer OK-Cancel buttons. The idea how it is divided is adopted from most of web browser applications which have navigation button at the top of the window followed by the main content. Lastly the intern adds the footer which contains the ok, cancel, show/hide and zoomTo buttons. The main content is divided into two parts the list of parameter and the list of constraint by reason of user will be easier to understand if the parameters and constraints are split.

Inside the toolbar, there is a navigation button which have undo and redo button function. Therefore user could go back to the previous state where the user has not change the value.

The list of parameter:

- Icons: indicates the parameter. X icon for invalid parameter, V icon for valid parameter.
- Name: the name of parameter.
- Value: list of possible value for the parameter.
- Locked: checked means user and engine cannot change parameter value, unchecked means user and engine can change parameter value.

The list of constraint:

- Icons: indicates the constraint. X icon for invalid constraint, V icon for valid constraint.



- Name: the name of constraint.
- Equation: the formula of the constraint.
- Clarification of constraint: sentences that are generated from formula to readable human expression.

Footer contains zoomTo, show/hide, ok and cancel buttons also notification of the hidden constraints.

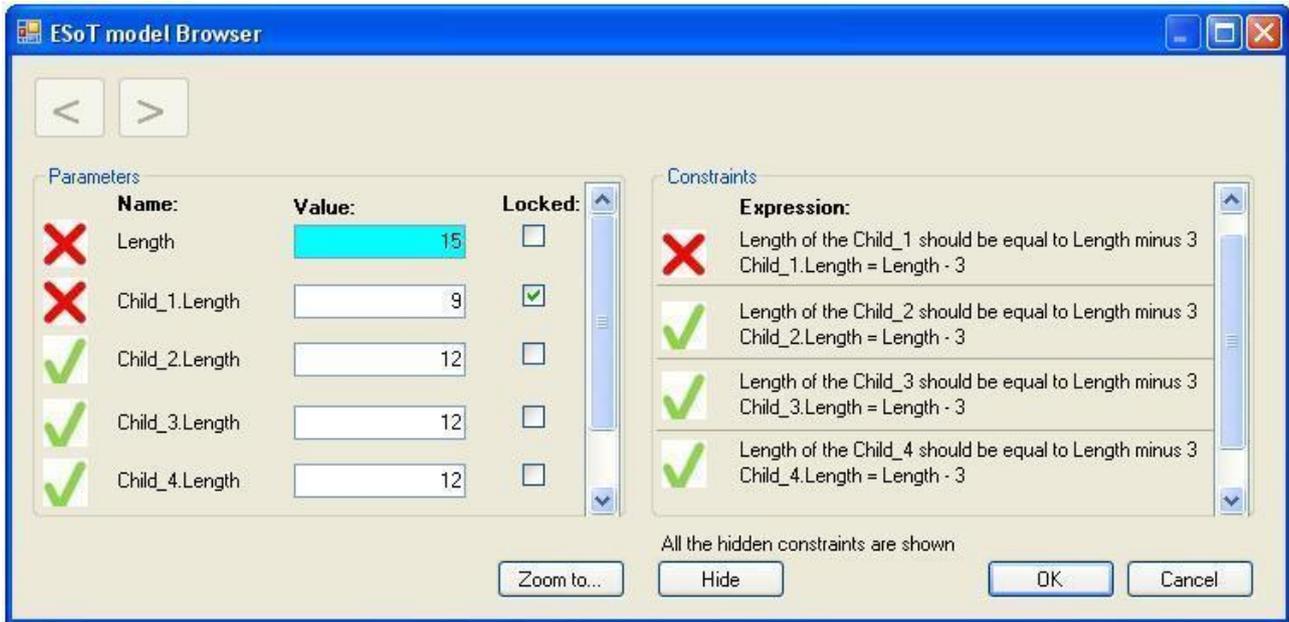


Figure 4.1: The first design of ESoT Model Browser

The advantages of the first version:

- User could change value from parameter.
- Could recognize which one is invalid with an icon.
- Could understand a formula better with the Clarification of constraint.
- User can submit a value that already changed.

The disadvantage of the first version:

- If there are a lot of violation constraints, user will get a difficulty to find the constraint which the user searches in the list of constraint violation.
- In this version, not the entire requirement covered such as “Has a function to display a specific parameter in Graphical Editor”. As a result, this version needs to be improved.

#### 4.2.2 The Second version

After knowing the weakness of the previous version, there are changes and additional functionalities in this version. The list of constraints has been change from using regular text box to tree view, this decision is made because in tree view the contents can be well organized, easily expanded and collapsed. Thus, functionality of show/hide button is changed from showing/hiding the constraint to expanding/collapsing all the constraint in the tree view. The footer now is exclusive only for ok and cancel buttons since the zoomTo and show/hide buttons are moved to the toolbar next to the navigation buttons also new functionalities such as freeze and configure buttons are added into toolbar. User will easily find the functionalities in the toolbar than in footer, also it can be confusing if some functions are in toolbar and others are in footer.

The new functionalities are added in this version which are configure and freeze button. The configure button will try to find valid values. The freeze button is divided ESoT Model Browser into two states: frozen and



unfrozen state. Frozen state is where user can change more than one parameter without invoking the validator also the background of EsoT Model Browser will change color to notify the user.

Even though the improvement is already implemented, there are still some flaw such as show and hide button give unsatisfactory result. User still will encounter the difficulty when amount if constraint are big (more than 20 constraints) although the user already helped by the show and hide button. To solve this problem, the idea needs to be simpler.

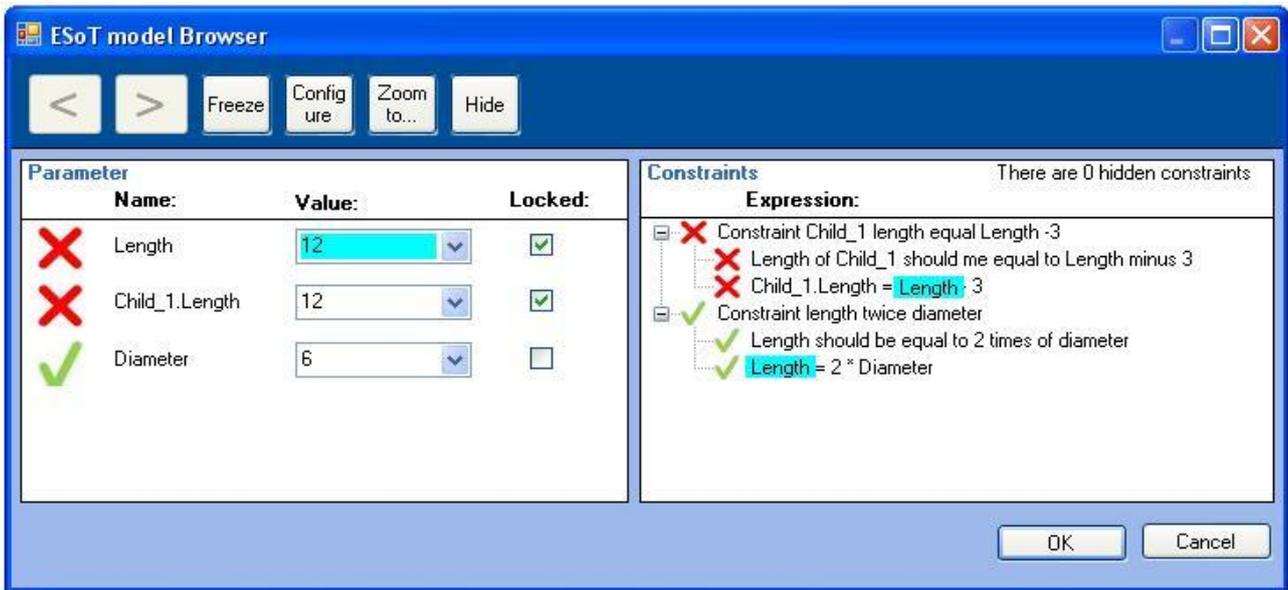


Figure 4.2: The second design of ESoT Model Browser

### 4.2.3 The Final and Initial version

A lot of changes have been made in this version and one of them is the main part's GUI is divided into 3 parts instead of 2 parts:

- Parameter: a list of parameters.
- Constraint tree view: a tree view that show all the violated constraint, non-violated constraint containing parameter, and non-violated constraint not containing parameter.
- Constraint detail: a tree view that show all the violated constraint, non-violated constraint containing parameter, and non-violated constraint not containing parameter.

Toolbar and footer have the same functionalities as in the previous version. With changing list of violation constraint to constraint tree view and constraint detail, it is easier for the user to know the error on the constraint. This concept is inspired from window explorer on Microsoft OS where the left side of the window is the tree view of directory and on the right side there is a content of the directory. When the user selects one of the constraints on the tree view, the detail of the constraint will show up on the constraint detail.

After built this version, the company supervisor suggested to use this version as the final version and continue with implementation. However, before the implementation this final version was reduced in order to build the initial version which will be implemented first. The initial version only has the must have requirements.

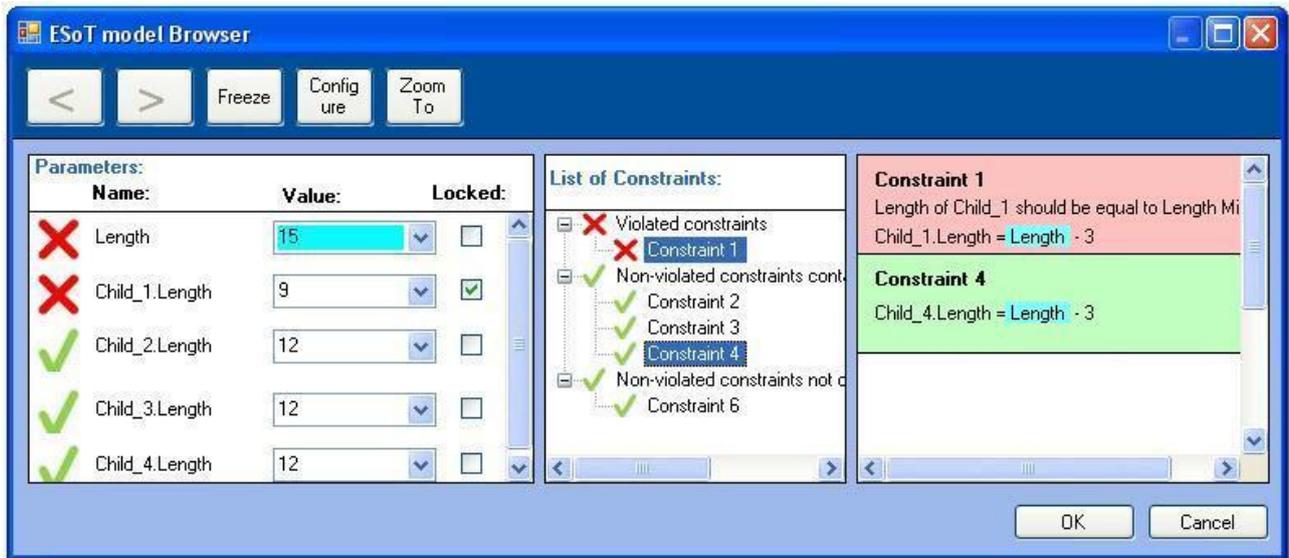


Figure 4.3: The final design of ESoT Model Browser

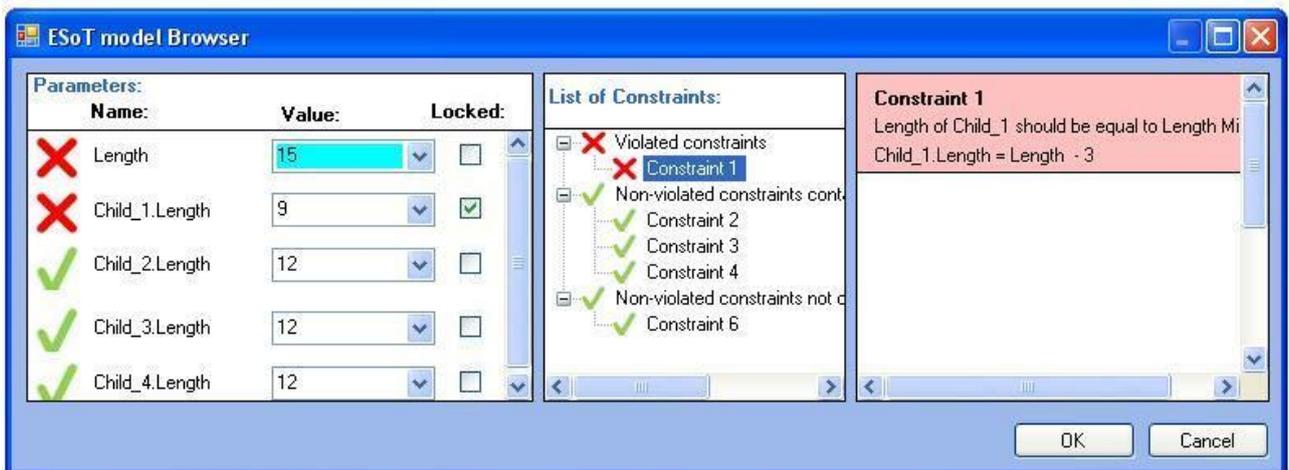


Figure 4.4: The initial design of ESoT Model Browser

### 4.3 PHASE 3: IMPLEMENTATION PHASE

The new application began to develop in this phase. In order to keep in track of the progress, weekly meetings with the company supervisor were held. At the end of this phase, the application should finish with all the functionality in the list of requirement document. The realization is not as planned, the delay occurs due to the intern had difficulty in understanding the source code of ESoT, it required new planning. A meeting with company manager and supervisor were held to discuss the new planning, the result is the initial version as the final deliverable. The result is base on some considerations such as: the final version is not possible to be implemented before end of this project; the initial version still a lot better than the current version and the company will use it; the company can use the design of final version if the company continues developing the ESoT Model Browser.

During the implementation phase, several technical problems were occurred. To handle these problems, discussions were been held between the intern and the company supervisor. With the help of company supervisor, the intern was success to solved them, more detail regarding the problem will be describe in chapter 5.4.



The GUI in the implementation cannot be the same as the GUI in design, because of the incompatible application. It is possible to make it compatible but it cost a lot of time in implementation and it is not on the requirement. The GUI was built based on the existing functionalities which also covered the same functions that are needed but with different style and appearance (see figure 4.5).

Result: The initial version of ESoT Model Browser (see figure 4.5).

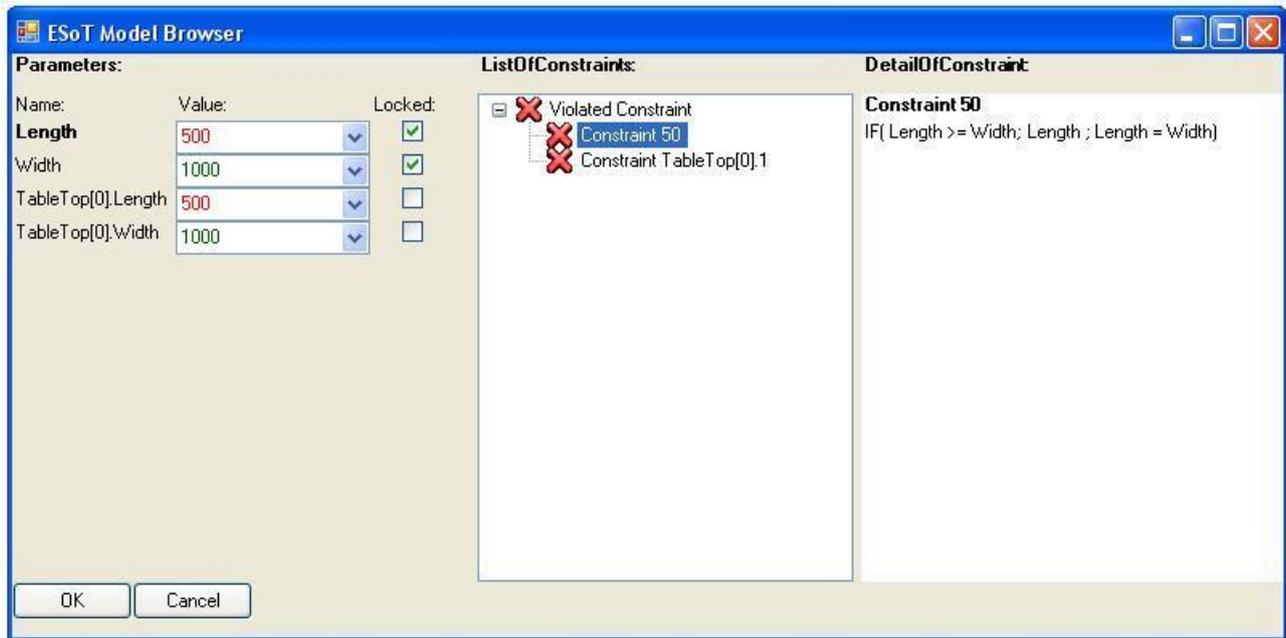


Figure 4.5: The initial design with implementation GUI of ESoT Model Browser

#### 4.4 PHASE 4: DELIVERY PHASE

The delivery phase is the last phase of the process. The application and final version of documentations were delivered to the company. This phase comes after final discussions about things that need to be changed from implement phase.

In this phase, the final report and presentation were prepared for final presentation and defense. Company signed the final report and some documentation based on school requirements. All these documents were collected by the intern and handed over to the school.

Result:

- Delivered the initial version of ESoT Model Browser.
- Delivered the final report.
- Delivered the documents to the school based of school requirements.



## 5 THE NEW COMPONENT

This chapter describes more details about ESoT Model Browser and how it works.

### 5.1 THE NEW ESOT MODEL BROWSER

The ESoT Model Browser is a component inside ESoT application that handles all violated constraints. Configurator and validator function are already in the ESoT. Configurator functions automatically change values which are not lock based on constraints. On the other hand, validator functions to check whether the value is valid or not. When a value is invalid then XML form of ESoT model browser will be generated. XML form of ESoT Model Browser will read by ESoT.

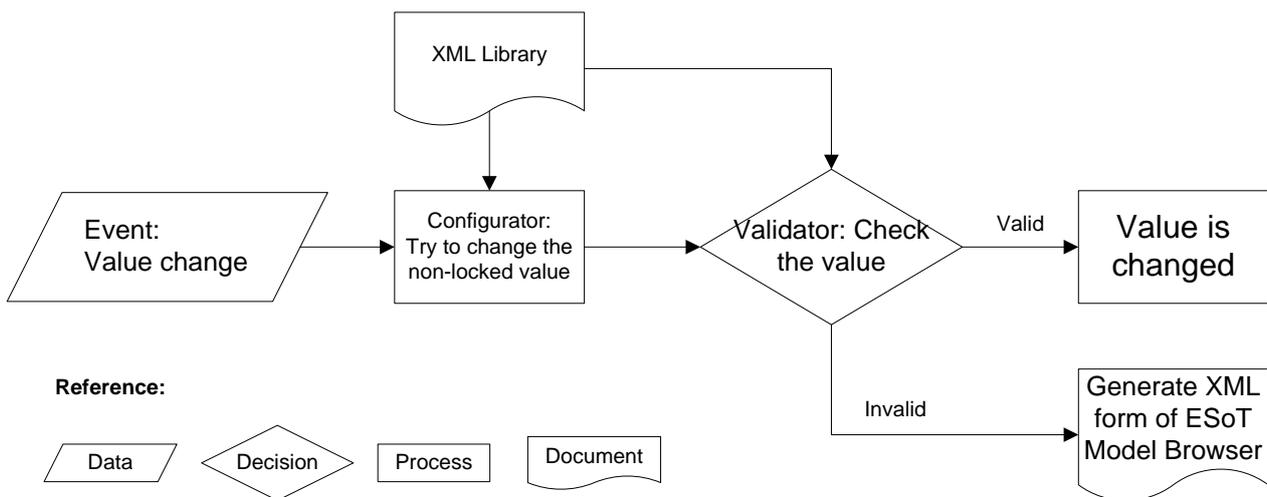


Figure 5.1: The process flow of configurator and validator

ESoT Model Browser displays violated constraints and parameters that consist in violated constraints, also valid constraints and parameters that are related to the violated constraints. ESoT Model Browser window is divided into three main parts:

- List of parameters

The list of parameters displays name, value and is-locked of parameters. The combo box is used for listing all the possible values. The check box is used for indicating is-locked, if the check box is checked then the parameter is locked and vice versa. This part is necessary in order user to make changes.

- Tree view of constraints

The tree view of constraints is divided into three groups; first is to list all the violated constraints, second is to list all the non-violated constraints and consist of parameter that is violated, third is to list all the non-violated constraint and not consist of parameter that is violated. For all groups, only the name of constraints is displayed. Tree view is used instead of a list because the concept is adopted from windows explorer and most people familiar with it.

- Detail of a constraint

The detail of a constraint displays name and equation formula of a parameter. Furthermore the clarification of the constraint will be added in this part. The idea of this part come along with the idea of tree view where this part is to display the detail of the constraint that is chosen from tree view.



## 5.2 EVENT MESSAGE

When generated XML form for ESoT Model Browser, some components such as combo box, tree view and button have event message to communicate with ESoT engine.

### 5.2.1 Tree view selected message

One of the components which have event message is tree view. The task of the event is showing the name and equation text in detail of constraint part. This process will proceed when user click one of the constraint in tree view and it will trigger *tree view selected message* which send a message to the engine that one of the constraint have been clicked. Engine will search equation text from the constraint in XML library and generate new XML form to update XML form ESoT Model Browser.

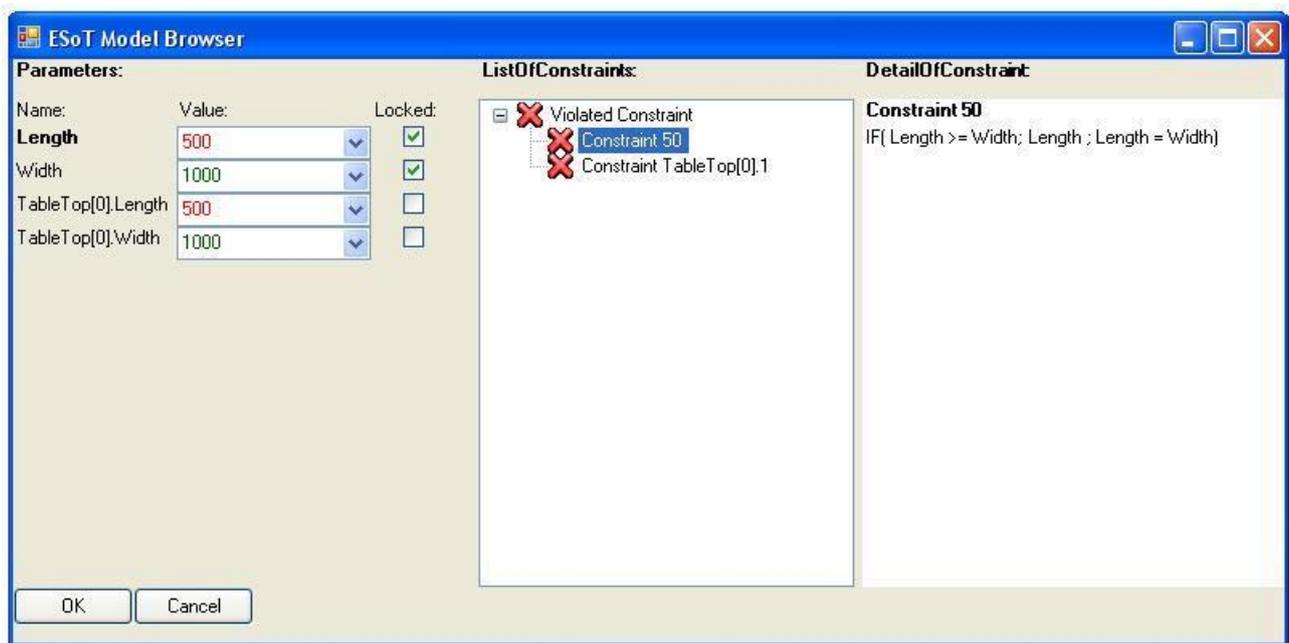


Figure 5.2: Example of tree view message selected (constraint 50 is selected).

### 5.2.2 Combo box selected message

When XML form of ESoT is made, there is just one value in list of combo box which is the current value. The task of *combo box selected message* is to update list of values of the combo box with all possible value from parameter of the combo box. There are three possible colors in combo box: red indicates invalid value, green indicated valid value and black indicates unidentified valid or invalid value.

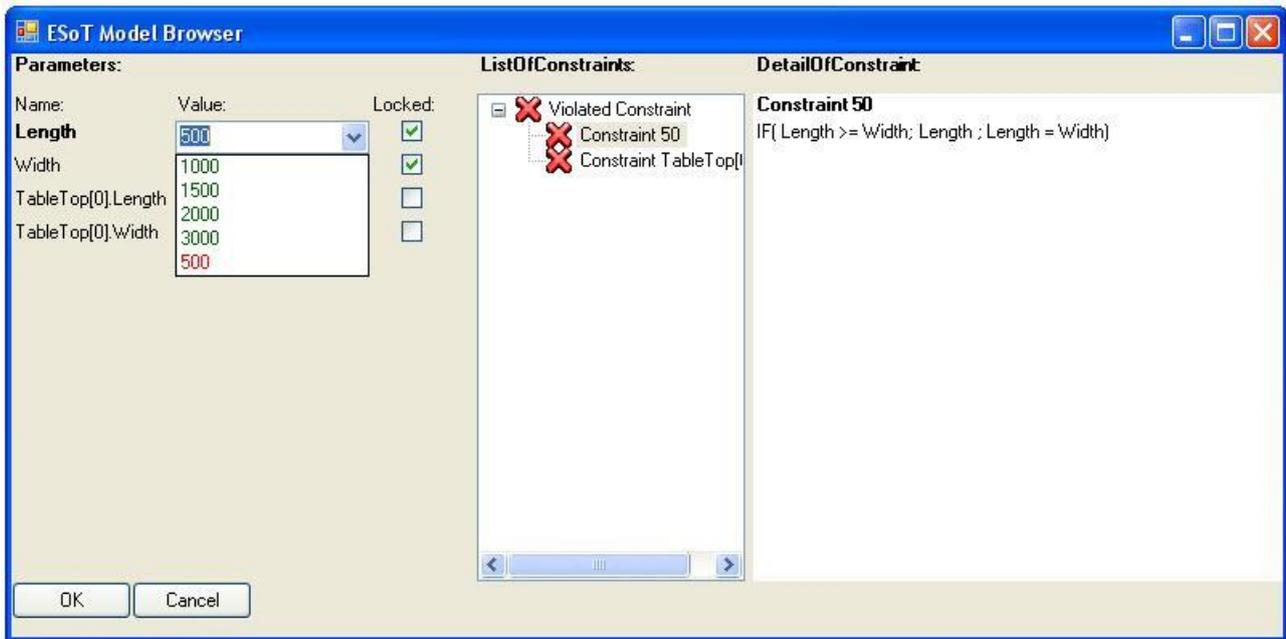


Figure 5.3: Example of combo box selected message (length is selected).

### 5.2.3 Combo box value changed message

While a user change value of the combo box, a message is sent to engine. After the engine receive the message, the engine will check whether the value which was sent is equal with the current value, if not equal then the engine will generate XML form to update combo box with a new value. While the value is being changed, the color becomes black because the validator functions not triggered.

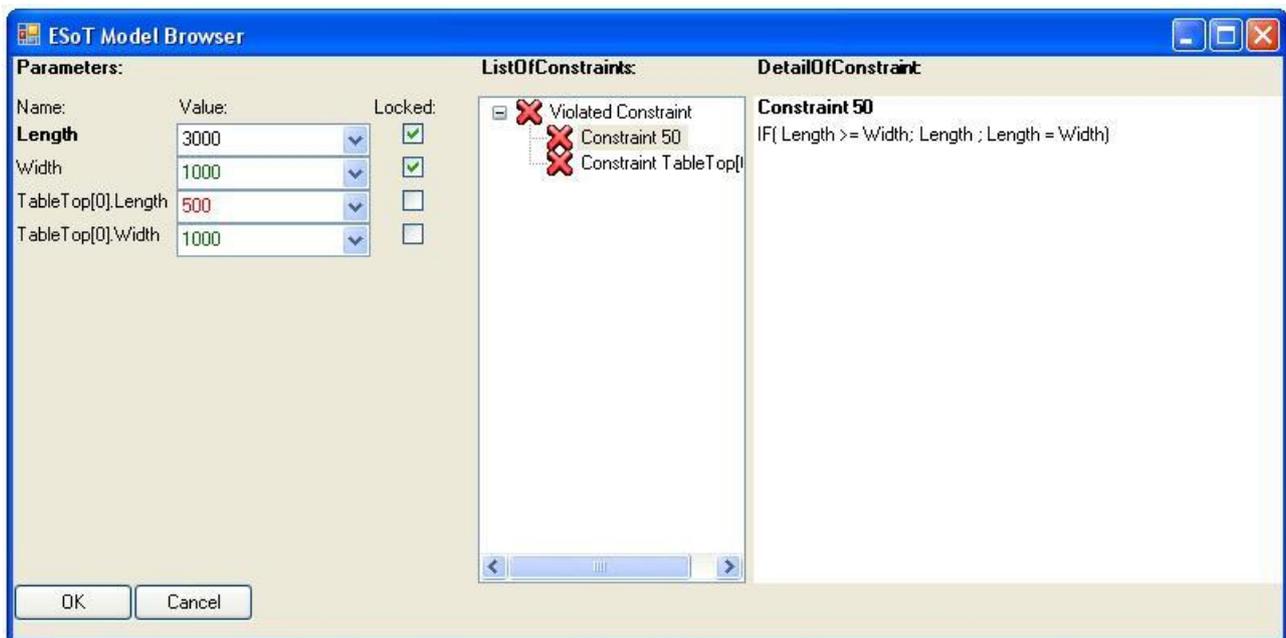


Figure 5.4: Example of value change message (length is changed from 500 to 3000).



### 5.3 SUBMIT THE VALUES

This sub chapter describes the process of submitting the value to ESoT and how it is created.

Several ideas come up in order to find the way to implement the submit functionality, and after several discussions the decision is made. The communication between the engine and ESoT Model Browser is a one way communication where the engine gives all information to ESoT Model Browser to be shown. This makes impossible to submit the value into engine with the current situation, to solved this problem new communication is built. This new communication sends the lists of parameters that were changed in ESoT Model Browser to the engine. However, two objects are created in ESoT Model Browser to store the previous valid values and the changed values, because these two object are needed to be sent to the engine.

When ESoT Model Browser is generated, the previous valid values are saved in an object called *original values*. Each time user change a value, the changes will be recorded in object called *list of changed values*. Both objects will be sent to change value in ESoT engine.

Cancel button send *the original values* to ESoT engine therefore the value in data editor are values when ESoT Model Browser not yet generated. OK button sends *the list of changed values* to ESoT engine therefore the values in data editor will be the same as values which are changed by the user in ESoT Model Browser. After *the list of changed values* accepted, engine will check whether the value is valid or not, if the value is valid then *the original value* will be deleted. If the value is invalid then engine will generate new XML form of ESoT Model Browser without changing the previous *original values*.

Example Case:

A table has length 1500 and width 1000. Length cannot be smaller than width. When the user change the length into 500 then ESoT Model Browser will appear. *The original value* will have value length 1500 and width 1000. While user changes width to 1500 and length 1000, the changes will be in *the list of changed values*. User click OK button, ESoT Model Browser will send values width 1500 and length 1000 which are collected from *the list of changed values*, and after checking the values and apparently the values are still incorrect because the length is smaller than width. Therefore new ESoT Model Browser window will appear and user change value into width 500. If the user clicks OK button then the value which will be sent are length 1000 and width 500, whereas the user click cancel button then the value which will be sent is length 1500 and width 1000 which came from *the original values*.

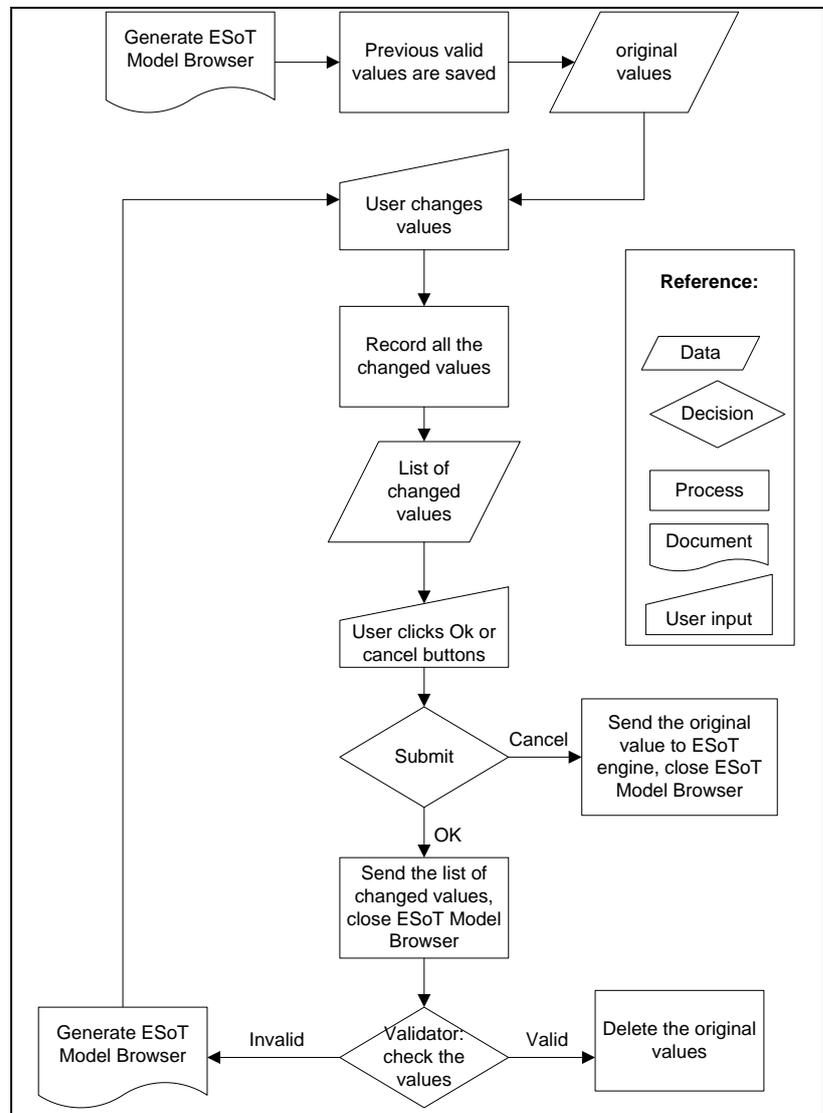


Figure 5.5: Process flow of ESoT Model Browser



## 5.4 PROBLEMS AND SOLUTIONS

This sub-chapter describes the problems and solutions that the intern encountered during this project.

The first problem is the lack of time in this project. The intern had miscalculated the estimated time of the project due to the limited knowledge of the intern in implementing the functionalities, therefore several meetings were held to find the solution. The result is to reduce the deliverable from the final version (all requirements in MoSCoW table. See table 3.1) to the initial version (the must have requirement in MoSCoW table). This is approved by the company with consideration of the initial version still being better than the current version, the intern that has limited knowledge still in a learning situation. The impacts are the changes of several functionalities (table 5.1) that are adapted for the initial version.

Initial function	Solution
The valid indication in the list of parameters should be based on <i>validator</i> where it is checked.	All the valid indications become <i>maybe</i> indication without calling the <i>validator</i> . The <i>validator</i> will be called when all the changed values are sent to the engine.
Updates the ESoT Model Browser when one of the values is selected or changed.	The ESoT Model Browser will not be updated. It will record all the changes that the user made, and when the Ok button is clicked, the record will be sent to the engine and call the <i>validator</i> . If the values are still not valid then a new ESoT Model Browser window will appear.

Table 5.1: The functionalities that are adapted to the initial version

The second problems are the possibility to have duplication of a parameter and two components may have the same name of a parameter. The engine generates a list of parameters based on every constraint and when a parameter is used in two different constraints the parameter will be shown twice. The intern suggested to add the name of constraints in the list of parameters to indicate a parameter belongs to which constraint, but after having a discussion with the company manager this idea is not accepted, because the possibility to have duplication is still not solved. With some suggestion from the company manager about the mechanism, the intern came up with new ideas which are to put the component's name in front of the parameter's name to solve the similar parameter name from different components, and filter the same parameter.

Many of the problems that the intern deals with are technical skills where the syntax error, creating or finding the right object to be used. Also, the intern had difficulty in analyzing the ESoT application structure. This is where the company supervisors gave the intern assistance. When the problem occurs regarding the basic C++ language, the intern used the internet. However, the intern consults what kind of object is used to the company supervisors when implementing a functionality.



## 6 RESULT

The new ESoT Model Browser was built with all functionalities in MoSCoW table (Must have areas, see table 3.1). It definitely improves from the old ESoT Model Browser, even though not all requirements are delivered. The company understand the situation that the project being delayed due to limited knowledge of the intern and the lack of the time, and change the deliverable from the functionalities in Must have and Should have areas to only the Must have area in MoSCoW table; furthermore, the company will approve and use the new ESoT Model Browser, after several tests that will be done by Technical Information Systems department.

The new ESoT Model Browser is still under development when this document is made. It is expected to finish before 22 June 2012. The intern also deliver the list of bugs and errors that the intern found while developed the new ESoT Model Browser, and tries to fix them with remaining time. At the end of the internship, the new ESoT Model Browser has been through some changes and updates to make it as perfect as it can.

The new ESoT Model Browser is the main component in this project. It handles the violated constraints and displays them in user-friendly user interface, and also user allow to make changes of the value and is-locked. The component is coded in C++ language using visual studio 2010, and displayed in AutoCAD. Using XML is the way of Technical Information Systems department to connect ESoT and AutoCAD, therefore the ESoT Model Browser inherits this method. The event messages are also the important aspect in this component. The message generate new XML file that is used to update the existing XML form.

The functionalities in Should have, Could have and Will not areas in MoSCoW table are not implemented due to the lack of time and priority. The company will consider continuing implement these functionalities in the future.

At the end of the internship, the documents also were handed over to the company. The requirement document is the most important document for the company. The company will used these document as a guide to develop the functionalities in Should have, Could have and Will not areas from MoSCoW table.

During the internship, Project Plan Document was made. These documents had undergone several changes, in order to enable a more efficient solution to the problems. The updated documents handed over to the company at the end of the internship.



## CONCLUSION AND RECOMMENDATION

ESoT Model Browser is a component in ESoT application that handles the violated constraint. It displays the detail of constraints. The new ESoT Model Browser enables the user of ESoT analyzes the constraints and changes the parameters.

The new ESoT Model Browser have more functionality and user-friendly than the old ESoT Model Browser. New features have been implemented in the new ESoT Model Browser such as the new GUI, possibility to make changes and submitted the changes to the engine. These features are to help user to have better understandability and be able to configure the value of parameters in order to get the valid value. Several features were not implemented in this project due to the lack of time, which the features are to cover all the requirement except the must have area (see table 3.1)

To conclude, the goals and objectives initially set for the project were not fully achieved due to the intern had difficulty in understanding the source code of ESoT and the limited amount of time. However the goals and objectives were decreased and adjusted. The new goals and objectives were achieved; the required deliverables were produced, revised and submitted to the company. The ESoT Model Browser has been through some changes and has been updates to make as perfect as possible.

The experience and the knowledge obtained during the internship are priceless. Not only related to the IT knowledge, but also project management and the Netherlands culture. It gives the opportunity to meet various personalities and to make acquaintances.

As for Vanderlande Industries (The company), personally to the Technical Information Systems department, I do not have any recommendation and suggestions, because it is well structured and well supervised. However, I have suggestions to the hiring department in Vanderlande Industries to be more organized in processing the documentation of an employee.



## EVALUATION

This graduation project, working in a company, was a great opportunity for me to gain experience and felt the real working world. I can apply the knowledge that I have learnt so far in the real environment. I learned a lot about project management, and of course the IT knowledge itself. I gained experience in using C++ in visual studio, AutoCAD software, XML file and how to make an efficient program. The one that also interesting and challenging is that I learnt how to develops a functionality in a well programmed software, to do that understanding the mechanism of a software is necessary. Furthermore, implementation of the code will be much easier. I gained the experienced how to analyze a software with helps of one of its developers.

I expanded my knowledge about XML that is a connection between ESoT and AutoCAD. I had limited knowledge about this, however after I had experience and work with XML, my knowledge about this is increased.

The two weekly meeting with the company manager and weekly meeting with company supervisor are a good method to keep on track my progress. I would suggest this method to use while the intern doing their internship or graduation project. With this method the company manager and supervisor can know the progress and is it according to the time plan.

At the end of my internship, I am a bit disappointed due to the component I built that not finished according to the plan. It is not an easy task to plan the estimated time, I found this aspect is something that only can be learned through experience. My mistakes are I had limited technical knowledge and I was not aware about unexpected errors such as finding the right object and syntax error.

Another aspect, which makes this project an unforgettable experienced for me, is the opportunity to work in one of the biggest company in the Netherlands. The Technical Information Systems department where I am doing my internship is a well-structured and well organized department.

I found the difficulty in social life at the office. Vanderlande Industries is a Dutch company even though most of them are speaking English, but they use Dutch as daily language. This makes me has difficulty to get along with the people in Vanderlande Industries. Also I have an introvert personality that makes it more difficult. However I learn from this experience, I will more active in social life and to learn Dutch language better.

Overall, I was happy to have opportunity working in here. It shows me that the skill and knowledge that I learned so far can be used for working. I definitely will look for opportunity to working in this company again!



## REFERANCE

[1] AutoCAD, <http://usa.autodesk.com/autocad/>

Vanderlande, <http://vanderlande.com/>



APPENDIX I: PROJECT PLAN



# Project Plan

Improving ESoT model browser

Nikolas Dandy

23 March 2012



## Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>29</b>
<b>2</b>	<b>Project statement .....</b>	<b>30</b>
<b>2.1</b>	<b>Formal Client.....</b>	<b>30</b>
<b>2.2</b>	<b>Project Leader.....</b>	<b>30</b>
<b>2.3</b>	<b>Current situation.....</b>	<b>30</b>
<b>2.4</b>	<b>Project justification .....</b>	<b>31</b>
<b>2.5</b>	<b>Project product .....</b>	<b>31</b>
<b>2.6</b>	<b>Project deliverable and non-deliverable .....</b>	<b>31</b>
<b>2.7</b>	<b>Project constraints .....</b>	<b>31</b>
<b>2.8</b>	<b>Project risks .....</b>	<b>32</b>
<b>3</b>	<b>Project Phasing .....</b>	<b>33</b>
<b>3.1</b>	<b>Introduction phase .....</b>	<b>33</b>
<b>3.2</b>	<b>Planning and design phase.....</b>	<b>33</b>
<b>3.3</b>	<b>Implementation phase .....</b>	<b>33</b>
<b>3.4</b>	<b>Testing phase .....</b>	<b>33</b>
<b>4</b>	<b>Project Management plan.....</b>	<b>34</b>
<b>4.1</b>	<b>Money .....</b>	<b>34</b>
<b>4.2</b>	<b>Skills .....</b>	<b>34</b>
<b>4.3</b>	<b>Quality .....</b>	<b>34</b>
<b>4.4</b>	<b>Information.....</b>	<b>34</b>
<b>4.5</b>	<b>Time .....</b>	<b>35</b>
<b>4.6</b>	<b>Organization.....</b>	<b>36</b>
	4.6.1 Jochen Cobussen (Formal Client) .....	36
	4.6.2 Peter Kamps (Architect Tutor).....	36
	4.6.3 Kevin Pluk (Technical Tutor) .....	36
<b>5</b>	<b>Communication plan .....</b>	<b>37</b>
<b>6</b>	<b>Reference .....</b>	<b>38</b>



## 1 INTRODUCTION

Vanderlande Industries(VI) is one of leading company in baggage handling, distribution and parcel & postal. VI uses high technology hardware and software to increase company performance, one of the software is Engineering Suite of Tools (ESoT). ESoT is used to design baggage systems and it was developed by Technical Information Systems (TIS) department.

ESoT was developed in C++ environment. It is a plug-in in AutoCAD [\[1\]](#). Its goal is to help users to build baggage systems that can be implemented in real situation. The main functions of ESoT are working, although there are some bugs to be fixed and new functions to be implemented.

This project plan contain of background, current situation and management plan of “ESoT model browser” which is one of the component of ESoT. ESoT model browser is a window that gives the user warning information about invalid parameter’s value and rules (constraints).



## 2 PROJECT STATEMENT

ESoT model browser is a window that triggers when a user put an invalid value in a certain parameter on ESoT. It shows all the violated parameters and constraints as a warning to user. In this window, user should be able to understand why the value is error and change it to right value, however these functionalities not implemented yet.

This project is divided in two parts. The first part also main part of this project is to re-design the current version of ESoT model browser window. The current version of ESoT model browser window is a mess, and it makes the user hard to understand the message given. Furthermore, it does not allow the user to change the value, lock, or unlock of parameters. TIS department as developer of ESoT demands a new design of ESoT model browser, that is more useful and not only give understandable message but also interactive, therefore the user can change it into a valid value.

The second part is optional for project, it will be executed when there is enough time after the first part is finished. The violated constraints' messages are given in formula as constraint in the library file. This makes the non-expert user hard to understand the messages, therefore the second part is to generate clarification of constraints. By doing this, the non-expert user will understand the messages and how to fix them.

### 2.1 FORMAL CLIENT

Jochen Cobussen.

From: Vanderlande Industries

Address: Vanderlandelaan 2 5466 RB, Veghel, Noord-Brabant, The Netherlands

Site: <http://www.vanderlande.nl/>

Email: [Jochen.Cobussen@vanderlande.com](mailto:Jochen.Cobussen@vanderlande.com)

### 2.2 PROJECT LEADER

Nikolas Dandy.

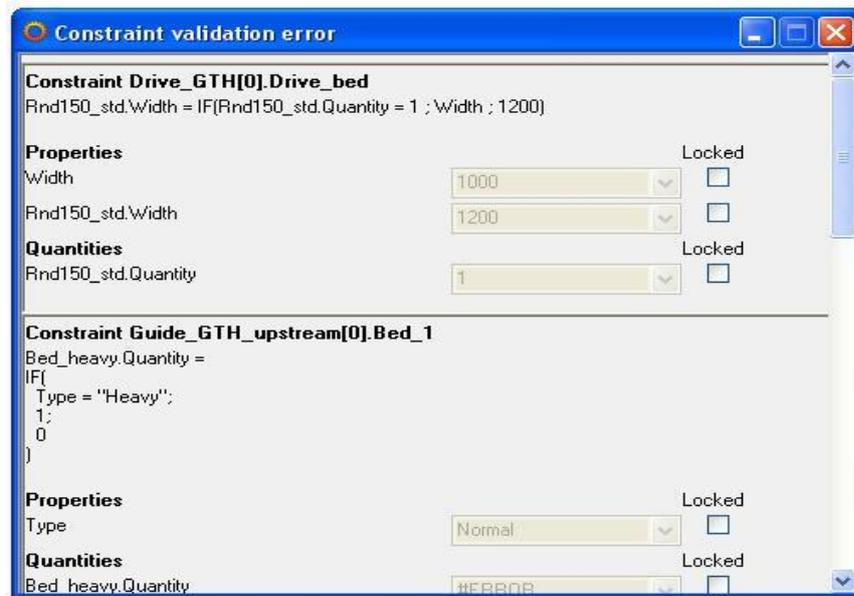
From: Fontys Hogescholen.

Email: [n.dandy@student.fontys.nl](mailto:n.dandy@student.fontys.nl), [Nikolas.Dandy@vanderlande.com](mailto:Nikolas.Dandy@vanderlande.com).

### 2.3 CURRENT SITUATION

ESoT is being developed by TIS department of Vanderlande Industries (VI). It was released, even though there are missing functionalities and having bugs. This is done because the main functionalities of ESoT already implemented and can be used by client. In the mean time, TIS improves the quality by adding new functionalities, re-designing functionalities to be more efficient and fixing bugs. One of the functionality needs to be improved is ESoT model browser.

ESoT model browser was named "Constraint Violation error". It is triggered when user insert an invalid value in a certain parameter. It shows all the violated constraints with its parameters but the information is in the formula which are hard to understand for non-programmer user and parameters are locked these make user cannot change anything (figure 1), therefore it is a bit useless and not user-friendly.



*Figure 1: Current ESoT model browser*

## 2.4 PROJECT JUSTIFICATION

VI's employees demands new ESoT model browser that have more functionality and better understandable message. The current version of ESoT model browser needs to be improved, in order to more useful and user-friendly.

## 2.5 PROJECT PRODUCT

- New design of ESoT model browser that have better usability and user-friendly.
- Use cases document
- Requirement document
- Test cases file
- Source code

## 2.6 PROJECT DELIVERABLE AND NON-DELIVERABLE

Deliverable

- Use cases document
- Requirement document
- Test cases file
- Source code

Non-deliverable

- Minute meeting
- Old version of use cases document, requirement document, test cases document and source code.

## 2.7 PROJECT CONSTRAINTS

Duration: 100 days

Development:

- C++
- Xml editor
- AutoCAD



## 2.8 PROJECT RISKS

The risks may occur are:

<b>Risk</b>	<b>Chance to happen</b>	<b>Impact</b>
Too much time is needed to implement a function, it cause the implementation cannot finish on time	Medium	Medium
Company system get malware, spyware, or virus that can cause loss all the data.	Low	Very high



### 3 PROJECT PHASING

As I explained in chapter 2, this project is divided by two parts. The first part is re-designing the current version of ESoT model browser, in this part there will be two or more iterations. The first is the initial design (prototype), followed by next iteration(this part may have more than one iteration) and the last is the final design. This is done because Jochen as client demands the working ESoT model browser as soon as possible. The second part is to generate clarification of the constraints.

The phases that I will explain later on, is base on agile methodology<sup>[2]</sup>. This means the phases will be repeated and the period for each phase is quite short.

This project will be divided to several phase which are:

- Introduction phase
- Planning and design phase
- Implementation phase
- Testing phase

#### 3.1 INTRODUCTION PHASE

- Activities:
  - In this phase I was trained by VI to the ESoT, what is ESoT and how it works.
  - Create new library to have a rough idea how the parameters, constraints and ESoT model browser are implemented.
- Sub-deliverables:
  - Show the result of new library to Peter Kamps (Architect tutor) as proof that I understand how ESoT work and to work with it.

#### 3.2 PLANNING AND DESIGN PHASE

- Activities:
  - Design new interface of ESoT model browser.
  - List all requirements for final design and initial design.
  - Make use cases for the requirements.
  - Make test cases file (automated test cases).
- Sub-deliverables:
  - Requirement document.
  - Use cases document.
  - Test cases file.

#### 3.3 IMPLEMENTATION PHASE

- Activities:
  - Exploring the code.
  - Modify the code.
  - Add functionality.
- Sub-deliverables:
  - New code.

#### 3.4 TESTING PHASE

- Activities:
  - Validate the test cases.
- Sub-deliverables:
  - Valid test cases document.



## 4 PROJECT MANAGEMENT PLAN

### 4.1 MONEY

Not relevant.

### 4.2 SKILLS

- C++
- XML
- AutoCAD

### 4.3 QUALITY

These are the to do list in order to improve the quality of the product.

- Review the documents.
- Testing the code.
- Validate the test cases.
- Fix bugs.

### 4.4 INFORMATION

	Project Plan	Use cases and requirement	Test cases	Project Report	Source code
Jochen Cobussen (Formal Client)	R,A,Di	R,A,Di	R	R,A,Di	R
Peter Kamps (Architect tutor)		Di	Di		A,Di
Peter Boots (Fontys Tutor)	R,A,Di	R	R	R,A,Di	
Nikolas Dandy (Project Leader)	Dr,S	Dr,S	Dr,S	Dr,S	Dr,S

**Legends:** Dr Draw up  
Di Discuss  
A Approve  
S Send  
R Receive



## 4.5 TIME

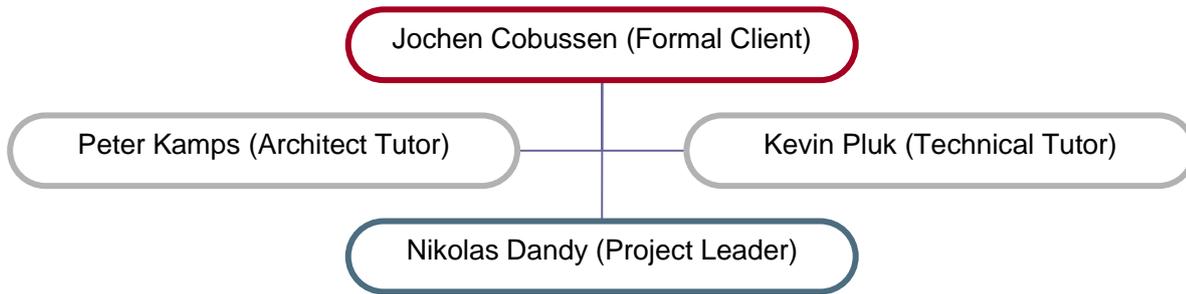
Duration of this project is 20 weeks.

Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Date (in 2012)	6/2-10/2	13/2-17/2	20/2-24/2	27/2-2/3	5/3-9/3	12/3-16/3	19/3-23/3	26/3-30/3	2/4-6/4	9/4-13/4	16/4-20/4	23/4-27/4	30/4-4/5	7/5-11/5	14/5-18/5	21/5-25/5	28/5-1/6	4/6-8/6	11/6-15/6	18/6-22/6
Introduction to ESoT	■																			
Planning & designing		■	■	■							■					■				
Exploring the code				■	■															
Making test cases					■	■	■				■	■				■	■			
Implementation							■	■	■	■		■	■	■	■		■	■	■	■
Testing first										■	■				■	■			■	■

- Legends:** ■ Progresses  
■ Deadlines



## 4.6 ORGANIZATION



### 4.6.1 Jochen Cobussen (Formal Client)

Jochen Cobussen is the Leader of TIS team. He is the one that I can ask regarding this project therefore he is my client.

### 4.6.2 Peter Kamps (Architect Tutor)

Peter Kamps is one of the architect in TIS team. He guides me in my research. I always discuss with him about “what should I do next?” and my ideas.

### 4.6.3 Kevin Pluk (Technical Tutor)

Kevin Pluk is one of the software programmer in TIS team. He knows the detail of the code. He guides me for all related to the code.



## 5 COMMUNICATION PLAN

This chapter will explain about how I as an intern communicate with Peter Boots as my Fontys mentor and Jochen Cobussen as my VI mentor. Peter Boots will visit VI twice, the first visit will be held between 5 March 2012 and 9 March 2012. It depends on availability of Peter Boots and Jochen Cobussen. The first visit will discuss about the project and its progress.

The second visit will be at the end of graduation period, it will be discussed. This visit will discuss about the result of my work in this project. At the end of every week, I will send an e-mail to Peter Boots about my progress in that week. I will have a meeting with Jochen Cobussen every two weeks to discuss my progress in this project.

In week 10(9 March 2010 -13 March 2012), I will send the first version of my final report to Peter Boots and Jochen Cobussen, afterward once every two weeks(week 12, 14,16 and 18) I will send new version of my final report.

### Contact info:

- Nikolas Dandy  
→ Email: [n.dandy@student.fontys.nl](mailto:n.dandy@student.fontys.nl), [Nikolas.Dandy@vanderlande.com](mailto:Nikolas.Dandy@vanderlande.com).
- Jochen Cobussen  
→ Email: [Jochen.Cobussen@vanderlande.com](mailto:Jochen.Cobussen@vanderlande.com)
- Peter Boots  
→ Email: [p.boots@fontys.nl](mailto:p.boots@fontys.nl)



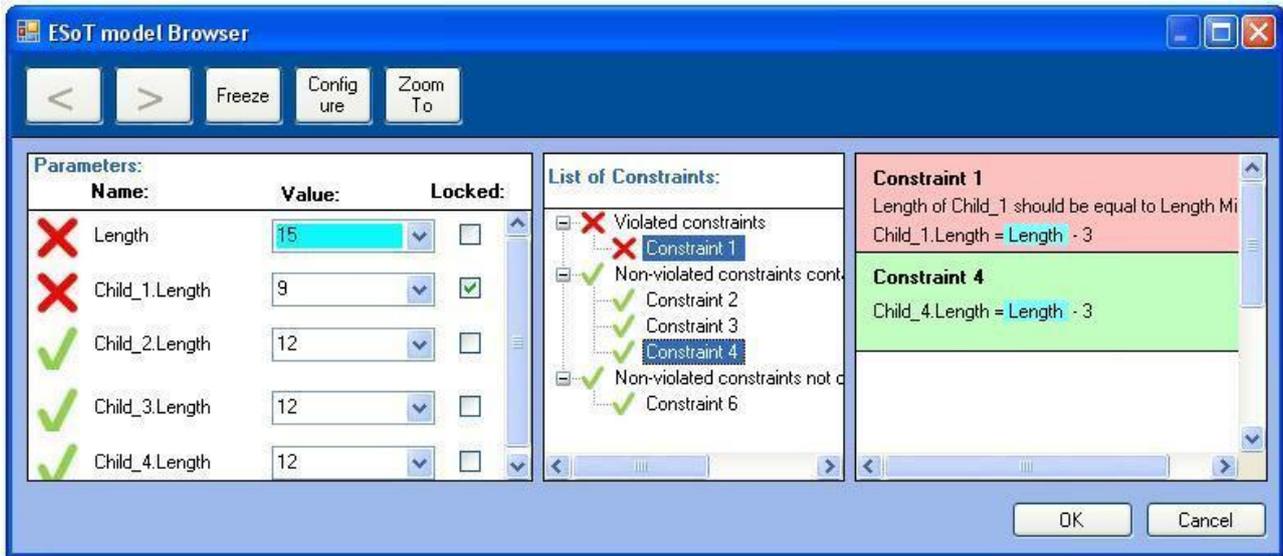
## 6 REFERENCE

[1] AutoCAD, <http://usa.autodesk.com/autocad/>

[2] Wikipedia, [http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development)



## APPENDIX II: FINAL VERSION REQUIREMENT DOCUMENT



### Requirements:

1. “ESoT model browser” maybe covers the function of the window better than “Constraint violation window”.

2. The window shows:

- The selected parameter (blue in the picture above),
- List of Constraints.
- Constraint’s detail<sup>[23]</sup>
- All parameters used in these constraint.
- All other parameters are violated.
- Red crosses and green checkmarks<sup>[4],[5],[6]</sup>
- OK button<sup>[8]</sup>
- Cancel button<sup>[9]</sup>
- Vertical scroll bars, one for parameters and one for constraints<sup>[11]</sup>
- Horizontal scroll bar for constraints<sup>[20]</sup>
- Navigation buttons<sup>[13]</sup>
- Freeze button<sup>[14]</sup>
- Unfreeze button<sup>[15]</sup>
- Configure button<sup>[16]</sup>
- Zoom to button<sup>[17]</sup>

3. All constraints that are not violated but containing the selected parameter are hidden. When the user selects another parameter, the parameters and list of constraints will be updated.

4. The red cross in front of a constraint indicates the constraint is violated; a green checkmark indicates the constraint is not violated.

5. The red cross in front of a parameter indicates the parameter is used in a violated constraint; a green checkmark indicates the parameter is not used in a violated constraint.

22. Red background in constraint detail indicates the constraint is violated; green background indicates the constraint is not violated.

6. The user can change parameter values and locks just as in the data editor, changing these however does not invoke the configurator. When the user changes the parameter’s value the red crosses/ green checkmark will be updated, unless it frozen<sup>[14],[15]</sup>.

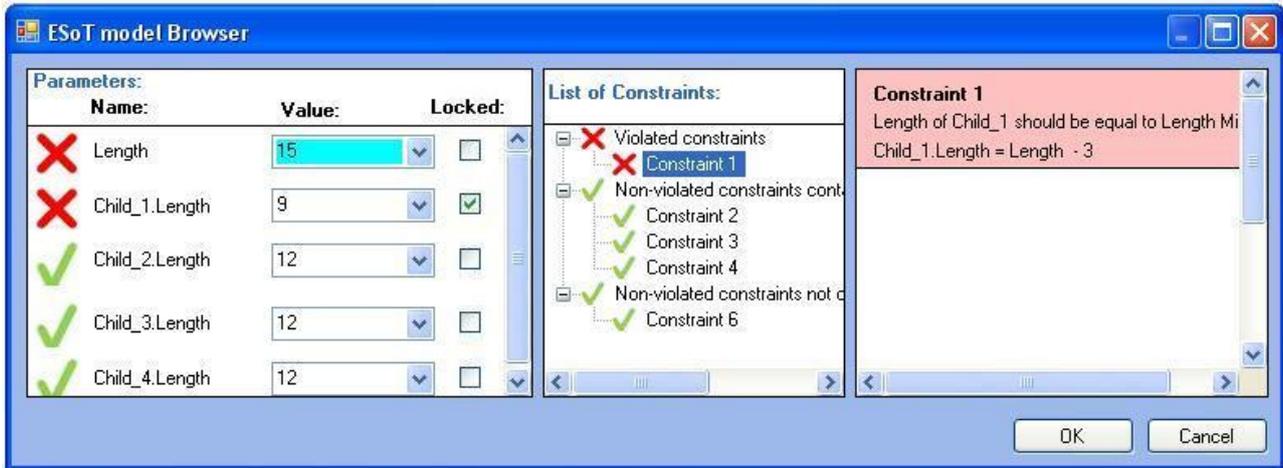
23. The Constraint’s detail will show detail of the constraint when the user clicks name of the constraint in the List of Constraints. The user can choose more than one constraint. The selected parameter will highlighted in the detail information of the constraint.



7. The constraints will be grouped by:
  - Violated before non-violated constraints.
  - Constraints contain the selected parameter, before constraints that do not have.
  - Constraints name in ascending order.
21. The parameters will be sorted by:
  - Violated before non-violated parameters and then other violated that have no connection to the selected parameter.
  - Constraints name in ascending order.
8. The Ok button is to commit all changes. When Ok button is clicked, if there is at least one violated parameter, than message box will appear to notify the user and ask if the user wants to proceed or cancel it. If the user proceed then ESoT model browser will commit all changes even though there are some invalid value and will automatically be closed, if it not then ESoT model browser stays. If all parameters are valid then ESoT model browser will automatically be closed.
9. The Cancel button is to cancel all changes and restore all the values to original value when the ESoT model browser was opened. When the Cancel button is clicked, the ESoT model browser will automatically be closed.
10. The Minimize button is disabled. When the zoom to button is clicked than ESoT model browser will automatically be minimized.
11. The Vertical scroll bars are automatically shown if there are too many parameters and constraints to fit on the panels.
20. The Horizontal scroll bar is automatically shown if there is too long constraint to fit on the panels.
12. All the buttons at the top of the window(navigation, freeze, zoom to, show) will have icon.
13. The Navigation buttons work similar as undo and redo. When the user selects or changes a parameter value or lock a parameter, "<" button will be enabled in order to allow the user to restore value of the parameter before he change it and ">" button will be enabled when the user click "<" button.
14. When the Freeze button is clicked then the system will enter the Frozen state. In Frozen state:
  - Background color of Parameters will change in order to notify the user that he is in Frozen mode.
  - The user can change more than one parameter and the system will not invoke the configurator and validator.
  - When the user click the OK button, it will automatically leave the frozen state and do the requirement [8](#).
  - The cancel button still work the same [9](#).
  - The "<" button will restore all value of the parameters into the value when freeze button is clicked.
  - The configure button [16](#) will work the same as in unfrozen state but in frozen state it will not invoke the configurator and validator.
  - The zoom to button still work the same [17](#).
  - The freeze button changes to the unfreeze button.
15. The Unfreeze button updates parameters and constraints [4](#),[5](#),[6](#),[7](#) based on the changes that have been made in freeze time. When Unfreeze button is clicked it will change to Freeze button.
16. The Configure button has a function that automatically tries to give the user the valid value without changing the locked parameters. If there is a valid value than system will change it, if not the parameter value will remain the same.
17. The Zoom to button is used to zoom to the smallest system component containing all listed parameters and select the component of the selected parameter in ESoT model browser.



## APPENDIX III: INITIAL DESIGN REQUIREMENT DOCUMENT



### Requirements:

1. “ESoT model browser” maybe covers the function of the window better than “Constraint violation window”.

2. The window shows:

- The selected parameter (blue in the picture above),
- List of Constraints.
- Constraint’s detail<sup>[23]</sup>
- All parameters used in these constraint.
- All other parameters are violated.
- Red crosses and green checkmarks<sup>[4],[5],[6]</sup>
- OK button<sup>[8]</sup>
- Cancel button<sup>[9]</sup>
- Vertical scroll bars, one for parameters and one for constraints<sup>[11]</sup>
- Horizontal scroll bars for List of constraints and Constraint’s detail<sup>[20]</sup>

3. All constraints that are not violated but containing the selected parameter are hidden. When the user selects another parameter, the parameters and list of constraints will be updated.

4. The red cross in front of a constraint indicates the constraint is violated; a green checkmark indicates the constraint is not violated.

5. The red cross in front of a parameter indicates the parameter is used in a violated constraint; a green checkmark indicates the parameter is not used in a violated constraint.

22. Red background in constraint detail indicates the constraint is violated; green background indicates the constraint is not violated.

6. The user can change parameter values and locks just as in the data editor, changing these however does not invoke the configurator. When the user changes the parameter’s value the red crosses/ green checkmark will be updated.

23. The Constraint’s detail will show detail of the constraint when the user clicks name of the constraint in the List of Constraints. The user only can choose one constraint.

7. The constraints will be grouped by:

- Violated before non-violated constraints.
- Constraints contain the selected parameter, before constraints that do not have.
- Constraints name in ascending order.

21. The parameters will be sorted by:

- Violated before non-violated parameters and then other violated that have no connection to the selected parameter.
- Constraints name in ascending order.



8. The Ok button is to commit all changes. When Ok button is clicked, if there is at least one violated parameter, than message box will appear to notify the user and ask if the user wants to proceed or cancel it. If the user proceed then ESoT model browser will commit all changes even though there are some invalid value and will automatically be closed, if it not then ESoT model browser stays. If all parameters are valid then ESoT model browser will automatically be closed.
9. The Cancel button is to cancel all changes and restore all the values to original value when the ESoT model browser was opened. When the Cancel button is clicked, the ESoT model browser will automatically be closed.
10. The Minimize button is disabled.
11. The Vertical scroll bars are automatically shown if there are too many parameters and constraints to fit on the panels.
20. The Horizontal scroll bar is automatically shown if there is too long constraint to fit on the panels.