Model Based Systems Engineering in Warehouse Design

Automatic report generation from SysML models.

Internship report Teun Plegt
Fraunhofer IML
01-11-2010 – 31-01-2011

Model Based Systems Engineering in Warehouse Design

Automatic report generation from SysML models.

Name:	Teun Plegt
student number:	S0075841
Period:	01-11-2010 - 31-01-2011
Host Institution:	Fraunhofer Institut für Materialfluss und Logistik
	Abteilung Intralogistik und –IT Planung
	Dortmund, Germany
	Supervisor: Diplom Logistikus Michael Schmidt
Home institution:	University of Twente
	Faculty Engineering Technology
	Department Production Management
	Enschede, the Netherlands
	Supervisor: Ir. Ronald Mantel

Preface

This report was written in the context of a three month internship at the Fraunhofer Institute for Material flow and Logistics IML in Dortmund. During my time at Fraunhofer I conducted research related to a project of determining the way warehouses are to be designed in the foreseeable future.

I would like to thank Mister Michael Schmidt for his help, advice and supervision during my three months at Fraunhofer, Mister Eric Venn for his constructive feedback and of course Mister Ronald Mantel for supervising me during my Master and opening the doors that ultimately enabled me to do this internship.

Teun Plegt

Date: 28-01-2011

Management Summary

As a part of a bigger research project to develop a new warehouse design process using model based systems engineering, research has been done on the automatic generation of a requirements report from a SysML instance model of a warehouse.

The research involved the creation of an instance model of a warehouse from an existing requirements report as there is no existing model yet as the proposed warehouse design process is still in a conceptual phase.

To create a warehouse model in SysML a structure was developed to contain all warehouse elements and requirements whilst keeping the model clearly structured and easily readable. The structure was tested with several different types of warehouses and was found to be very useful and might in the future be used for other applications.

The created model was used as a basis for automatically generating a requirements report. The report generation was performed using the report generator tool from MagicDraw UML, the used SysML modeling environment. The report generator uses a template written in the Apache Velocity Template Language to collect data from the model and present them in a report. An existing template was heavily modified to get to a satisfying requirements report.

The resulting report was found to be quite useful. It showed all requirements in a clear way that was easily readable and very well suited for the envisioned target audience as it reads like a checklist. With some small modifications the report could be used in practice and might even replace the existing requirements reports.

The automatic report generation is also found not to be a bottleneck in a warehouse design process that uses model based systems engineering and fully supports the envisioned new warehouse design process.

Contents

Introduction

Warehouse design is a very specific field of work. It involves extended knowledge of the subject and a thorough consideration of all parameters involved to get to a proper warehouse design that allows for efficient storage of goods with minimal overhead and enables high throughput and fast delivery.

Today, warehouse design is done by professionals with extended experience in the field who use their knowledge and insight to get to good results. This works, but the process of getting to a warehouse design can be lengthy and results depend highly on the designers involved.

In a project to determine the way warehouse are to be designed in 2020, a systems engineering perspective is used to look at the subject. The use of models should allow for the preservation of knowledge and enable faster design with more reliable results.

The project discussed in this report is a small but nonetheless vital part of the bigger project of developing the new warehouse design process. This project will address the question how the results of a warehouse design process that involves modeling could be used and presented.

Background

Warehouse design in 2020

The goal of the overall project is to come to a faster, leaner, more accurate way of designing a warehouse through the application of Model Based Systems Engineering. Systems engineering and modeling will play a major role in this as it should enable fast, robust design of the complex system that a warehouse basically is. The project is a cooperation between the Fraunhofer Institute for Material flows and Logistics in Dortmund, Germany and the Georgia Institute of Technology, Atlanta, United States.

The project addresses the following topics:

- The creation of an engineering discipline of warehouse design.
- The exploitation of emerging technology in the software industry.
- The re-conceptualization of the warehouse design process.
- The treatment of warehouse design as a systems engineering problem.
- The suggestion that warehouse design is fundamentally based on models.

Systems Engineering

According to the International Council on Systems Engineering, Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem [1]. Frequently, models are used to design and analyze complex interdisciplinary systems. It enables capturing all characteristic system elements, connecting them and conceptualizing systems in any abstract level. The models can be analyzed and improved and be used to give a detailed description to eventually realize the modeled system. Systems engineering originates from software development but was later adapted by for example NASA and airplane manufactures to develop complex systems like satellites and commercial airliners.

SysML

To model systems in a consistent and logical way that is easily understandable a common modeling syntax has to be used. Since 2006 there is a modeling language specifically aimed at systems engineering called the Systems Modeling Language (SysML). It is a visual modeling language derived from UML (Unified Modeling Language) which has been the standard software modeling language for several years. The specifications of SysML are defined by the Object Modeling Group (OMG) and are regularly updated. [2]

SysML uses diagrams and visual elements like blocks and connections to fully describe the model of a system. The language is quite easy to understand, but modeling with it might proof more difficult because of the design decisions that have to be made to come up with a correct model. As SysML is quite new (established in 2006) there are not a lot of modeling guidelines and best practices available and the use of it in the warehouse market has no precedent. However, it shows great potential in interconnecting the different subsystems of a warehouse, like ICT and mechanical systems, that could enable major improvements in warehouse design.

Advantages of SysML

- It can be customized to the warehouse design, producing a warehouse specific language
- SysML can also be used to model the design workflow
- SysML supports the description of both structure and behavior
- SysML models can be integrated with a variety of solvers

In the future SysML can be customized to warehouse specific needs and a library can be created with all common warehouse elements. The integration of computational solvers could enable computing of the expected performance of the proposed warehouse. The warehouse design process would then only involve gathering all the necessary warehouse elements from the library, connecting them and entering values for all relevant parameters which should result in an accurate model of the desired warehouse.

Modeling with SysML is possible with several different softwaretools that enable easy application of the language while correctly excecuting the language specific syntax. One of those tools is MagicDraw of No Magic Inc. A tool designed for modeling with UML but with an additional plug-in also capable of modeling with SysML. At the moment MagicDraw is by far the most widely used tool for modeling with SysML. [3]

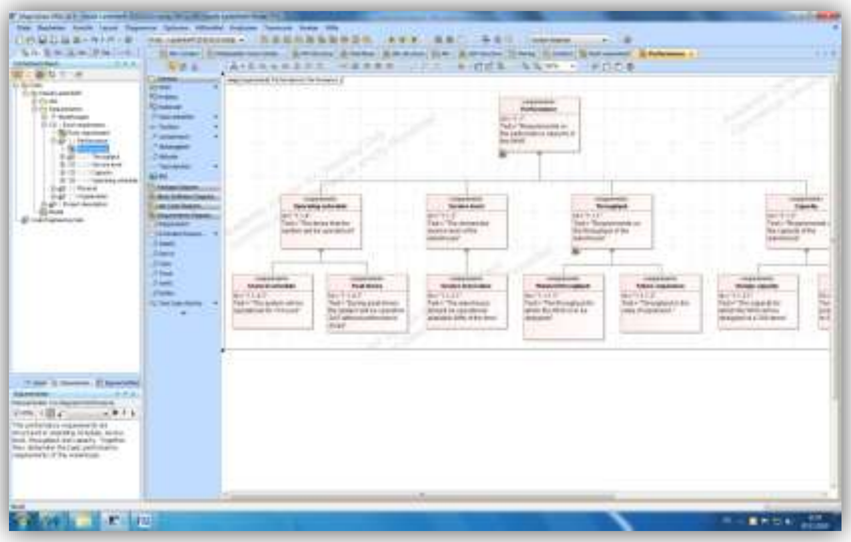

Figure 1, The MagicDraw modeling environment

Generic Warehouse Model

To aid in the warehouse design process, a generic warehouse model will be created. With the generic model it should be possible in the future to describe and model every single type of warehouse just by using model elements from the generic warehouse model.

To aid in the design process, a model will be realized with three parts, the design model, the workflow model and finally an instance model. The general idea is to capture raw customer requirements in the design model, rework them in the workflow model*** to end up with the final requirements and description for the warehouse in the instance model.

The division in three models is currently state of the art and none of the models is fully defined yet.

Requirements report

When a new warehouse is planned and designed, as is done by Fraunhofer IML for their customers, the results are presented in a requirements report. The report is then handed out to several vendors who are asked to make a bid on the construction of the warehouse described in the report. The intended reader of the report is also a warehouse specialist himself who has to be able to make a complete detailed design of the warehouse described in the report.

The report also contains detailed specifications about capacity, throughput, IT and all other, to the customer relevant, specifications. It usually contains appendices for specific subjects like technical details and the warehouse management system.

Problem definition

The proposed warehouse design process is still in a highly conceptual phase. There are a lot of unknowns and there might be several undiscovered complications. A part of the process will be to translate the warehouse instance model, which will be the result of the new warehouse design process, to a format more suitable for distributing to vendors like a requirements report. This is only a small part of the whole process but nonetheless vital.

Regarding the results of the warehouse design process, the following questions have been defined.

Research question

What would the end-result of a warehouse design process, using system engineering, look like and how could this replace the existing requirements reports?

Sub questions

- Is the generation of results a bottleneck regarding the whole design process?
- Could the new design process involve improvements on the existing requirements report?

General approach

To determine the end-result of the new warehouse design process, there should be a model of a warehouse to start from. As there is no complete warehouse modeled just yet, it was decided to reverse engineer a requirements model from an existing requirements report. This could provide insight in the desired format and structure the resulting model should have to be able to convey all relevant information in a structured manner.

When the existing requirements report is integrated in the model, an attempt should be made to get it “out of the model” as would be the case in a real warehouse design process. It should be done in a time-efficient way and the result should be suitable for distributing to vendors.

Modeling an existing requirements report in SysML

The first step is modeling an existing requirements report in SysML and finding a structure that is generic so it can hold for different requirements reports.

Warehousing in SysML

A SysML model usually consists of several types of diagrams and elements. Model elements are usually depicted as blocks, the basic elements of SysML. The blocks are then defined in Block Definition Diagrams (BDD) which show what the block consists of and how it relates to other blocks. In warehouse design, a block could be for example an AS/RS, a conveyor or a unit of handling. The inner workings of a block are defined in Internal Block Diagrams (IBD) in which ports are defined through which material or information can flow in or out the block. More information can be added to a model by means of activity, state-machine and parametric diagrams. Requirements are also elements in SysML and can be shown in Requirements Diagrams. [3]

As mentioned before the new warehouse design process will consist of three models. The design model, which is used to capture customer requirements for a new warehouse. The workflow model, which is used to translate and rework the design model to finally end up with the instance model. The instance model is the final description of the proposed warehouse and therefore is the model we will have to try to model ourselves to see what it possibly could look like.

The instance model should contain all information necessary to describe a real world warehouse. It contains all the elements and all the requirements that have to be fulfilled. The question is how to structure the model and especially the requirements in such a manner that it is generic enough for several different warehouses and clear enough to easily view and interpret the model.

Structuring

SysML requirements can be connected by several relationships of which “containment” is the most common. With the containment relationship a requirement can be regarded as a child of its parent requirement. Building on this, a whole containment tree can be created, starting with top-level requirements and dividing down the way in more detailed, specific requirements.

The approach to branching the tree from the top level “root” requirement to the detailed requirements is hereby of importance as it is preferable to have one structure that can be reused for several different types of warehouses. Warehouses consist of a great number of different elements and finding a structure that holds for an automatic storage and retrieval system as for a small manual warehouse is quite difficult. The correct and logical structuring is important as the ultimate goal of the project is to create a library of warehouse elements including their requirements. When combining the different elements the standardized structure of the requirements should prevent the model from becoming chaotic.

The process of finding a structure is more or less by trial-and-error. When a new structure is proposed it is tried to include different elements of an existing requirements report in the structure and it is checked whether it is a good fit or not.

Functional and non-functional

When looking in literature one encounters several possibilities of structuring requirements. Common practice in software modeling with UML is to divide requirements in functional and non-functional. [6] Where functional requirements are about what a system should do, like “store goods”, and non-functional requirements are about what a system should be, for example “fast” or “inexpensive”. However, this division does not make a lot of sense for a real life system as a warehouse as one would end up with only a few functional requirements and more than 90 percent non-functional requirements. Rendering the structure effectively useless.

The functional non-functional division clearly originates from software development as software is by definition a tool that should serve a specific functional purpose. The requirements about what the system should do are by far the most important. There are still enough non-functional requirements (the system should be reliable, fast and easy to work with) but not being a real life physical object effectively limits the amount of non-functional requirements. When looking in a requirements report of a warehouse the functions are actually quite clear from the start and almost all of the report is about technical details and other descriptions of what the system should be, not what the system should do.

Performance, Organization and Physical

It was then tried to find a structure on the basis of the nature of the requirements. Of all the different possibilities a division at the top level in “performance”, “organization” and “physical” looks the most promising. In this structure, all performance related requirements, like capacity and throughput are grouped under performance. Organization contains all requirements that are not physical and not about performance and physical contains all the physical requirements. This manner of structuring seemed logical as this division groups the requirements in clearly defined groups. This structure was tested and seemed to be able to contain the requirements of several different requirements reports.

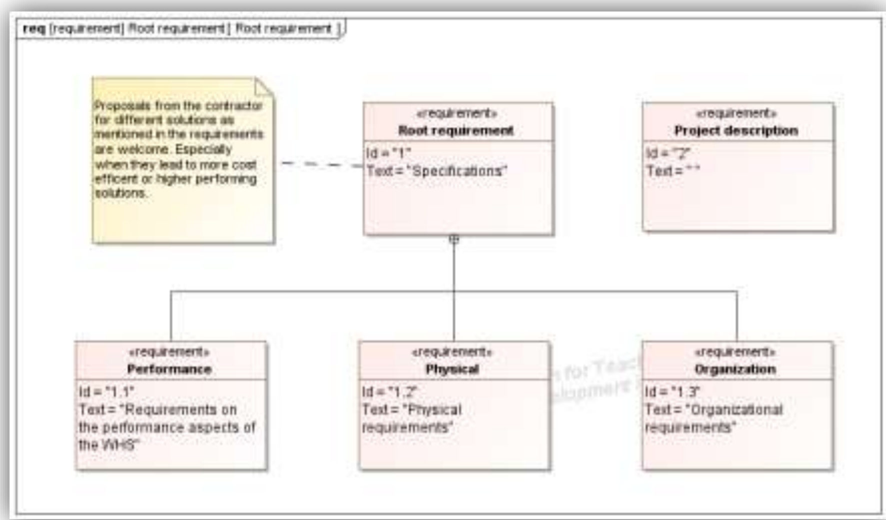

Figure 2, requirements structure

However, it is not that convenient to have all requirements in one big tree. Most requirements can, and should, be linked to structure parts in a model, conveyor requirements should for example ideally be linked to the conveyor parts in the model to convey that the conveyor parts satisfy the conveyor requirements.

Recurring Structure

The International Council on Systems Engineering (INCOSE) employed two Model Based Systems Engineering challenge teams that researched the implementation of SysML in real life systems to come up with a systematic approach and best practices in SysML modeling. In their model of a telescope system [4] they apply a recurring structure where a system is divided in physical parts and all these parts contain requirements, structure and behavior. Sub parts are then modeled under the structure of the parent part and also contain requirements, structure and behavior. It was chosen to use this approach as it allows for a clearly structured system that is easy expendable and allows for extensive models. The requirements of the several parts were then modeled with the earlier suggested performance, organization and physical division.

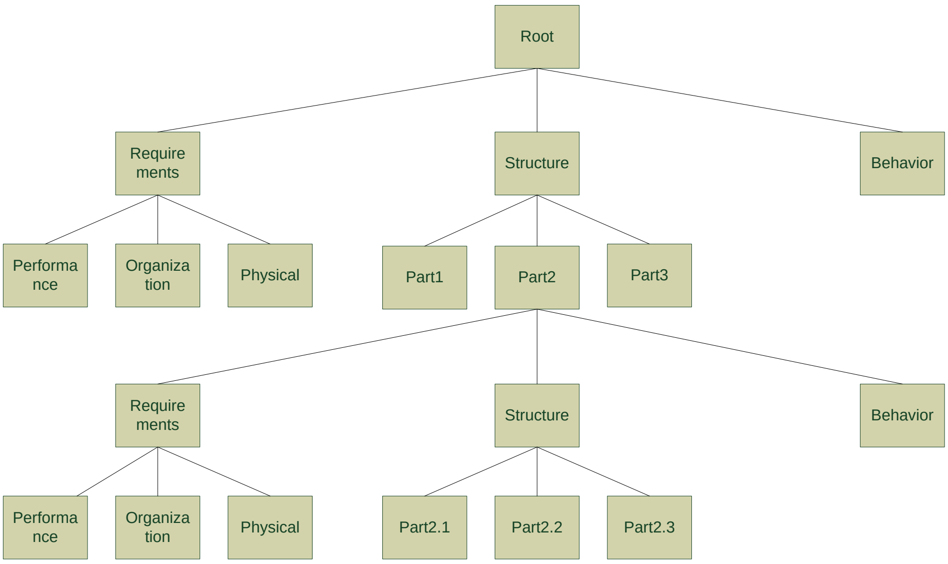

Figure 3, the recurring structure

This all leads to a structure with a quite large amount of elements but when the structure is used consistently everything is easy to find, even when the model grows extensively. All model elements related to one part are also grouped together in the structure.

The model

As a proof of concept an existing requirements report was modeled using this structure. The used requirements report is of a warehouse of a firm producing valves and fittings that was in need of a new warehouse. The requirements report describes how an Automatic Storage and Retrieval System should be implemented and used in the firm.

All the requirements that are described in the requirements report were derived from the document and entered in the SysML model as requirements. The requirements were grouped and relations were added. Using the proposed structure this lead to a division in the following sub-parts:

- AS/RS
  - Alleys
  - Manipulation Area
    - Conveyor
    - Workstation
  - Shuttle (RBG)
  - Unit of handling

In this structure all relevant information can be contained. Requirements are implemented in the folder of the corresponding part when possible and more general requirements move up the hierarchical tree.

When it was proven that the structure could contain the requirements report for an automatic storage and retrieval system, other requirements reports with different warehouse set-ups were tried to further test the chosen structure to be able to contain any model.

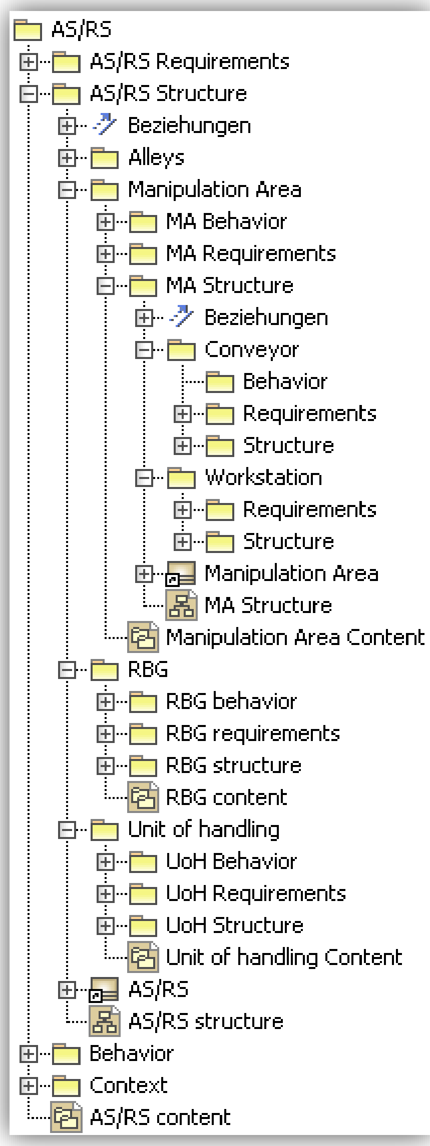

Figure 4, the model structure as shown in the containment tree

Report Generation

"Create a MagicDraw template to convert the SysML requirements model to a Text document to be used as a requirements report."

Report generation in MagicDraw

A quick and easy way of generating a requirements report is a vital part of the new warehouse design process. The model is only of little benefit when a separate requirements report still has to be written to convey requirements to customers and contractors. A fast automatic way of translating a model to a written report is also a necessity.

The modeling tool that is used for SysML, MagicDraw UML from NoMagic, features a report generator that is able to collect data from the model and represent that in a text report with a proper lay-out [3]. MagicDraw has a standard requirements report template that already does a quite good job in displaying the modeled information. It features a table of contents, different bookmarks for every level in the model and it shows the appropriate diagrams.

However, the standard generated report also leaves a lot to be desired. It is very hard to read as all requirements are displayed in a large list and despite of the different style bookmarks and different Id´s it is hard to understand the structure and logic behind the different requirements. Next to that the report fails a title page, introduction pages and also shows irrelevant diagrams.

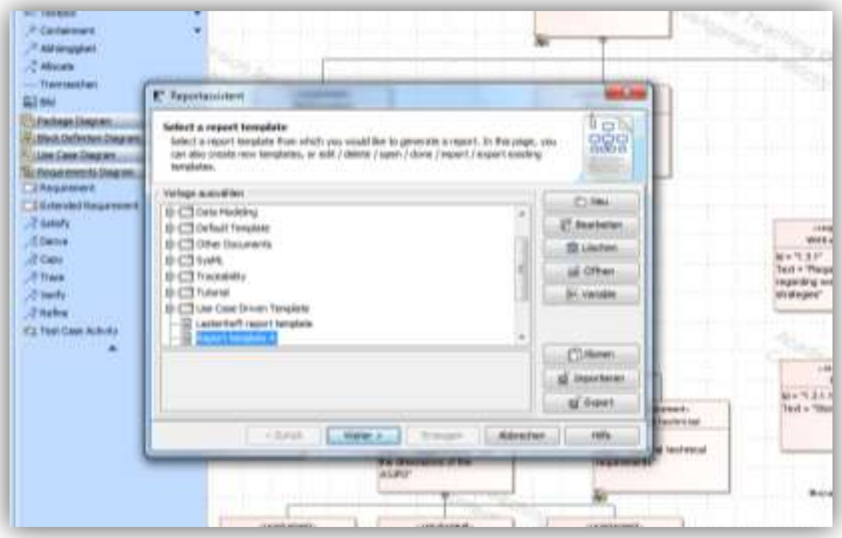

Figure 5, the MagicDraw Report Generator

The Velocity Template Language

The report generator of MagicDraw uses the Velocity Template Language (VTL) by the Apache software foundation to generate reports [7]. VTL uses a text based template file that contains the lay-out and the code to collect the appropriate information from the model and to display it in the correct spots of the template. The idea is that only the template is to be changed and that when it is satisfactory it can be used to output a report from different models without needing to change the template.

The use of VTL makes it relatively easy, but still quite cumbersome to modify the template. To create your own template of the level of sophistication of the build-in templates, a degree in, or at least affinity with, programming is necessary. A professional template easily consists of 15 pages of code and as the code is written in an ordinary text editor, debugging the code is extremely difficult and cumbersome. However, it is possible to modify the existing templates by reusing code and combining different templates to end up with a template that gives more or less satisfactory results.

```
                #foreach ($rat in $rationalGroup.get($object.ID))
                $rat.body
                #end
#end
#if ($problemGroup.contains($object.ID))
        Problems:
                #foreach ($prob in $problemGroup.get($object.ID))
                $prob.body
                #end
#end
#if ($commentMap.containsKey($object.ID))
#set ($commentGroup = $commentMap.get($object.ID))
#foreach ($name in $sorter.sort($commentGroup.groupNames()))
        $name:
                #foreach ($com in $commentGroup.get($name))
                $com.body
```

Figure 6, Example of code from a template written in the Velocity Template Language

The major challenge of generating a report from a model is to not lose the structure and overview. The requirements in the model consist of a lot of elements that are related to each other by a parent child relationship which can consist of over 7 levels of parent child relationships. When converting this to a text report the overall structure is easily lost as the only tool to make the structure clear to the reader is the use of different style bookmarks.

The big problem with the report is that when working with models with up to seven levels, overview is quickly lost as there is no way of clearly displaying the hierarchy within the model with only different style bookmarks.

1.2.2.3.8.2 Fastening
Requirements on fastening elements

1.2.2.3.8.2.1 Bolted connections
Bolted connections should be used in combination with a security nut according to DIN 985

1.2.2.3.8.2.2 Connection with the building
All connections with the building should be bolted or clamped

1.2.2.3.8.3 Electrics
Requirements on electrics

To improve the readability of the generated requirement report it was tried to stop displaying requirements as text, starting from level 4, and instead display them in a table. This was achieved by combining the standard templates “requirement report” and “requirement table” where additional efforts had to be made to get the tables to show up inside the structure of the requirement report, as to not disrupt the order in which the requirements are displayed.

However, when using the new structure where the requirements are not in one big tree but in several different trees belonging to the several sub-parts, a model probably no longer has trees that have enough levels of requirements to justify displaying them in tables. But it is still a possibility when necessary.

To get to a generated requirements report that closer resembles an original requirements report it was tried to include the introductory chapters of a requirements report in the generation of the report. The template will use a second text document that has sections for an overview, summary etc. that will have to be written by the writer of the report. During generation the report generator will find the links to this template in the document and include the sections of text were necessary. The process will now include a model, a fixed template and a text file.

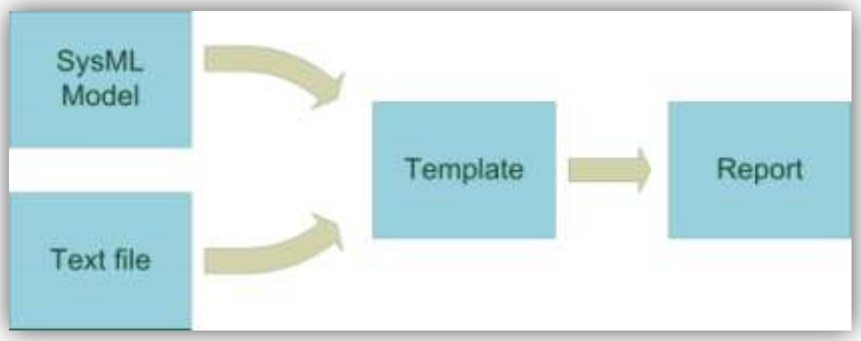

Figure 7, Report generation by a template

Results

The results of the modeling process and the report generation are discussed below.

Modeling

As the warehouse design process does not exist yet, the creation of an instance model of a warehouse was quite a challenge. The big questions were what the model should ideally look like and how it was to be modeled in SysML. Of course there are numerous approaches of creating the model but the goal is to show that it is possible, not to get the perfect model directly. It is foreseeable that the future instance model will look quite different from the in this project created model when it really is the result of the warehouse design process and the warehouse design workflow.

The modeling of the instance model is also to be seen as a proof of concept, but there are numerous lessons to be learned from it. First of all that it is very well possible to model a requirements report in SysML and that it is quite a comprehensible way of using and displaying the data. The proposed structure also works quite well as it is able to contain all the data in a structured manner, disregarding the size and complexity of the model.

Report generation

The generation of the report works quite well. The procedure in itself is quite easy and only takes a short amount of time, somewhere between 30 and 45 seconds. The creation of a proper template is the most difficult part but modifications can be made by anybody with a little programming experience and sufficient time. When a satisfying template is created it can be used to generate different reports from several models. This makes it very time-efficient.

The question is whether the generated report is good enough to replace existing requirements reports or even is of any use at all. After all, one can create the best template that shows everything in an optimal manner; the displayed information is, at most, just as good as the information that is in the model. All requirements in the report are short textual descriptions that are taken from the requirements elements from the model and will be displayed as such in the report. Does the translation from model elements to report requirements lead to a readable document?

The resulting document contains all requirements, clearly structured, with the necessary diagrams to easily get a good overview and even contains a table of contents and all textual segments. The question remains whether it is a useful report in the practice of warehouse planning and design.

Feedback

To validate this possibility and to judge the qualities of the generated report, the whole process and the resulting report were presented to the team leader of Warehouse Planning at Fraunhofer IML to get feedback on the process and show which possibilities and opportunities the SysML modeling, and the report generation in particular, could have.

This feedback session resulted in some very useful information. The report itself was considered to be quite promising. The different appearance of the report could actually be quite welcome for vendors, the actual readers of the report, as requirements were clearer as they did not have to be derived from the text. The report could be read as being a kind of checklist, which is actually how these reports are used by vendors when they produce their tender. A test to use a SysML model instead of, or next to the Microsoft Excel sheets used today to get feedback from tenders will be considered.

The diagrams in the report were deemed to be quite useful as they helped navigate through the document. Requirement matrices that connect requirements to specific warehouse structural elements were probably not that important. They don´t really show anything a warehouse planner doesn´t know. Also, the lay-out of the report needed some work.

The proposed structure of the model was deemed to be very useful. It´s a promising way of displaying the description of a warehouse and might be used in the future for different applications. The creation of a large poster version of the structure in Microsoft Visio was considered.

The biggest problem that prohibits acceptance of using SysML directly as a tool for warehouse planning is the MagicDraw modeling software as it lacks user friendliness. A different front-end and a simpler user interface could help solve that problem and the development of this interface should be considered.

Conclusion

In this project the automatic generation of a requirements report from a SysML model was researched. The project is a part of a bigger project to develop a new warehouse design process that should bring improvements in speed and accuracy to the design of warehouses.

In an attempt to recreate a part of the proposed warehouse design process, a model was created from an original requirements report in SysML. It is shown it is very well possible to use SysML as a tool to model and display this kind of information. The structure that is proposed to contain the information is a useful approach to modeling a warehouse and might be of use even when disregarding the SysML modeling.

The automatic generation of a requirements report was shown to be very well possible and the resulting report to be a useful, or at least promising, document. Although not perfect, the document does possess some qualities that might even be called an improvement over the existing report. The structured and clear presentation of the contained information is very promising and might be used in the near future to convey requirements to warehouse vendors.

In relation to the in development warehouse design process it is clear that the report generation will at the very least not be a bottle neck in the whole process. When the process is going to turn out as proposed and envisioned, warehouse design might very well be done by dragging and dropping warehouse elements from a library in a software tool. Specifications are automatically generated and solved and when finished the requirements report is generated by the press of a button.

Further Research

The new warehouse design process is still in a highly conceptual phase. There is still a large amount of general research to be done before the process will get at a state in which it will be of benefit to the warehouse designer.

Closer related to the subject more research could be performed on the final instance model and the format and structure it should be in. My research proposed a possible format that could be used, as far as our knowledge goes today, but when more research is done on the whole process and the warehouse design workflow is implemented the chosen format should be checked to be consistent with the warehouse design workflow.

The report generation template created works quite well but the inclusion of activity, state-machine and/or parametric diagrams should be considered. Although this might ask for a lengthy coding process as the template has to be modified considerably. Next to that the templates should be checked on other types of models. The reports that were generated in the context of this research were all technical appendices. A model that describes the warehouse management system for example might ask for a completely different template. Of course the ideal is to have one template that is suitable for all models. Further research should reveal whether that is possible.

To enable warehouse planner to more easily interact with and create a model, a special front-end might be developed to enable non-SysML experts to make use of the possibilities SysML offers.

References:

[1] What is systems engineering? The International Council on Systems Engineering, http://www.incose.org/practice/whatissystemseng.aspx

[2] Official homepage SysML homepage, http://www.omgsysml.org/

[3] "A Practical Guide to SysML, the systems modeling language", Friedenthal, Moore and Steiner 2009 ISBN: 978-0-12-378607-4

[4] Online SysML model of a telescope system, das Gesellschaft für Systems Engineering http://mbse.gfse.de/documents/36.html

[5] Online car access model, website of Tim Weilkiens http://example.system-modeling.com/

[6] Erfolgreich mit BPMN und UML ein Hafensystem modellieren, Scheithauer 2009

[7] Apache Velocity User Guide, the Apache software foundation http://click.apache.org/docs/velocity/VelocityUsersGuide.pdf

[8] Der Prozess Lagern und Kommissionieren im Rahmen des Distribution Center Reference Model (DCRM), Wisser 2009

Appendix

Fraunhofer IML

Fraunhofer is Europe's largest application-oriented research organization and is involved in research in almost every field. It works closely together with commercial and public partners to develop knowledge, new products and techniques all geared to human needs.

The Fraunhofer association consists of more than 80 research units and 60 Fraunhofer Institutes all over Germany and employs almost 17,000 people of which the majority are qualified scientists and engineers.

In Dortmund the Fraunhofer Institute for Material flows and Logistics is located which is involved in research in logistics, warehousing and related topics. The institute employs 400 and is partly responsible for the logistics chair of the TU Dortmund.

At the department Intralogistics and IT-planning research is performed on warehouses and several consultants are employed who assist customers with building new warehouses and reorganizing old ones.

At Fraunhofer IML many projects are performed for customers who are in need of a warehouse. This might be a completely new warehouse or a renovation of an old one. As warehousing is a very specific field of work, most companies in need of a warehouse have no knowledge about warehousing themselves and have difficulties to translate their needs for storage to warehouse specifications. Of course there are a lot of players in the market who would be interested in the job of constructing a warehouse but for a company without specific warehousing knowledge it is difficult to find the right partner and find the best deals available.

Consultants of the department of internal logistics and IT-planning assist companies in this process. They analyze the storage needs of a customer and write a report that contains all requirements for the warehouse that is to be build. This report is distributed to several vendors of warehouse solutions who are asked to make an offer on the warehouse that is proposed in the requirement report.

Reflection

A personal evaluation of my experiences during my internship.

Learning topics:

Warehousing

Although I had several encounters with the subject of warehousing during my studies, the three months at Fraunhofer IML gave me a much better understanding of what is happening in the field of warehousing than I had before. The WMS market report, the conversations with colleagues, the project reviews; they all helped to get a better grip of the subject.

SysML

The Systems Modeling Language is an interesting development that has the potential to become more dominant in the future. During my stay at Fraunhofer I got a thorough look at the language and although I cannot say I know the language inside out, I am able to read a SysML model properly and to create simple models by my own.

German

One of the reasons to do an internship in Germany is of course to learn German. As I was already able to understand most of the German I mostly tried to extend my own German speaking vocabulary. I definitely improved my German during my stay but I did not make as much progress as I hoped. This is mostly due to the fact that most Germans speak English quite well and therefore I had the possibility to also express myself in English. Perhaps more often than I wanted I switched to English to make myself clear or to discuss subjects with others. It would have been better for my German to talk German all the time but one should remember that the purpose of communicating is not always to improve language skills but mostly to be able to understand each other. When I noticed that my German was preventing me from making something clear to my audience I switched to English as to not disturb the discussion.

Research

In the standard curriculum of the program Mechanical Engineering at the University of Twente, research is not one of the dominant subjects. Therefore it was quite interesting to work for three months in a research environment.

The research project that my internship was a part of could be considered quite fundamental research. The use of systems engineering and SysML in warehousing has no precedent and the research is in a highly exploratory phase. This both made it interesting and sometimes frustrating to work on the project. It is interesting to be part of something new and to explore previously undiscovered grounds but the purpose and direction the assignment should head was not always clear. This lead to numerous occasions where the direction headed was deemed to be wrong and another direction was chosen. This is of course an integral part of performing research but nonetheless quite frustrating as it sometimes means finished work is no longer of value.

Personal evaluation

Preparation

As to prepare for my internship, the three weeks before the start I made a beginning with studying the visual modeling Language, SysML, that I would be using during my assignments. To this end I used the book “A Practical Guide to SysML” from Friedenthal, Moore and Steiner. From this I only learned the very basics of the language, as it is almost impossible to learn driving a car from a book; it is very difficult to master the language without using it. But it gave me a quite good understanding of the concept which made it a lot easier when I finally could work with the real program.

At my start at Fraunhofer there were several documents to work through which brought me up to speed with Fraunhofer IML and, more specifically, the work being done in my department “Intralogistik und –IT Planung”. This included the Warehouse Management System market report 2010 and several reviews of finished projects.

Housing

It was quite difficult to find a room in Dortmund, as Fraunhofer did not have a lot of ways to assist me other than give advice and the TU Dortmund could not provide me with a room as I wouldn´t be an official student I had to look for a room on the internet. The TU Dortmund finally gave me a list with rooms from external parties and one of them had a room available at the start of my internship.

The room was located in an old student flat that was home to many African students. However, this room proofed to be a poor solution as I had to move to another room after one night because of a sanitation problem that involved insects…

Luckily the student housing service of the TU Dortmund agreed to make an exception and offer me one of their student rooms. This room proofed to be much nicer with my own bathroom and even a personal fridge in the room. Only the very unreliable internet connection was cause for some problems.

Social life

I decided to do my internship in Germany as to not overcomplicate the rest of my study program but still experience working in an international environment. Just a few months before I started my internship I came back from an Erasmus exchange semester in Sweden. After this international experience I decided that Germany was far enough for my internship and did not need to visit another continent for the experience. Next to that I believe Germany is an interesting market for a Dutch engineer and when mastered the English language the German language is a good language to learn to effectively expand your competences.

At Fraunhofer the working environment is quite informal and not at all as “German” as I expected. The German working culture was supposed to be quite hierarchical and formal but that is not at all as I experienced it. People, including bosses, spoke to each other on a first name basis and all communication was quite informal. This might be because of the connection to the University and the high number of student assistants and the fact that the average age of Fraunhofer employees is quite young.

As Dortmund is about two hours by train I have been going home to my corridor in Enschede almost every weekend. That doesn´t mean I did not have any social experiences during my stay in Dortmund. Two of my friends I got to know in Sweden studied in Dortmund and I visited the centre of the city with them on several occasions. As I was in Germany during the Christmas season I also had the opportunity to experience Christmas the German way. This means Christmas parties and in the case of Dortmund, the biggest Christmas market in the world, complemented with the biggest Christmas tree. Combined with the good relationship I had with my colleagues I would say my stay in Dortmund was quite pleasant.

However, the city of Dortmund itself was a little bit disappointing. Although the University is quite big, as far as I can tell after three months student life is not really dominant in the city. Most students in Dortmund still live at their parent's place which becomes apparent in the lack of student pubs in the city centre. When looking for a city that closer resembles the student life of Enschede, Münster or Bochum would be better choices. But Dortmund still has a lot to offer in terms of shopping and during the Christmas season the city flourishes.

Finance

As I didn´t had to book an expensive airplane ticket it was not a real big problem to finance my internship. From Fraunhofer I got a monthly fee that was just over the limit of the Twente Mobility Fund to receive a grand. However, I had no problem paying my room from the funds Fraunhofer provided me with. The rest of the money I used to pay for my train tickets to get home in the weekends which were quite expensive at 20 Euros for a one-way ticket. Luckily, discounts can be found when you look for them.