



**KTH Industrial Engineering
and Management**

MANUFACTURING SYSTEMS MODEL INTEROPERABILITY IN DISCRETE EVENT SIMULATION

Author

Natalia Svensson Harari

Supervisor

Dr. Daniel T. Semere

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Production Engineering and Management

Abstract

The use of simulation has been said to be a useful tool to analyze manufacturing systems, Discrete Event Simulation - DES for instance under the occurrence of different events. Information management in Manufacturing Systems is an important issue and so it is in simulation studies because some of the difficulties in building, reusing and integrating simulation models with other applications used in manufacturing systems are related with the data. In this context different efforts have been made to facilitate the use of simulation and overcome interoperability problems through improving the information management, one of this is the Core Manufacturing Simulation Data Information Model - CMSDIM developed by The National Institute of Standards and Technology - NIST.

The objective of this thesis is to contribute to the use of the CMSDIM in ExtendSim V8. A method to import databases structured based on the CMSDIM into the Simulation software ExtendSim V8 is developed and applied in a case study in a production line of SCANIA.

Keywords: *Manufacturing Systems, Discrete Event Simulation, Interoperability, CMSDIM.*

ACKNOWLEDGMENT

I would like to thank the Department of Production Engineering at KTH, for giving me all the facilities to study and develop my master's thesis.

I also thank my supervisor Dr. Daniel T Semere for his guidance during the development of this thesis work.

I would like also to thank SCANIA where I did the case study for allowing me to collect the data and to the workers in the production line for being willing to explain all that was needed.

I also thank Björn Hedin lecturer and responsible of the course XML för publicering at KTH who helped me to improve my understanding of the XML technologies.

Last but not least, I would like to thank my family for their love, encouragement, support and understanding.

Table of contents

Abstract	1
Acknowledgement.....	2
Table of contents	3
List of figures	4
List of tables.....	4
1. Introduction	5
1.1 Project context	5
1.2 Problem description.....	7
1.3 Related Work and Literature Review.....	8
1.4 Core Manufacturing Simulation Data Information Model – CMSDIM...	9
1.5 Objective.....	10
2. Outline	11
3. Development of the Method.....	11
4. Case Study.....	26
5. Conclusions, Discussion and Future Work.....	28
6. References.....	30
7. Appendix.....	32

List of figures

Figure 1. Information Flow Diagram in a Manufacturing System.....	6
Figure 2. Steps in a Simulation Study.....	7
Figure.3 CMSDIM Packages.	10
Figure.4 Outline of the thesis.....	11
Figure 5. Description of the Method	12
Figure.6 Production Planning Package.....	13
Figure.7 CMSDIM - UML representation of Process Plan Class and related classes.....	13
Figure.8 Generic Plan Definition in CMSDIM.....	14
Figure. 9 Generic Plan Step in CMSDIM.....	14
Figure.10 Simulation Model.....	15
Figure 11. Flowchart of the Method.....	15
Figure 12. Validation Process.....	16
Figure 13. Resources Class in CMSDIM	27

List of tables

Table 1. Heading Guide	18
Table 2. Field Format Description in ExtendSim V8	19
Table 3. Distributions in CMSDIM.....	22

1. INTRODUCTION

1.1 PROJECT CONTEXT

The changes in production occur for different reasons, but most of them in response to changing demands of the markets. The ability of the companies in adapting their production systems could mark the difference between competing, surviving or disappearing. In order to compete and survive the companies should be prepared to face the future challenges, so they must be able to in some way predict or be aware of different events.

Due to the globalization process the companies adopt strategies like produce in different countries or have suppliers in different regions of the world. This kind of strategy needs to be supported by efficient ways of information sharing and communication, so they have to take into account information systems, formats, programs versions, etc.

Manufacturing companies in this context are facing every time shorter manufacturing system life cycles. They need to be able to visualize different scenarios in production in limited periods of time i.e. in terms of infrastructure and data to support the decision-making process.

The decision-making process could vary from company to company, from centralized to decentralize according to their organization. However, it is supposed to collect information from different production and management areas.

The decision-making processes are for instance important in Production planning and scheduling. There will be analyzed the available resources i.e. workers and machines and their allocation, the necessary operations/processes and its sequence to produce a part/product, orders, due dates, workload and there will be information exchange, processing and communication between different functions and systems.

In relation with this, even though some companies could develop their own information systems, some of the most common systems are Master Production Schedule (MPS) and Enterprise Resource Planning (ERP). In the manufacturing systems there is a flow of materials but also a flow of information. The **Figure 1** shows an Information Flow Diagram in a Manufacturing System.

The accuracy of the data is important for the decision-makers. The activities of process planning and scheduling will also consider the system configuration in order to for example take into account the number of machines for a job and manage the due dates.

The manufacturing systems configuration could be divided in three models: series, parallel, and hybrid configurations models (series-parallel or parallel-series). [1]

When it comes to analyze changes in production, there are different valid ways how to visualize different scenarios. There are different techniques and tools that can be used depending on the analysis and purpose, and these could vary from animated presentations, technical drawings, data tables, diagrams, mathematical models, simulation models, etc.

Simulation allows modeling, representing approximations and describing real manufacturing systems in a period of time, and it has been said to be a useful approach for example in the design and analysis of manufacturing systems.

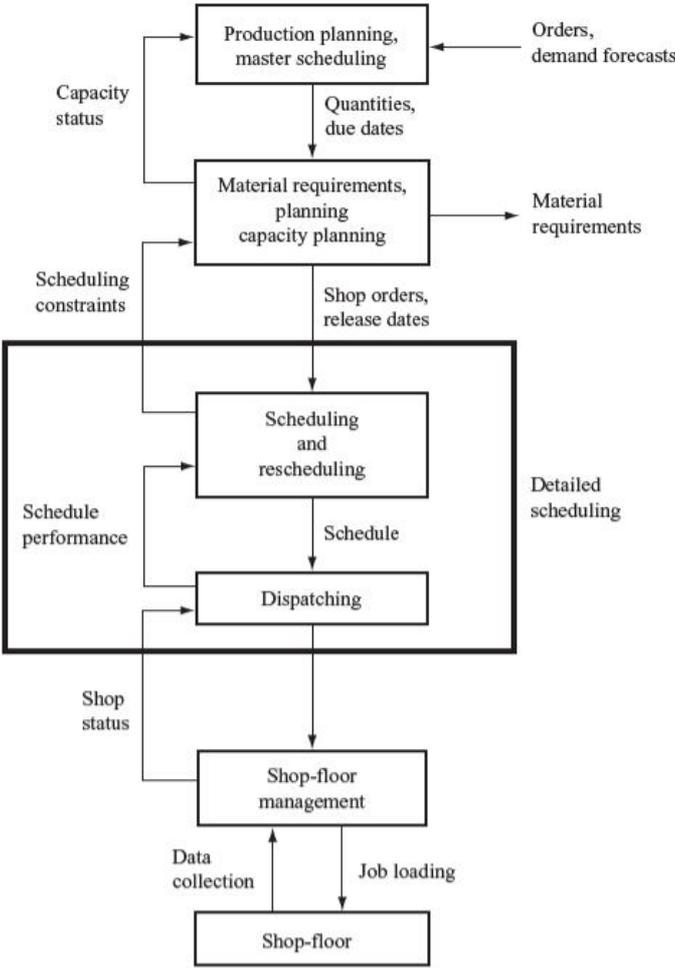


Figure 1. Information Flow Diagram in a Manufacturing System. Taken from: Pinedo, Michael L. Planning and Scheduling in Manufacturing and Services. Second Edition. Springer. NY, USA, 2009. P. 10 [2].

The importance of developing a model is that it gives the possibility to in some way predict and analyze different aspects of a manufacturing system under certain conditions and period of time. Discrete event simulation (DES) for instance will be useful when analyzing a defined system in response to the occurrence of certain events. [3] However, it has been said that the use of simulation models is addressed to specific projects or analysis for example related with changes in technology or processes. Most of the problems related with the difficulties in building and reusing simulation models have been related with information management. The management of information in manufacturing systems is an important issue, and so it is when it comes to a simulation study, but also to the possibilities to integrate simulation with those manufacturing applications used for instance in production planning in order to reduce time in collecting the data, have more accuracy in the data, etc. [4]

In the next section these issues will be better explained.

1.2 PROBLEM DESCRIPTION

Simulation related research, among other issues, focus on reduction of cost, the time of building a model, the reusability of the models and the possibility of integrate or exchange information with other manufacturing applications [5,6].

Simulation has a standard based analysis methodology. In the **Figure 2** the steps in a simulation study according to (Banks, 1998) are presented.

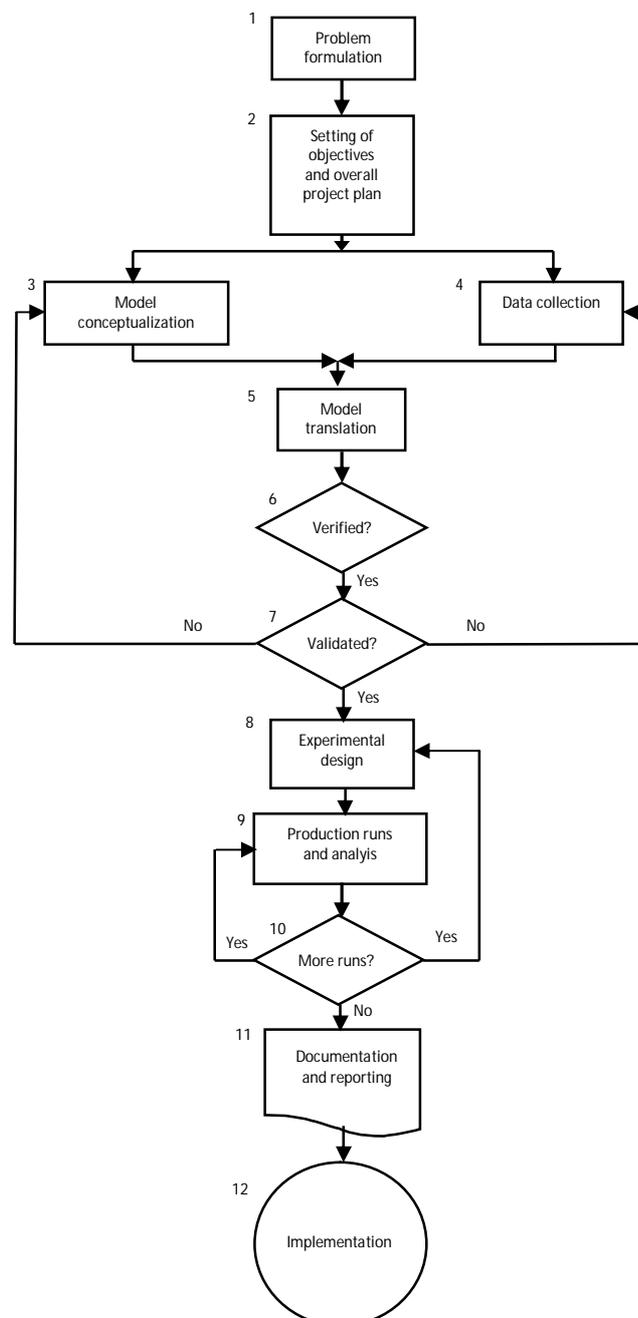


Figure 2. Steps in a Simulation Study. Taken from: Banks, Jerry. Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons. (1998) [7]

Building a simulation model for a manufacturing system taking into account the correspondence with production/process planning and the coherence with the system configuration is time consuming. The experience at the department of Production Engineering in KTH has proved that building a simulation model could take more than a half of the time in a simulation study. With verification and validation process included, the complete model construction takes the majority of time and cost. This problem has been related with the difficulties in handling the simulation data, gathering the data, more specifically with the data collection and the simulation data structuring as well as interoperability and reusability of models.

1.3 RELATED WORK AND LITERATURE REVIEW

Attempts have been made by different sectors, academy, industry, institutes, etc., to solve some of these problems of simulation. Most of the commercial tools have their own proprietary integrated solution where a data hub is built to integrate different application tools. There are different simulation software like, Delmia Quest, Tecnomatrix, ExtendSim, etc. However, the difficulties come when there is the necessity to use a model or data from one programme into another, not only between the simulation programs but also with programs used in production. In that sense the possibility to use a standard model of information developed in a neutral format could counteract these difficulties.

One of the major efforts has been done by The National Institute of Standards and Technology - NIST addressed to standardize the use of the Core Manufacturing Simulation Data Information Model - CMSDIM. A Machine Shop Information model was developed by a group of NIST (McLean et al., 2005) to represent and exchange machine shop data [8].

The machine shop information model has been transferred to the Core Manufacturing Simulation Data - CMSD Product Development Group - PDG of the Simulation Interoperability Standards Organization - SISO, to generate a CMSD information model – CMSDIM. This is defined using the Unified Modeling Language – UML and XML. [9]

The CMSDIM has been developed to facilitate the exchange of information, the use of manufacturing simulation and the integration of DES with other manufacturing applications. It describes neutral definitions of manufacturing entities and its relationship to create manufacturing simulation [4].

The CMSDIM does not provide a methodology on how it should be implemented in different software for example, but there are some studies about how the CMSDIM could be implemented using different software like: Arena, Legin, Enterprise Dynamics, etc. Some examples of these studies have been done in a company of the automotive industry, related with modeling approaches and data structure (Johansson et al. 2007) [5], other study has been done in the aerospace industry (Lu et al. 2008) [10]. There are also examples related with interoperability (Mazhari and Son 2009) [11], development of methodologies related with input data management (Bengtsson et al. 2009) [12], and data architecture (Boulonne et al. 2010) [13].

1.4 CORE MANUFACTURING SIMULATION DATA INFORMATION MODEL (CMSDIM)

The CMSDIM is defined in two types: 1) Unified Model Language (UML) and 2) eXtensible Markup Language - XML schema. The UML representation is a graphical representation while the XML is a textual representation.

The UML based CMSDIM

The first format of the CMSDIM representation is based on UML Diagrams, as a definition the **Unified Model Language (UML)** "Is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct, and document the artifacts of a software-intensive system. The UML is appropriate for modeling systems ranging from enterprise information systems to distributed Web-based applications and even to hard real time embedded systems." [14]

The UML approach in the CMSDIM consists of a collection of packages with information of different manufacturing areas modeled as UML diagrams. A UML package diagram is defined that shows the dependency relationships between the packages and the key manufacturing entities defined within. The manufacturing entities within each package are modeled as UML classes and the characteristics associated with each entity are modeled as UML class attributes. [4]

A class diagram "shows a set of classes, interfaces, and collaboration and their relationship. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system". [14 b)]

The XML based CMSDIM

The second format of the CMSDIM representation is based on XML schemas, as a definition the **Extensible Markup Language (XML)** "is a W3C-endorsed standard for document markup. It defines a generic syntax used to mark up data with simple, human-readable tags. It provides a standard format for computer documents that is flexible enough to be customized for domains as diverse as web sites, electronic data interchange, vector graphics,..." [15 a)] However, the definition of the **XML Schema** "is an XML document containing a formal description of what comprises a valid XML document. A W3C XML schema Language schema is an XML Schema written in the particular syntax recommended by the W3C." "An XML document described by a schema is called an *instance document*. If a document satisfies all the constraints specified by the schema, it is considered to be *schema-valid*." [15 b)]

The XML schema representation will facilitate the development of the XML instance document, which then should be validated with validation tools. The XML schema should be consistent with the UML diagrams.

The CMSIM is composed of packages which are presented in the following **Figure 3**.

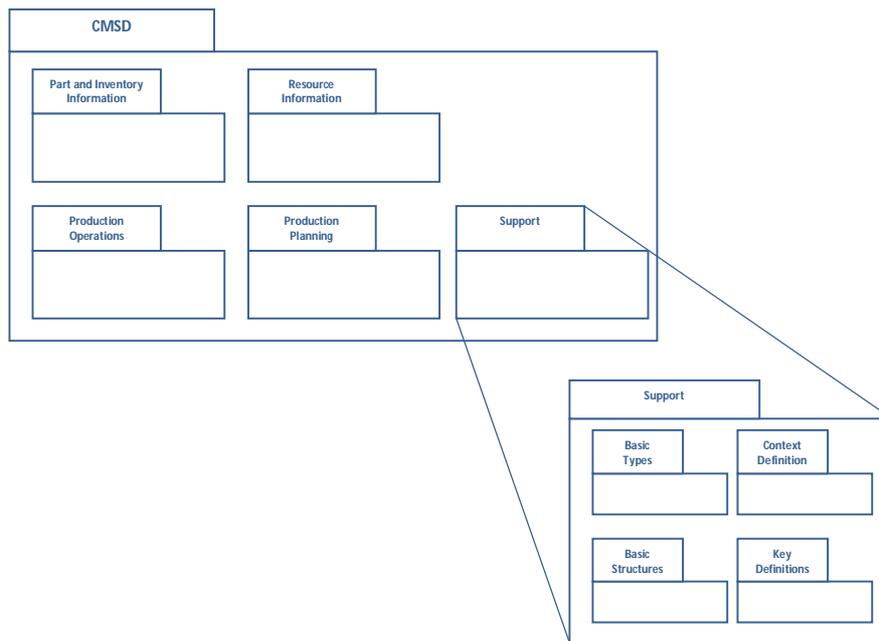


Figure.3 CMSDIM Packages

A list of the major Manufacturing Information Categories in the CMSDIM is presented below:

- Organization
- Calendar
- Resource
- Skill definition
- Setup definition
- Operation definition
- Part
- Bill-of-materials
- Process plan
- Maintenance plan
- Work
- Schedule
- Revision
- Reference
- Probability distribution

1.5 OBJECTIVE

Given some of the background and delimitations, the general objective is stated.

The objective of this thesis work is to contribute to the use of the CMSDIM in ExtendSim V8.

Specific objectives

- *Develop of a method to import databases structured based on the CMSDIM into the Simulation software ExtendSim V8.*
- *Apply the method in a case study.*

There were not found implementations tests of the CMSDIM using the ExtendSim V8. The argumentation for using ExtendSim V8 is that this software is used at the Production Engineering Department at KTH and it is also known to be used in some companies in Sweden.

2. OUTLINE

The outline of the thesis could be summarized in the following figure.

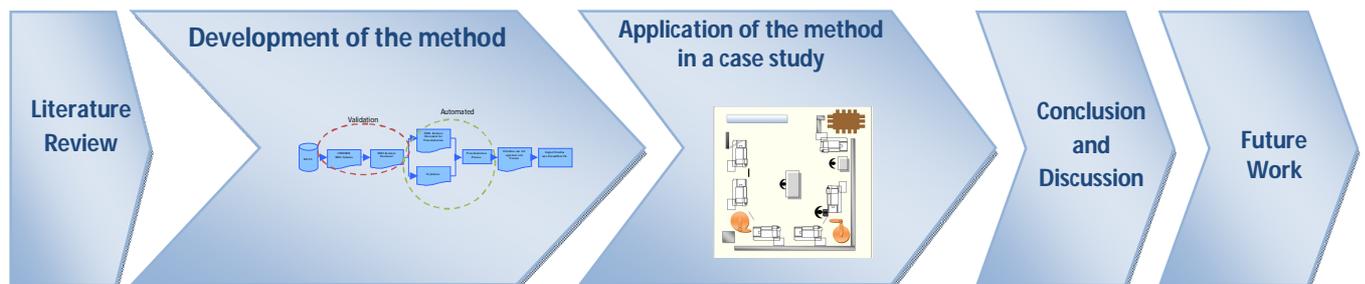


Figure.4 Outline of the thesis

The thesis will be organized as follows: The first section will be addressed to the introduction, problem description, literature review and objective of this work, which have been already presented. In the second part, the development of the method will be described and documentation will be presented. The next section will present the application of the method in a case study. Finally, conclusion, discussion and future work will give some elements to reflect about this topic and thesis work.

3. DEVELOPMENT OF THE METHOD

In this section the description of the method will be presented. It is important to describe three important issues related with the software ExtendSimV8 and important aspects to justify why the method was developed in that way:

- 1) It was not possible to import XML files directly into the software. Information was reviewed (For example from the Discussion Forum of the software page, which, while answering a question, informed that XML as text file format could be read in the software directly using ModL to parse it. It was also mentioned a situation in which that it would be an appropriate approach)¹. This possibility was superficially proved without success and not further information about how to do this in ExtendSim V8

¹ This information was taken from: <http://www.extendsim.com/forums/viewtopic.php?f=2&t=1312> Opened 2012-01-30.

[.xml communication](#) by [palmer.ap](#) » Mon Oct 18, 2010 1:16 pm

Is there a way to read information directly out of an xml file? I know it is possible to translate xml into other formats, but I'd like to read data out of the xml file directly. Thanks.

[palmer.ap](#) Posts: 6 Joined: Tue Jul 21, 2009 7:56 am

[Re: .xml communication](#) by [dkrahl](#) » Mon Oct 18, 2010 5:31 pm Since XML is a text file format, you can read it directly into ExtendSim and use ModL to parse it. If the XML file has relatively few tokens, this could be a good approach. However, if you want a more general purpose solution, you may want to consider either converting the file to another format or communicating to a XML parser. davek [dkrahl](#) Posts: 744 Joined: Thu Apr 27, 2000 12:00 am

was found. This possibility was not deeply explored due to the available time to develop this thesis.

- 2) ExtendSim V8 is well suitable working with Microsoft Excel and transformed documents from it. It was therefore necessary to consider the requirement of the software to import the data. A one tab separator text was found to be an appropriate format to the development of the method, based on previous experience in the department of Production Engineering at KTH and supported by information in the ExtendSim V8 user guide. In that sense it was necessary to include a heading of data description into the documents to be able to import the database into ExtendSim V8. This heading will be described in the following sections.
- 3) Probability distributions are really important in simulation because some information can be expressed in that way. In order to use probability distributions, it was necessary to create the text data in a format the software would read. This will be explained later on.

The method is composed of some stages. A description is provided below; however a flowchart with a better understanding of the process itself is presented later on.

As it is possible to observe in the **Figure 5**, we begin with the necessity of modeling some data. This data will be firstly understood and then an XML schema based in the CMSDIM will be created. The next step consists on the creation of the XML Instance Document based on the initial XML schema.

The instance document will be then validated against the XML schema. Then it will be necessary to identify the information is needed for the simulation model, and prepare the xml instance document. This xml instance document will be related with the stylesheet. The stylesheet will be elaborated with the aim of identify only the information needed, and prepared to obtain an output that does not require adding any extra data to import it into ExtendSim V8.

A process of transformation will follow, and then prepare the document to finally import it into a model. One final and important step will be to check the data in the simulation model, for instance in the case of the probability distributions to see if they are working correctly.

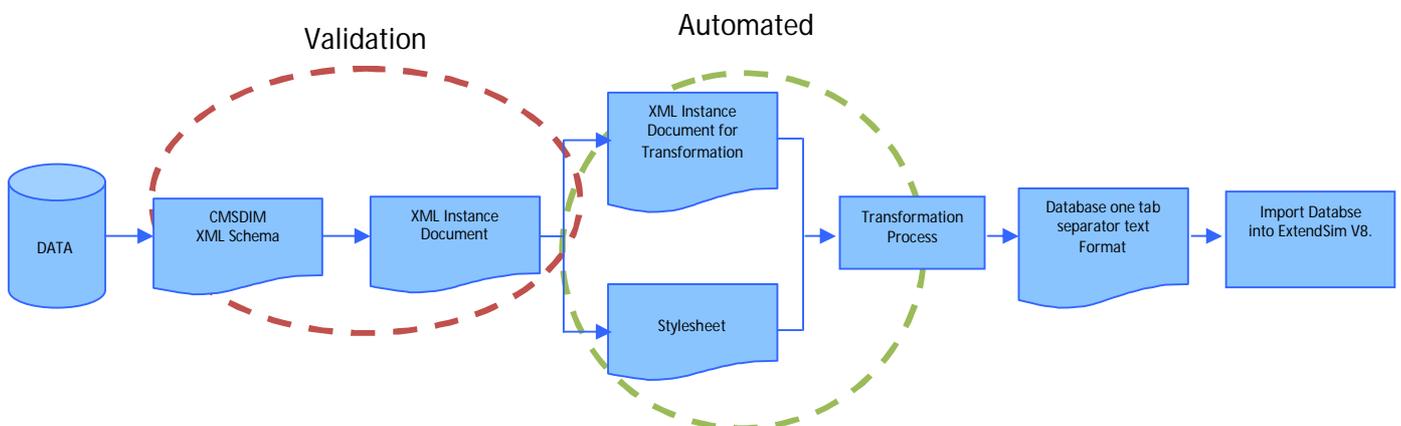


Figure 5. Description of the Method

As it was mentioned the CMSDIM is the base for structuring the information. The method will be developed based on the production planning package, a part of CMSDIM as seen in **Figure 6** above.

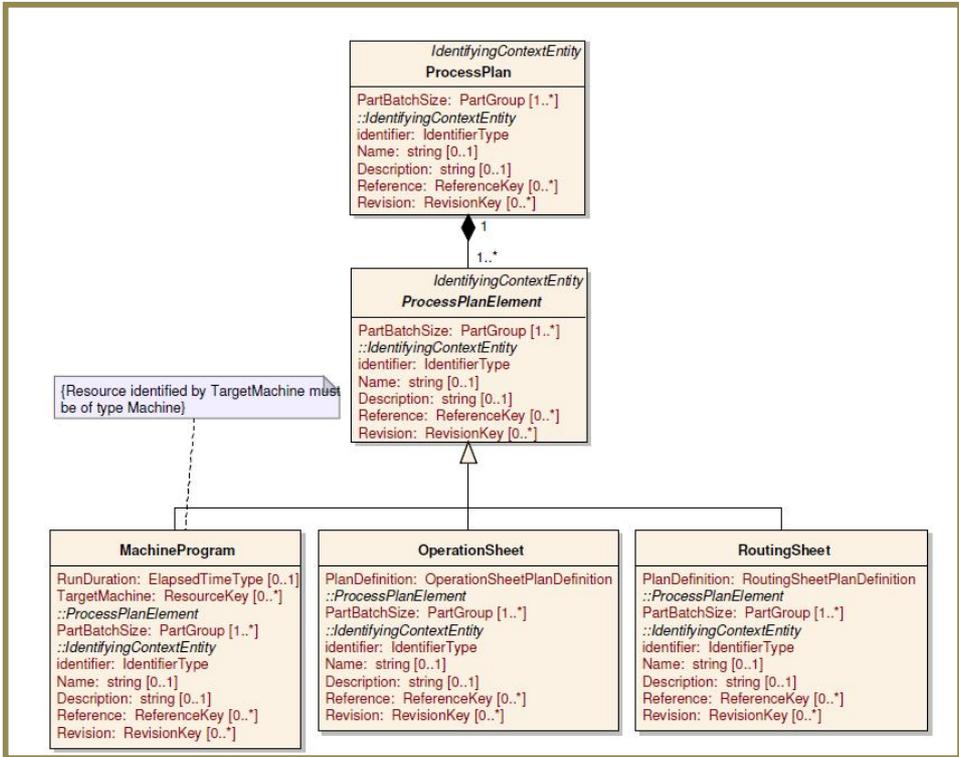
The Process Planning package is composed of the following classes:

- | PRODUCTION PLANNING PACKAGE |
|--|
| <ul style="list-style-type: none"> • Calendar, Shift, and Related Classes • Organization and Related Classes • <u>Process Plan and Related Classes</u> • Operation Definition Class • Maintenance Plan Class |

Figure.6 Production Planning Package

According to the CMSDIM, the **Production Planning Package** “contains definitions for UML classes that represent information describing the organizations involved in manufacturing process, the plan for dates and hours of operation for production resources, and the detailed procedures necessary to manufacture products.” [4]

The focus will be put in the process plan class and specifically in the routing sheet. According to the CMSDIM the Process Plan Class provides a means to define a structured collection of information that describes a detailed strategy for creating a part.



The Routing Sheet Plan Definition will allow the description of a sequence of stations and the processing steps. The Plan definition attribute is defined by the Generic Plan Definition Class. The sequences of operations in the Plan Definition are defined as Plan Steps.

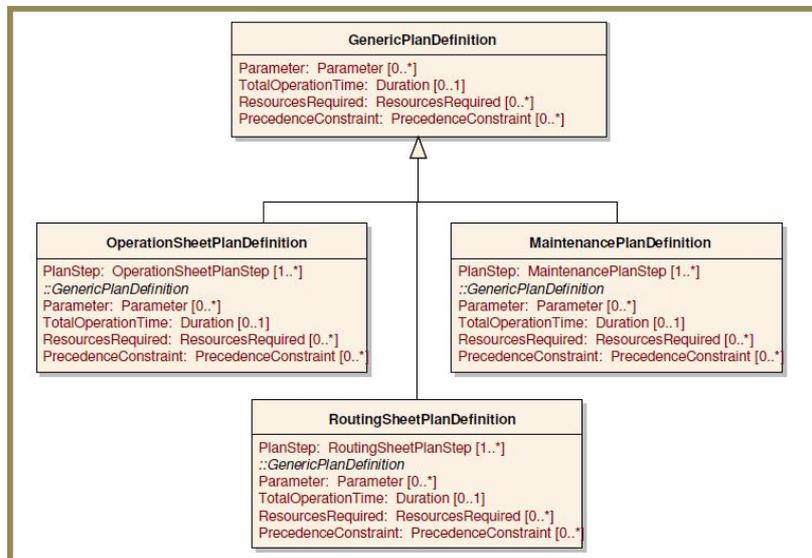


Figure.8 Generic Plan Definition in CMSDIM

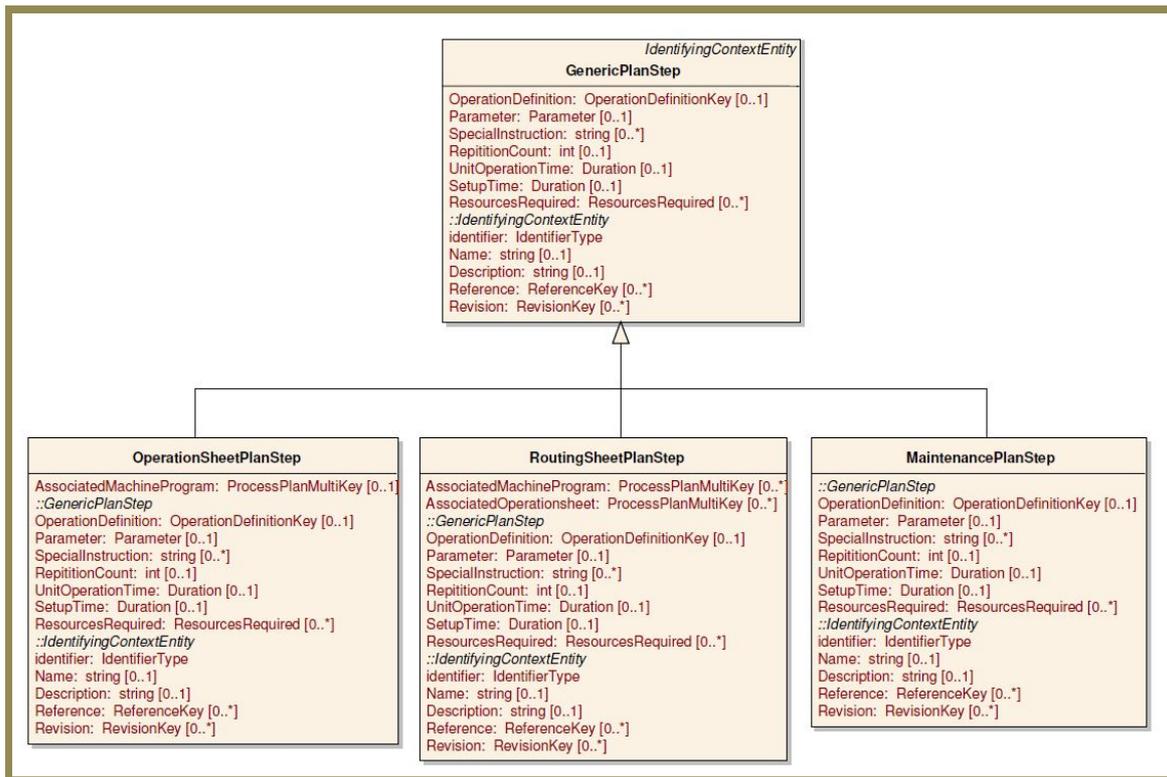


Figure. 9 Generic Plan Step in CMSDIM

The method will be based in the Generic Plan Step, to come to a general alternative.

The development of the method will now be explained. To develop the method a simple simulation model was created. **Figure 10.** This model will be used to test the data importation.

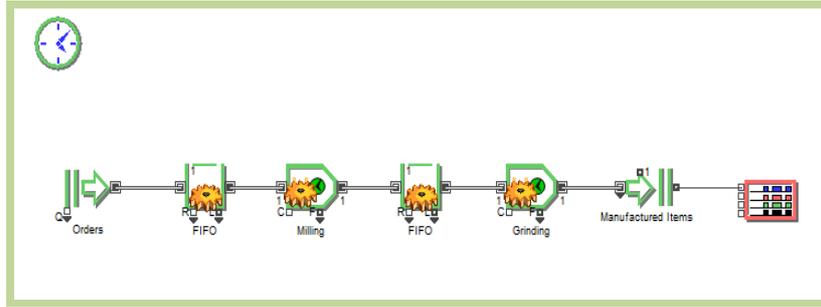


Figure.10 Simulation Model

A flowchart with the description of the stages in the method has been created. **Figure 11.**

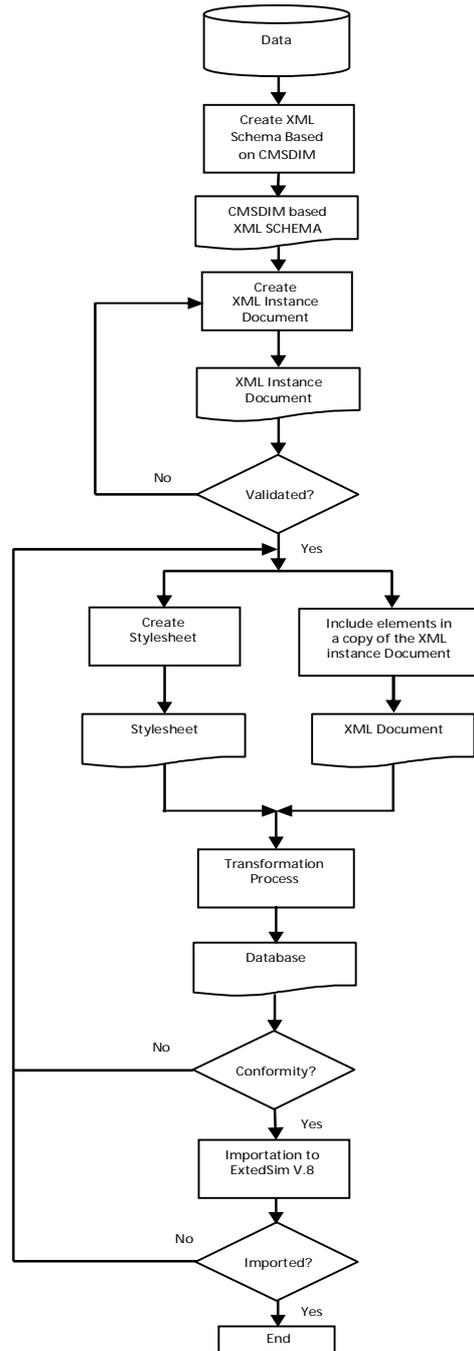


Figure 11. Flowchart of the Method

At this initial state, to develop the method, the data to be used will be based on the created model. That is, the data at this stage is not from an actual production line. Following, the stages and the validation as well as the automation processes will be explained.

The process of creation and validation of the XML documents was done following the steps in the **Figure 12** below.

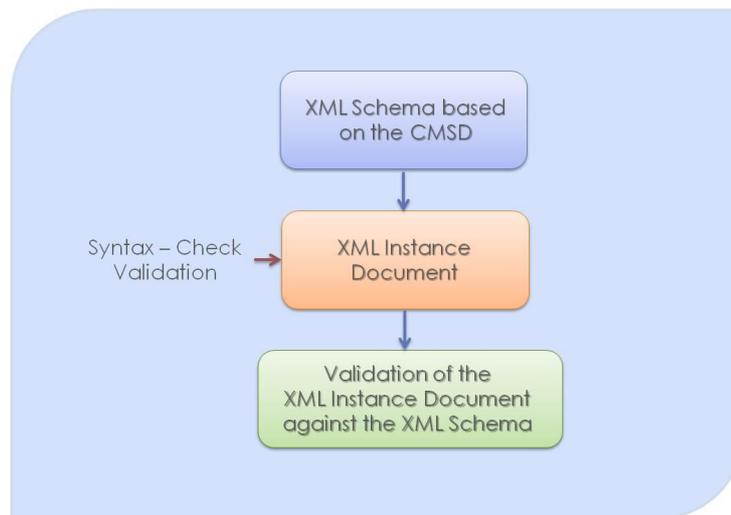


Figure 12. Validation Process

A schema based on the Process Plan and routing sheet was created based on the production line presented above.

The schema was created manually from the CMSDIM UML representations. Then a XML instance document was also created manually, and it was validated against the schema. A syntax-check was also done in http://www.w3schools.com/xml/xml_validator.asp (last accessed, February 22, 2012) .A schema validation was also done: <http://www.w3.org/2001/03/webdata/xsv#hlp-check> (last accessed, February 22, 2012) .

Both, the schema (Appendix 1) and the instance document were created using:

- Notepad++, and
- XML Notepad

The validation process in this stage was done in XML Notepad and also with an online validator <http://tools.decisionsoft.com/schemaValidate/> (last accessed, February 22, 2012) The XML instance document is presented below.

The XML instance document

```
<?xml version="1.0" encoding="UTF-8"?>
<CMSD xsi:schemaLocation="http://www.csc.kth.se/~harari/Test_total Test_total.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.csc.kth.se/~harari/CMSD_total">
  - <Heading>
    <Line1>File_Name</Line1>
    <Line2>Number of Tabs</Line2>
    <Line3>Tab Names</Line3>
    <Line4>Table Name</Line4>
    <Line5>Table Index</Line5>
    <Line6>Rows</Line6>
    <Line7>Cols</Line7>
    <Line8>Table Format</Line8>
    <Line9>Table Note</Line9>
    <Line10>Number of Tabs</Line10>
    <Line11>Tab Indexes</Line11>
    <Line12>Field Locations</Line12>
    <Line13>Field Notes</Line13>
    <Line14>Field Format</Line14>
    <Line15>Field Decimal</Line15>
    <Line16>Field Comma</Line16>
    <Line17>Field Unique</Line17>
    <Line18>Field Access</Line18>
    <Line19>Record ID</Line19>
    <Line20>Field Initializer</Line20>
    <Line21>Do Initialization</Line21>
    <Line22>Init First Run Only</Line22>
  </Heading>
  - <ProcessPlan ProcessPlanIdentifier="1">
    - <RoutingSheet>
      <RoutingSheetIdentifier>Routing_1</RoutingSheetIdentifier>
      - <PartBatchSize>
        <PartClass partIdentifier="PartA"/>
        <PartQuantity>500</PartQuantity>
      </PartBatchSize>
      - <PlanDefinition>
        - <PlanStep identifier="1a">
          <OperationDefinition>Milling</OperationDefinition>
          - <SetupTime>
            <unit>minute</unit>
            - <Distribution>
              <Name>Normal</Name>
              <Description>Normal</Description>
              <Value1>10</Value1>
              <Value2>15</Value2>
            </Distribution>
          </SetupTime>
          - <UnitOperationTime>
            <unit>minute</unit>
            <Value>5</Value>
          </UnitOperationTime>
        </PlanStep>
        - <PlanStep identifier="1a">
          <OperationDefinition>Grinding</OperationDefinition>
          - <SetupTime>
            <unit>minute</unit>
            - <Distribution>
              <Name>RealUniform</Name>
              <Description>Inside of a certain interval</Description>
              <Value1>15</Value1>
              <Value2>20</Value2>
            </Distribution>
          </SetupTime>
          - <UnitOperationTime>
            <unit>minute</unit>
            <Value>8</Value>
          </UnitOperationTime>
        </PlanStep>
      </PlanDefinition>
    </RoutingSheet>
  </ProcessPlan>
</CMSD>
```

As it is possible to see, the document contains what will be denominated in this thesis as “heading” information. A separated heading schema (Appendix 2) was created in order to maintain the CMSDIM schema alone for other possible applications. This was done also to facilitate the automation process which will be explained later on.

Heading

The heading is composed of 22 lines. This is necessary to include in the document to import it into the software. Also, this is important because some of the problems detected during the importation were related with extra spaces after the rows, or invalid definitions. In the **Table 1** it is possible to find a guide on how the information should be taken into account in the heading.

No.	HEADING	
1	File_Name:	Not mandatory
2	Number of Tabs	Not mandatory
3	Tab Names	Not mandatory
4	Table Name	Not mandatory, however important to identify the table
5	Table Index	Could be zero or empty
6	Rows	Required and mandatory according to the data
7	Cols	
8	Table Format	Could be zero or empty
9	Table Note	Could be empty or with a number or text, no influence
10	Number of Tabs	Must be always more than 1. Otherwise, there will be problems with Field Format and Field Decimal
11	Tab Indexes	Could be empty, zero or more
12	Field Locations	Could be empty
13	Field Notes	Could be empty
14	Field Format	Must be filled according to the Field Format specification.
15	Field Decimal	Should be zero, unless another data is needed
16	Field Comma	Should be zero, unless another data is needed
17	Field Unique	Should be zero, unless another data is needed
18	Field Access	Should be zero, unless another data is needed
19	Record ID	Should be zero, unless another data is needed
20	Field Initializer	Should be empty, unless another data is needed
21	Do Initialization	Should be zero, unless another data is needed
22	Init First Run Only	Should be zero, unless another data is needed

Table 1. Heading Guide

Special attention is given to the Rows, Cols (Columns) and Field Format. These three will be part of the automation process that will be explained later on, but the field format is now described more in detail. More attention is given to the field format because every data should be identified with a determined number to be imported into the model. The **Table 2** shows this description.

FIELD FORMAT		
Field Properties in ExtendSim V.8		
FIELD TYPE		
No.	Name	Identification
1	NUMBER, GENERIC FORMAT	2000
2	NUMBER, SCIENTIFIC NOTATION	2001
3	NUMBER, PERCENT	2002
4	NUMBER, CURRENCY	2003
5	NUMBER, INTEGER	1000
6	STRING	4000
7	DATE TIME	2004
8	DATA ADDRESS (DB or GA)	2005
9	BOOLEAN CHECKBOX	1001
10	LIST OF TABLES	4001

Table 2. Field Format Description in ExtendSim V8

The identification number will be included in the heading.

Transformation and Automation Process

For the transformation process Extensible Stylesheet Language Transformations (XSLT) was used. "XSLT is a language which can transform XML documents into any text-based format, XML or otherwise". [16]

This will allow the transformation of the XML document in to one tab separator text format.

Following the XSLT created document will be presented. As it is possible to see in the blue circles, the stylesheet contains some functions like count and `<xsl:if>` that are used for the automation of the transformation process. The green circle is related with the distribution, which will be explained in the following section.

This Stylesheet is related with a copy of the original XML instance document that was developed and validated before for these reasons:

- The first one was related to the schema for validation, and
- In order to automate the transformation some "flags" will be added in the XML document to automate the counting of the columns.

The function `<xsl:if>` is used to automate the filling process of the File Format. In addition, to have an initial column with numbers in the table the following Xpath expression was included:

`<xsl:value-of select="position()"/>` .

Xpath is important in the use of Stylesheets. "Xpath is a non-XML language for identifying particular parts of XML documents". [15 c]

The stylesheet and the XML document are presented below.

Stylesheet

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="text"/>
  - <xsl:template match="/">
    - <xsl:for-each select="CMSD/Heading">
      - <xsl:value-of select="Line1"/>
      Data for Simulation
      <xsl:text> </xsl:text>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line4"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="//ProcessPlan[@ProcessPlanIdentifier='1']//RoutingSheetIdentifier"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line5"/>
      <xsl:text> 1 </xsl:text>
      <xsl:value-of select="Line6"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="count(//ProcessPlan[@ProcessPlanIdentifier='1']//PlanStep[@identifier='1a'])"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line7"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="count(//ProcessPlan[@ProcessPlanIdentifier='1']//PlanStep[@identifier='1a']//flag)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line8"/>
      <xsl:text> 0 </xsl:text>
      <xsl:value-of select="Line9"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line10"/>
      <xsl:text> 2 </xsl:text>
      <xsl:value-of select="Line11"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line12"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line13"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line14"/>
      <xsl:text> </xsl:text>
      - <xsl:if test="//OperationDefinition">
        <xsl:text>4000 </xsl:text>
      </xsl:if>
      - <xsl:if test="//UnitOperationTime/Value">
        <xsl:text>2000 </xsl:text>
      </xsl:if>
      - <xsl:if test="//SetupTime/unit">
        <xsl:text>2000 </xsl:text>
      </xsl:if>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line15"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line16"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line17"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line18"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line19"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line20"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="Line21"/>
      <xsl:text> 0 0 0 </xsl:text>
      <xsl:value-of select="Line22"/>
      <xsl:text> 0 0 0 </xsl:text>
    </xsl:for-each>
    Field Names Operation Process_time SetupDistribution
    - <xsl:for-each select="//ProcessPlan[@ProcessPlanIdentifier='1']//PlanStep[@identifier='1a']">
      <xsl:value-of select="position()"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="OperationDefinition"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="UnitOperationTime/Value"/>
      <xsl:text> </xsl:text>
      <xsl:text> [ </xsl:text>
      <xsl:value-of select="SetupTime//Name"/>
      <xsl:text> ; </xsl:text>
      <xsl:value-of select="SetupTime/Distribution//Value1"/>
      <xsl:text> ; </xsl:text>
      <xsl:value-of select="SetupTime/Distribution//Value2"/>
      <xsl:text> ; </xsl:text>
      <xsl:text> ] </xsl:text>
    </xsl:for-each>
    <xsl:text> EndOfFile </xsl:text>
  </xsl:template>
</xsl:stylesheet>

```

XML Document for Transformation

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="Test_transformation.xsl"?>

<CMSD>
  <Heading>
    <Line1>File Name:</Line1>
    <Line2>Number of Tabs</Line2>
    <Line3>Tab Names</Line3>
    <Line4>Table Name</Line4>
    <Line5>Table Index</Line5>
    <Line6>Rows</Line6>
    <Line7>Cols</Line7>
    <Line8>Table Format</Line8>
    <Line9>Table Note</Line9>
    <Line10>Number of Tabs</Line10>
    <Line11>Tab Indexes</Line11>
    <Line12>Field Locations</Line12>
    <Line13>Field Notes</Line13>
    <Line14>Field Format</Line14>
    <Line15>Field Decimal</Line15>
    <Line16>Field Comma</Line16>
    <Line17>Field Unique</Line17>
    <Line18>Field Access</Line18>
    <Line19>Record ID</Line19>
    <Line20>Field Initializer</Line20>
    <Line21>Do Initialization</Line21>
    <Line22>Init First Run Only</Line22>
  </Heading>
  <ProcessPlan ProcessPlanIdentifier="1">
    <RoutingSheet>
      <RoutingSheetIdentifier>Routing_1</RoutingSheetIdentifier>
      <PartBatchSize>
        <PartClass partIdentifier="PartA" />
        <PartQuantity>500</PartQuantity>
      </PartBatchSize>
      <PlanDefinition>
        <PlanStep identifier="1a">
          <OperationDefinition>Milling<flag/></OperationDefinition>
          <SetupTime>
            <unit>minute</unit>
            <Distribution>
              <Name>Normal</Name>
              <Description>Normal<flag/></Description>
              <Value1>10</Value1>
              <Value2>15</Value2>
            </Distribution>
          </SetupTime>
          <UnitOperationTime>
            <unit>minute</unit>
            <Value>5</Value>
          </UnitOperationTime>
        </PlanStep>
        <PlanStep identifier="1a">
          <OperationDefinition>Grinding<flag/></OperationDefinition>
          <SetupTime>
            <unit>minute</unit>
            <Distribution>
              <Name>RealUniform</Name>
              <Description>Inside of a certain interval</Description>
              <Value1>15</Value1>
              <Value2>20</Value2>
            </Distribution>
          </SetupTime>
          <UnitOperationTime>
            <unit>minute</unit>
            <Value>8</Value>
          </UnitOperationTime>
        </PlanStep>
      </PlanDefinition>
    </RoutingSheet>
  </ProcessPlan>
</CMSD>

```

Probability Distributions

In the XML document, some distributions were included with the purpose to see if it would be readable by the software.

As was told earlier, the data included in the description of the technique, does not correspond to a real production line, but to the model mentioned at the beginning.

The CMSD information model presents the following distributions.

Distributions			
Name	Parameter 1	Parameter 2	Parameter 3
constant	value		
uniform	minimum	maximum	
normal	mean	standardDeviation	
exponential	mean		
lognormal	mean	standardDeviation	
triangular	minimum	mode	maximum
weibul	alpha	beta	
gamma	alpha	beta	
beta	alpha	beta	
poisson	Mean		
erlangK	mean	K	
chiSquare	K		

Table 3. Distributions in CMSDIM

In the Stylesheet a concatenation process was used to transform the distribution in text format that ExtendSimV8 could read.

```

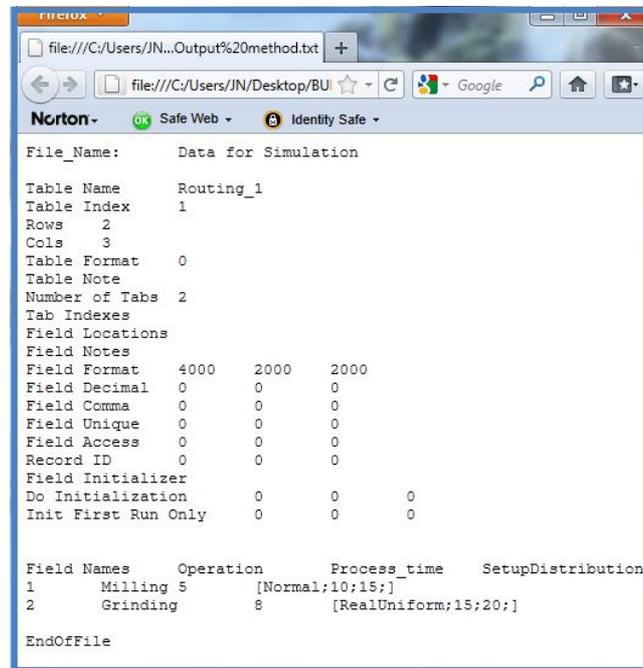
<xsl:text></xsl:text>
<xsl:text>[</xsl:text>
<xsl:value-of select="SetupTime//Name"/>
<xsl:text>;</xsl:text>
<xsl:value-of select="SetupTime/Distribution//Value1"/>
<xsl:text>;</xsl:text>
<xsl:value-of select="SetupTime/Distribution//Value2"/>
<xsl:text>;</xsl:text>
<xsl:text>]</xsl:text>
</xsl:for-each>

```

Output

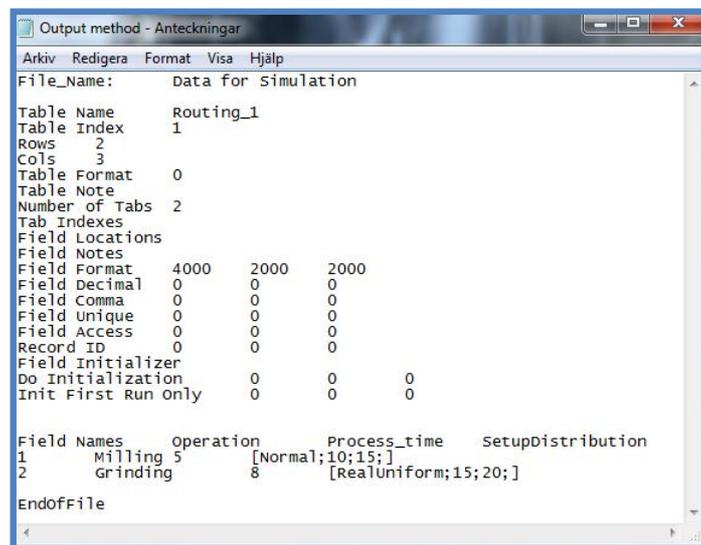
To see the output of the transformation process the related XML document should be dragged into a browser. Firefox has been proved to be more appropriate for this.

The XML file related to the Stylesheet opened in Firefox is presented below:



```
file:///C:/Users/JN...Output%20method.txt
file:///C:/Users/JN/Desktop/BU...
Norton- Safe Web Identity Safe
File_Name: Data for Simulation
Table Name Routing_1
Table Index 1
Rows 2
Cols 3
Table Format 0
Table Note
Number of Tabs 2
Tab Indexes
Field Locations
Field Notes
Field Format 4000 2000 2000
Field Decimal 0 0 0
Field Comma 0 0 0
Field Unique 0 0 0
Field Access 0 0 0
Record ID 0 0 0
Field Initializer
Do Initialization 0 0 0
Init First Run Only 0 0 0
Field Names Operation Process_time SetupDistribution
1 Milling 5 [Normal;10;15;]
2 Grinding 8 [RealUniform;15;20;]
EndOfFile
```

This data can be easily copied into Notepad. The data in notepad is presented below:



```
Output method - Anteckningar
Arkiv Redigera Format Visa Hjälp
File_Name: Data for Simulation
Table Name Routing_1
Table Index 1
Rows 2
Cols 3
Table Format 0
Table Note
Number of Tabs 2
Tab Indexes
Field Locations
Field Notes
Field Format 4000 2000 2000
Field Decimal 0 0 0
Field Comma 0 0 0
Field Unique 0 0 0
Field Access 0 0 0
Record ID 0 0 0
Field Initializer
Do Initialization 0 0 0
Init First Run Only 0 0 0
Field Names Operation Process_time SetupDistribution
1 Milling 5 [Normal;10;15;]
2 Grinding 8 [RealUniform;15;20;]
EndOfFile
```

Special attention should be paid to the spaces. Usually extra spaces could create problems in the importation process.

As it is possible to see, the automation and transformation processes will give as a result a finished table with the data we want and the necessary information in the heading. This information below is ready to be imported into the model.

Table to be imported in to ExtendSim V8

File_Name: Data for Simulation

Table Name Routing_1

Table Index 1

Rows 2

Cols 3

Table Format 0

Table Note

Number of Tabs 2

Tab Indexes

Field Locations

Field Notes

Field Format 4000 2000 2000

Field Decimal 0 0 0

Field Comma 0 0 0

Field Unique 0 0 0

Field Access 0 0 0

Record ID 0 0 0

Field Initializer

Do Initialization 0 0 0

Init First Run Only 0 0 0

Field Names Operation Process_time SetupDistribution

1 Milling 5 [Normal;10;15;]

2 Grinding 8 [RealUniform;15;20;]

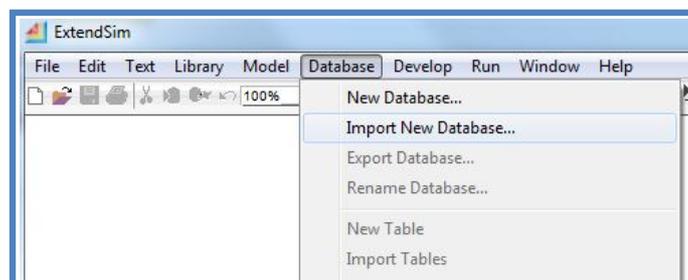
EndOfFile

It is important to notice that the "Field Names" column does not count in the cols number.

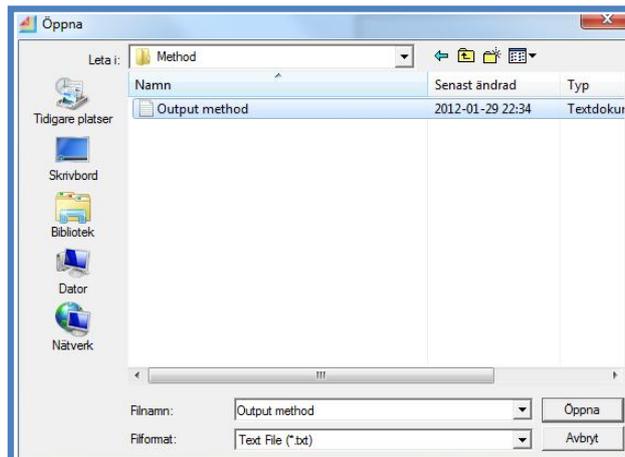
Process of Importation into ExtendSim V8

The steps in the importation process are shown below.

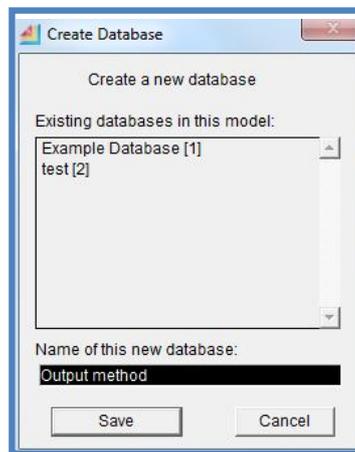
Step 1



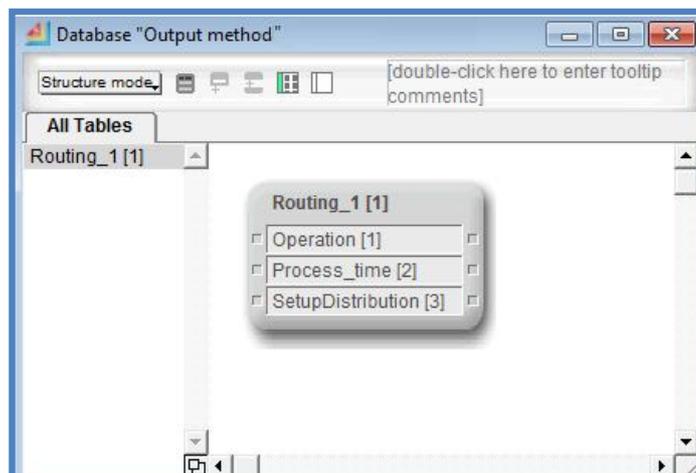
Step 2



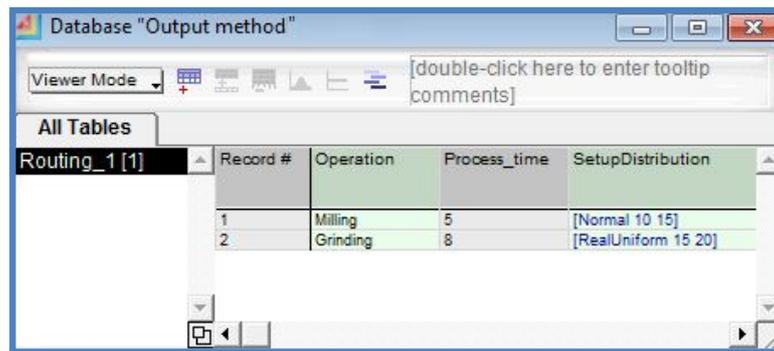
Step 3



Step 4 – One view mode



Step 5 – Another view mode



Record #	Operation	Process_time	SetupDistribution
1	Milling	5	[Normal 10 15]
2	Grinding	8	[RealUniform 15 20]

These results could be read in the appropriate blocks.

4. CASE STUDY

The method was applied with real data of a production line of SCANIA in Sweden. A simulation model for this line has been developed by a group of Researchers of KTH. The purpose of the application of the method is to structure some data needed for the model based in the CMSDIM.

For this study, visits to the plant were made to understand the process and gather the necessary information. The visits were important in order to have an understanding of the production flow in the line, how the different processes are connected, how the routes are organized, how the jobs are scheduled.

This production line produces five different products in five defined routes. The line is composed of six operations. Specific information of the process as well as the data cannot be revealed in this study. However, some information is provided below in order to understand the process.

In the study the production planning package described in the development of the method was used, and for the case it was also used the resources information package in order to be able to create information tables in terms of other information if necessary.

The Resource Information Package according to the CMSDIM “contains for UML classes that represent definitions for the characteristics of the equipment and workers used in the manufacturing process, the skills associated with workers, and setup information required for the efficient use of the equipment”. The UML representation of the Resource Class in the CMSDIM is presented in the **Figure 13**.

In order to provide information about the dimension of the Production Line where the case study was developed, a flowchart of the process is presented to give an understanding of the system configuration of the line. To provide a visual representation of the production line, a Layout is also presented.

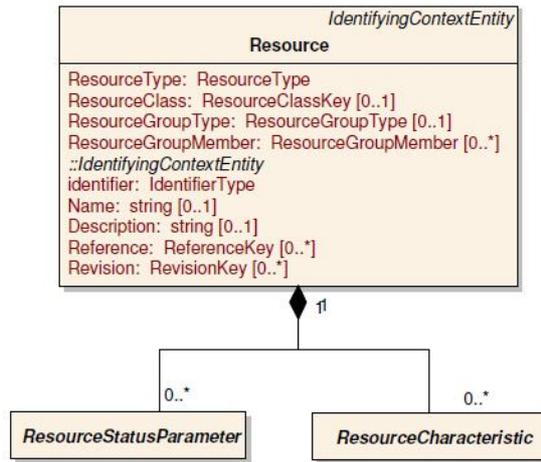
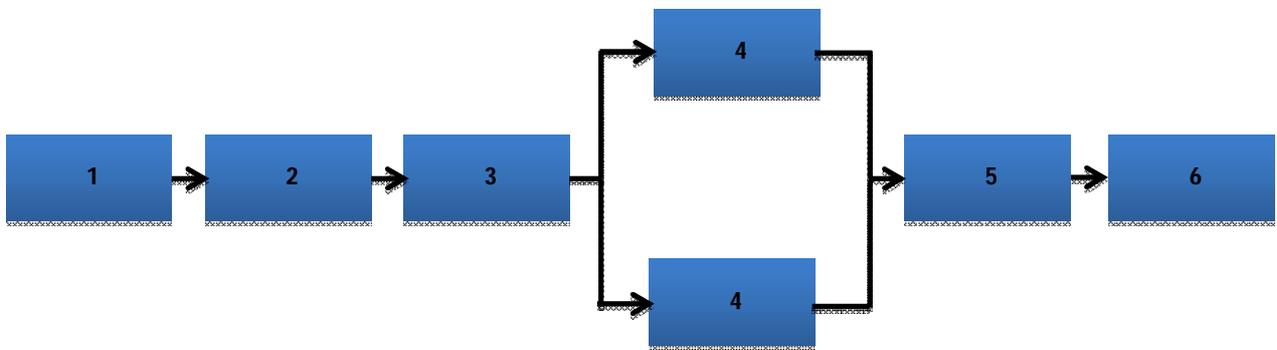


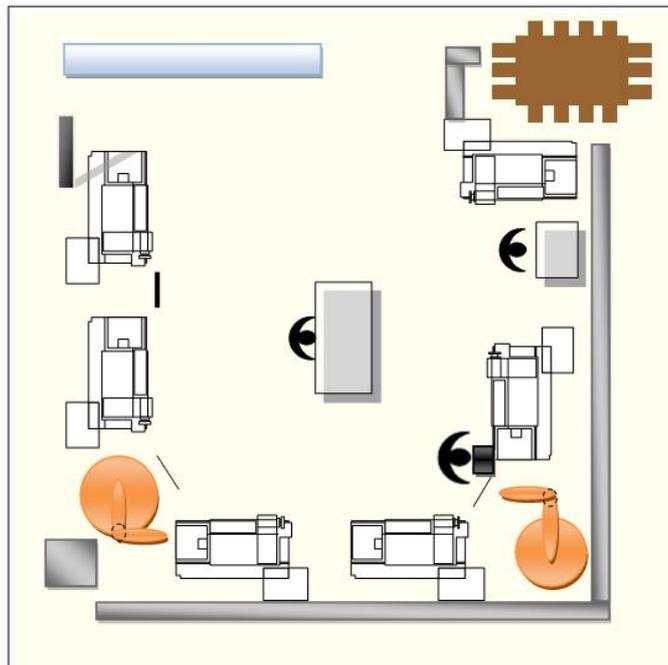
Figure 13. Resources Class in CMSDIM

FLOWCHART OF THE PROCESS – System Configuration



LAYOUT OF THE PRODUCTION LINE

As a way to visualize the production line, a layout has been created.



Time was dedicated to gather information in the production line. Some tables were developed for this purpose (Appendix 3). All the information needed and the explanations about the process were gathered with the collaboration of the worker in the production line, and support was also given by the development department and Production Planning department.

This information was later processed by the researchers of KTH who developed the model, and some information was selected.

Some of the information selected was structured and, as a result, eight tables were created in terms of:

- Process Time
- Setup Time (Including distributions)
- Measurement Operations and time
- Parts
- Routes
- Batch Sizes.

In the tables is included an Operations Directory, in the corresponding sequence.

All these tables were created in a unique database file that was imported in the ExtendSim V8. The behavior of the probability distributions was tested and worked correctly.

The schema contains also the definition of other data that could be included and is used in the model. If it is the case there will be only the need to instantiate the required data and continue with the method to obtain the tables.

5. CONCLUSIONS, DISCUSSION AND FUTURE WORK

It is possible to create databases from structured information based on CMSDIM as a neutral format and import it in text format to ExtendSim V8. With the approach used in this thesis work, structured data could help the identification of parameters and the system configuration which in turn could help the production planning function. This experience has also confirmed that the use of databases facilitates the use of the information in ExtendSim V8, this could also contribute to the re-use of simulation models through the facilitation in changing parameters in databases or relating databases. The use of databases could also contribute to an easier verification of the input data information, and a better organization of the information.

In addition, neutral formats facilitate the exchange of information, and XML is a readable language that also facilitates the understanding of the information.

The method developed is an alternative that can contribute to the use of the CMSDIM as a base to structure the information with ExtendSimV8. It has been proved to be useful in a production line with the characteristics that were presented in the case study. However, it is important to highlight the difficulties that this work presented with ExtendSim V8 when trying to import XML files. Even though it was mentioned before, this work could not focus

on how to overcome those difficulties due to the available time. If ExtendSim V8 could facilitate the use of XML maybe the process of using CMSDIM in it would be easier. Those limitations were also the ones that call for the inclusion of a Heading into the tables to import the databases in text format into ExtendSim V8. It was useful that the CMSDIM schema was kept as an individual document and related with the different schemas to validate the instance document even though they were also validated individually. This approach could allow the use of the created CMSDIM schema also for other applications.

Even though this method is applicable in a production line like the one in the case study, maybe some more work could be needed to analyze the situation in a production line with still more levels of complexity.

Future work could be done. There are still possibilities to develop more work or generate solutions for example for returning a text format database to XML, so as to also be able to report and use the data from the simulation in a structured way based on the CMSDIM. There could be also possibilities to explore deeply the way to import XML files into ExtendSim V8.

6. REFERENCES

- [1] Lv, Chao, et al. Research and optimization of reconfigurable manufacturing system configuration based on system reliability. *Kybernetes*. Vol. 39 No. 6, 2010 pp. 1058-1065 (DOI 10.1108/03684921011046834)
- [2] Pinedo, Michael L. *Planning and Scheduling in Manufacturing and Services*. Second Edition. Springer. NY, USA, 2009. P. 10
<<http://www.springerlink.com.focus.lib.kth.se/content/978-1-4419-0909-1#section=381156&page=1>> (accessed January 31, 2012)
- [3] ExtendSim Webpage. Simulation Software Overview
http://www.extendsim.com/sols_simoverview.html (accessed January 31, 2012)
- [4] CMSD Product Development Group. 2006. Core Manufacturing Simulation Data Information Model (Draft). Part 1: UML Model. NIST.
- [5] Johansson, M.; Johansson, B.; Skoogh, A.; Swee Leong; Riddick, F.; Lee, Y.T.; Guodong Shao; Klingstam, P.; , "A test implementation of the core manufacturing simulation data specification," *Simulation Conference, 2007 Winter* , vol., no., pp.1673-1681, 9-12 Dec. 2007
doi:10.1109/WSC.2007.4419789
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4419789&isnumber=4419576>
- [6] Banks, J.; Hagan, J.C.; Lendermann, P.; McLean, C.; Page, E.H.; Pegden, C.D.; Ulgen, O.; Wilson, J.R.; , "The future of the simulation industry," *Simulation Conference, 2003. Proceedings of the 2003 Winter* , vol.2, no., pp. 2033- 2043 vol.2, 7-10 Dec. 2003
doi:10.1109/WSC.2003.1261668
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1261668&isnumber=28189>
- [7] Banks, Jerry (1998). *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*. (pp: 16, 17, 18). John Wiley & Sons.
Online version available via:
http://www.knovel.com.focus.lib.kth.se/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=1452&VerticalID=0 (accessed the 6th of February 2012)
- [8] McLean et al., *Shop Data Model and Interface Specification*, NIST, 2005
- [9] Lee, et. Al. *Machine Shop Information Model Application, Next Step*. Manufacturing Systems Integration Division, Manufacturing Engineering Laboratory, National Institute of Standards and Technology. 2005

- [10] Lu, R.F.; Leong, S.; Bengtsson, N.; Johansson, B.; Riddick, F.; Lee, T.; Guodong Shao; McLean, C.; Salour, A.; Hazlehurst, L.N.; Ly, S.; , "Implementation of Core Manufacturing Simulation Data in aerospace industry," *Simulation Conference, 2008. WSC 2008. Winter* , vol., no., pp.2930, 7-10 Dec. 2008
doi: 10.1109/WSC.2008.4736420
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4736420&isnumber=4736042>
- [11] Mazhari, E. and Y.-J. Son. Inter-operability of Manufacturing Applications in a Simulated Environment via CMSD Information Model. Systems and Industrial Engineering, The University of Arizona. Available via:

<<http://www.sie.arizona.edu/faculty/son/NIST3.html>> (accessed the 22th of February 2012)
- [12] Bengtsson, N., Shao, G., Johansson, B., Lee, Y.T., Leong, S., Skoogh, A., Mclean, C., "Input Data Management methodology for Discrete Event Simulation", *Winter Simulation Conference (WSC), Proceedings of the 2009*, On page(s): 1335 - 1344, Volume: Issue: , 13-16 Dec. 2009
- [13]Boulonne, A.; Johansson, B.; Skoogh, A.; Aufenanger, M.; , "Simulation data architecture for sustainable development," *Simulation Conference (WSC), Proceedings of the 2010 Winter* , vol., no., pp.3435-3446, 5-8 Dec. 2010
doi: 10.1109/WSC.2010.5679033
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5679033&isnumber=5678856>
- [14] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. Addison-Wesley. April 2001. 14b) p.25
- [15] Harold, R E. and Means, S W. XML in a nutshell. Third Edition. O'Reilly. 2004 a)p.3 b)p.278 c)p.162
- [16] Hunter, David, and Kurt Cagle. "Chapter 4 - XSLT". Beginning XML, 2nd Edition: XML Schemas, SOAP, XSLT, DOM, and SAX 2.0. Wrox Press. © 2003. Books24x7. <<http://common.books24x7.com.focus.lib.kth.se/toc.aspx?bookid=6262>> (accessed January 31, 2012)

7. APPENDIX

1) Schema developed for the method.

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.csc.kth.se/~harari/Test_total"
xmlns="http://www.csc.kth.se/~harari/Test"
elementFormDefault="qualified">

<xsd:element name="CMSD" type="CMSD"/>
<xsd:complexType name="CMSD">
  <xsd:sequence>
    <xsd:element name="ProcessPlan" type="ProcessPlan" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ProcessPlan" type="ProcessPlan"/>
<xsd:complexType name="ProcessPlan">
  <xsd:sequence>
<xsd:element name="RoutingSheet" type="RoutingSheet" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
<xsd:attribute name="ProcessPlanidentifier" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="IdentifyingContextEntity" abstract="true">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
<xsd:attribute name="identifier" type="identifiertype"/>
</xsd:complexType>

<xsd:simpleType name="identifiertype">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:complexType name="PartGroup">
  <xsd:sequence>
    <xsd:element name="PartClass" type="PartKey"/>
    <xsd:element name="PartQuantity" type="xsd:int"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PartKey">
  <xsd:attribute name="partidentifier" type="identifiertype"/>
</xsd:complexType>

<xsd:complexType name="ProcessPlanElement" abstract="true">
  <xsd:sequence>
    <xsd:element name="RoutingSheetidentifier" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PartBatchSize" type="PartGroup" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

```

    </xsd:sequence>
</xsd:complexType>

<xsd:element name="RoutingSheet" type="RoutingSheet"/>
<xsd:complexType name="RoutingSheet">
  <xsd:complexContent>
    <xsd:extension base="ProcessPlanElement">
      <xsd:sequence>
        <xsd:element name="PlanDefinition" type="RoutingSheetPlanDefinition" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RoutingSheetPlanDefinition">
  <xsd:complexContent>
    <xsd:extension base="GenericPlanDefinition">
      <xsd:sequence>
        <xsd:element name="PlanStep" type="RoutingSheetPlanStep" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="GenericPlanDefinition">
  <xsd:sequence>
    <xsd:element name="Parameter" type="Parameter" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="PartsRequired" type="PartGroup" minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="PrecedenceConstraint" type="PrecedenceConstraint" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="ResourcesRequired" type="ResourcesRequired" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="TotalOperationTime" type="Duration" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PrecedenceConstraint">
  <xsd:sequence>
    <xsd:element name="PrecedenceRelationship" type="PrecedenceRelationship"/>
  <xsd:element name="LagDuration" type="ElapsedTimeType" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="LagPercentage" type="xsd:double" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="Description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="AnyItemSatisfiesConstraint" type="xsd:boolean" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

```

<!--

PrecedenceRelationship

These values should be interpreted as follows:

Precedence Relationship Meaning

SS The constrained item cannot start until the constraining items start

SF The constrained item cannot start until the constraining items finish

FS The constrained item cannot finish until the constraining items start

FFThe constrained item cannot finish until the constraining items finish
Source: CMSDIM Part1. pp. 13.

-->

```
<xsd:simpleType name="PrecedenceRelationship">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SS"/>
    <xsd:enumeration value="FS"/>
    <xsd:enumeration value="SF"/>
    <xsd:enumeration value="FF"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="ElapsedTimeType">
  <xsd:sequence>
    <xsd:element name="unit" type="TimeUnit" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Value" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RoutingSheetPlanStep">
  <xsd:complexContent>
    <xsd:extension base="GenericPlanStep">
      <xsd:sequence>
        <xsd:element name="AssociatedMachineProgram" type="ProcessPlanMultiKey" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="AssociatedOperationsheet" type="ProcessPlanMultiKey" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProcessPlanMultiKey">
  <xsd:sequence>
    <xsd:element name="processPlanIdentifier" type="identifiertype"/>
    <xsd:element name="machineProgramIdentifier" type="identifiertype" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="operationSheetIdentifier" type="identifiertype" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="routingSheetIdentifier" type="identifiertype" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="planStepIdentifier" type="identifiertype" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="GenericPlanStep" type="GenericPlanStep"/>
  <xsd:complexType name="GenericPlanStep">
    <xsd:complexContent>
      <xsd:extension base="IdentifyingContextEntity">
        <xsd:sequence>
          <xsd:element name="OperationDefinition" type="xsd:string" minOccurs="0" maxOccurs="1"/>
          <xsd:element name="Parameter" type="Parameter" minOccurs="0" maxOccurs="1"/>
          <xsd:element name="PartsRequired" type="PartGroup" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="RepititionCount" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="ResourcesRequired" type="ResourcesRequired" minOccurs="0"
maxOccurs="unbounded"/>
  <xsd:element name="SetupTime" type="Duration" minOccurs="0" maxOccurs="1"/>
```

```

        <xsd:element name="SpecialInstruction" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="UnitOperationTime" type="Duration" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="OperationDefinitionKey">
    <xsd:sequence>
        <xsd:element name="OperationDefinitionIdentifier" type="identifiertype"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Parameter">
    <xsd:sequence>
        <xsd:element name="Name" type="xsd:string"/>
        <xsd:element name="DataType" type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Value" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ResourcesRequired">
    <xsd:sequence>
        <xsd:element name="ResourceType" type="ResourceType"/>
        <xsd:element name="MaximunNumber" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="MinimumNumber" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Resource" type="ResourceKey" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="AllowableSetup" type="SetupDefinitionKey" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element name="RequiredEmployeeSkill" type="SkillMultiKey" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ResourceKey">
    <xsd:attribute name="resourceIdentifier" type="identifiertype"/>
</xsd:complexType>

<xsd:complexType name="SetupDefinitionKey">
    <xsd:attribute name="setupDefinitionIdentifier" type="identifiertype"/>
</xsd:complexType>

<xsd:simpleType name="ResourceType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="machine"/>
        <xsd:enumeration value="station"/>
        <xsd:enumeration value="crane"/>
        <xsd:enumeration value="employee"/>
        <xsd:enumeration value="fixture"/>
        <xsd:enumeration value="tool"/>
        <xsd:enumeration value="other"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:complexType name="SkillMultiKey">
  <xsd:sequence>
<xsd:element name="skillIdentifier" type="identifiertype"/>
<xsd:element name="skillLevelIdentifier" type="identifiertype" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Duration">
  <xsd:sequence>
    <xsd:element name="unit" type="TimeUnit" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Distribution" type="DistributionData" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DistributionReference" type="DistributionDefinitionKey" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="Value" type="xsd:double" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="TimeUnit">
  <xsd:restriction base="xsd:string">
<xsd:enumeration value="second"/>
<xsd:enumeration value="minute"/>
<xsd:enumeration value="hour"/>
<xsd:enumeration value="day"/>
<xsd:enumeration value="week"/>
<xsd:enumeration value="month"/>
<xsd:enumeration value="year"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="DistributionData">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
<xsd:element name="Description" type="xsd:string" minOccurs="0" maxOccurs="1"/>
<xsd:element name="Value1" type="xsd:int" minOccurs="0" maxOccurs="1"/>
<xsd:element name="Value2" type="xsd:int" minOccurs="0" maxOccurs="1"/>
<xsd:element name="Value3" type="xsd:int" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DistributionDefinitionKey">
  <xsd:sequence>
<xsd:element name="distributionDefinitionIdentifier" type="identifiertype"/>
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

2) Heading Schema used in the method

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema version="1.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.csc.kth.se/~harari/Test_total"
xmlns="http://www.csc.kth.se/~harari/Heading_schema"
elementFormDefault="qualified">
  <xsd:complexType name="HeadingType">
    <xsd:sequence>
      <xsd:element name="Line1" type="xsd:string" />
      <xsd:element name="Line2" type="xsd:string" />
      <xsd:element name="Line3" type="xsd:string" />
      <xsd:element name="Line4" type="xsd:string" />
      <xsd:element name="Line5" type="xsd:string" />
      <xsd:element name="Line6" type="xsd:string" />
      <xsd:element name="Line7" type="xsd:string" />
      <xsd:element name="Line8" type="xsd:string" />
      <xsd:element name="Line9" type="xsd:string" />
      <xsd:element name="Line10" type="xsd:string" />
      <xsd:element name="Line11" type="xsd:string" />
      <xsd:element name="Line12" type="xsd:string" />
      <xsd:element name="Line13" type="xsd:string" />
      <xsd:element name="Line14" type="xsd:string" />
      <xsd:element name="Line15" type="xsd:string" />
      <xsd:element name="Line16" type="xsd:string" />
      <xsd:element name="Line17" type="xsd:string" />
      <xsd:element name="Line18" type="xsd:string" />
      <xsd:element name="Line19" type="xsd:string" />
      <xsd:element name="Line20" type="xsd:string" />
      <xsd:element name="Line21" type="xsd:string" />
      <xsd:element name="Line22" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

3) Tables used to collect information in the production line

Table Name: _____ Route									
No.	Sequence	Operation	Process Time	Unit Min/Sec/Hours	ToolChangeOver	Unit Min/Sec/Hours	Setup Time	Unit Min/Sec/Hours	Extra information

Part	Route	Batch_Size

Measurement Operations	Frequency	Time	Unit Min/Sec/Hours

The Setup Time was added in a separate table, but it is shown together here, as it was considered.