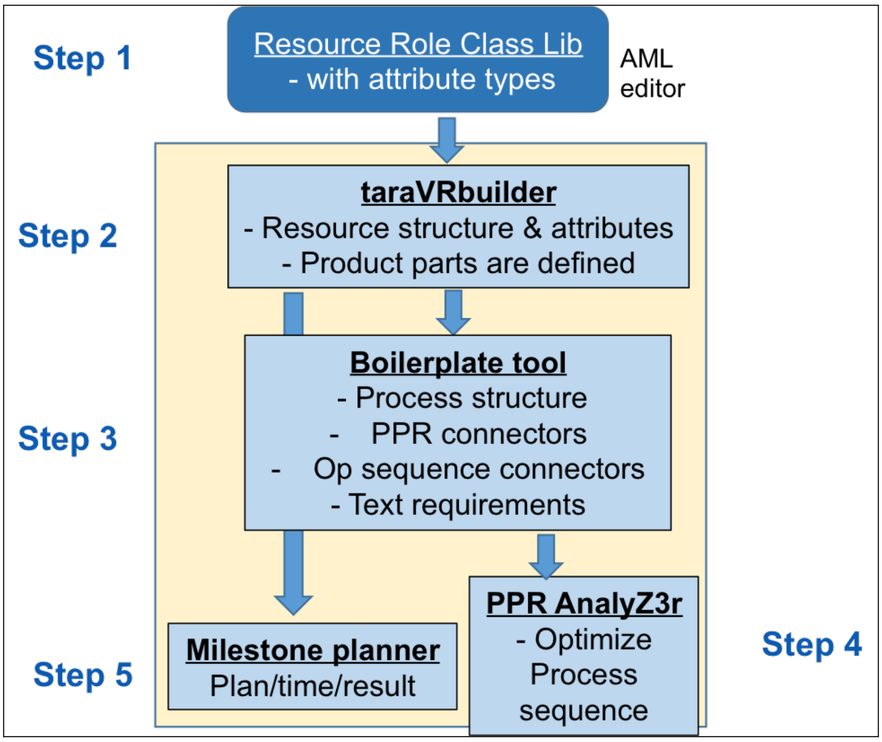
CHALMERS
UNIVERSITY OF TECHNOLOGY



An Engineering Tool Chain for Production System Design

Master's thesis in Systems, Control and Mechatronics

ABDULSALAM JEBER

PETER MALUGE

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

MASTER'S THESIS IN SYSTEMS, CONTROL AND MECHATRONICS

Engineering Tool Chain for Production System Design

ABDULSALAM JEBER
PETER MALUGE

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2019

An Engineering Tool Chain for Production System Design
A formalized requirements engineering approach
ABDULSALAM JEBER
PETER MALUGE

Supervisor: Johan Vallhagen, Volvo Group
Examiner: Knut Åkesson, Department of Electrical Engineering

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Summary of the workflow with the tool-chain used for the ENTOC formalization approach

Typeset in LaTeX
Printed by Chalmers University of Technology
Gothenburg, Sweden 2019

Evaluation of an Engineering Tool Chain for Production System Design
A formalized requirements engineering approach
ABDULSALAM JEBER
PETER MALUGE
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Production system requirements are traditionally presented and written in natural language text with some parts described using images and figures. They are spread over many different file formats and information in these files is not strongly linked giving room for different interpretations of the requirements. This thesis describes the requirements specification formalization approach developed in the ENTOC project together with the chain of tools developed to aid in this formalization. It also gives an evaluation of the formalization approach and the tool-chain.

The evaluation is based on the performance of the formalization on two industrial test cases with the first test case being an existing robot cell with one product assembly and the other being a proposed production cell with two products with one product having multiple variants. Each tool in the tool-chain was used to formalize a specific class of requirements. The taraVRbuilder tool was used to formalize requirements associated to the production cell's layout, its resources as well as the products. The Boilerplate tool formalized requirements in the process description as well as the general or standard company constraints. The milestone planning tool was used to formalize requirements pertaining to time scheduling for commissioning. The PPR_AnalyZ3r tool was used to analyze the feasibility in resource allocation as well as optimizing the scheduling of the different processes involved in the production. The AutomationML file format was used for data exchange between the tools.

The evaluation of the formalization on the test cases showed that a high percentage of the requirements pertaining to the layout, product and resources could be formalized using the approach. For the process description requirements, a high percentage was formalized and the PPR modelling was also formalized with the output being both human readable and computer interpret-able. Most of the standard and general requirements were formalized however a big percentage of the output is not computer interpretable. The evaluation also showed that with the aid of the tools, the time for formalization is greatly reduced. The formalized output is reusable and can easily be edited to cater for any adjustments in the production cell. The output file is human readable however as it currently stands, it cannot be used with any other engineering tools and further work is needed to make this possible.

Keywords: Formalization, Requirements, AutomationML, Production system, Variants, PPR, ENTOC.

Acknowledgements

We would like to express our sincere appreciation and gratitude to our families for the encouragement and support they have provided us throughout our education and without whom we would not be where we are today. We are truly grateful for all the sacrifices they have made for us.
It gives us immense pleasure to thank our supervisors, Johan Vallhagen and Knut Akesson for allowing us the opportunity to do our thesis work with them and for their support, continuous guidance, meticulous suggestions and astute criticism during the entire compilation of this paper.
A special thanks to all the ENTOC project partners who have tirelessly worked on the tool development used in this thesis project and continuously listened to our feedback and complaints. And lastly we would like to thank The Almighty God for His unending grace and mercy that he bestows to all those who trust in Him.

Peter Maluge, Abdulsalam Jeber, Gothenburg, June 2019

Contents

List of Figures

List of Tables

1
Introduction

Today, production system requirements are traditionally presented and written in natural language text with some parts described using images and figures. This however presents a major challenge as the requirements are spread over many different file formats and information in these files is not strongly linked. This gives room for different interpretations of the data presented in these files.

Before production systems are built or updated, a Request for Quotation (RFQ) with requirement specifications is sent out to several companies/solution providers with requests to deliver proposals for solutions and total cost estimates. To simplify this process and also make it more efficient for both the seller and buyer of the solution, it is desirable that the requirements are easily interpret-able by the solution providers. To ensure this, the requirements should follow a predefined structure i.e. a standardized syntax and should not be ambiguous. This is not the case in production systems design today however since requirements are expressed in natural language text without any clear structure leading to misinterpretation.

This thesis work presents a formal approach to the requirements engineering process developed as part of the Engineering Tool Chain for production system design (ENTOC) project, a European Union project in which Volvo Group is a partner.
It is aimed at all parties involved in production system design, that is both the customer and the vendor/solution provider between whom requirement specifications are to be exchanged. The conclusions of this thesis work will be of interest to both these parties as it will improve their efficiency and interpretation of the requirement specifications ensuring the right solution is proposed.

1.1 Background

As is the current practice today for many industries, written statements in a natural language with references to images, figures and tables are used to describe requirement specifications for production systems to be installed or updated. These requirements are stored in form of documents, for example, Microsoft word documents, Microsoft excel sheets or PowerPoint slides. As a result of spreading this information over different file formats, the information about the requirements is usually not linked, in some cases ambiguous and not easily evaluated by computer programs. As a consequence of this, when the requirements are sent over to the solution provider for proposals and cost estimates, these requirements may be misinterpreted.

These shortcomings with the current practice of requirements specification necessitate the need for a formalized approach to requirements specification which would ensure efficiency and do away with misinterpretations, making the process of production system installation or update quicker.

Volvo GTO Research & Technology Development(R&TD) as part of a European research project ENTOC (Engineering tool chain for efficient and iterative development of smart factories), has a goal to develop new methods and technologies for enabling digital design, development, simulation, commissioning and maintenance of production systems. As part of the validation and demonstration of the project results, an industrial use case was developed at Volvo, the basis on which this thesis work was done in order to demonstrate the formalized requirement specification engineering approach developed in this project.

1.2 Aim

This thesis involves demonstrating the methods and tools for the formalized requirements specification engineering approach developed in the ENTOC project on the Volvo test cases. One of these test cases is an existing production cell the other is a new proposed production cell producing product variants. The product being assembled in the first test case is a front lid, a sheet metal component, Body in White (BIW) for cabs, produced in an automated cell with industrial robots. The second test case is a proposed production cell, including an element of product variability with two products being produced in this cell. These products will be the front lid and the trunk lid.

The main aim of this thesis work is therefore to test and demonstrate the formalization approach with the aid of the entire chain of tools developed in the ENTOC Project on the two Volvo test cases and then evaluate the approach's performance on these test cases.
The secondary aim for this thesis is to design the new production cell with product variant handling and apply the formalized requirements specification approach on this cell as well.

1.3 Research Questions

In achieving the aim of this thesis, the following research questions have been answered:

- To what extent does the ENTOC formalization approach and tools cover the requirement specification of the Volvo test cases? This questions aims to explore what percentage of the requirements in the test case specification can be covered using the ENTOC formalization approach.

- What are the re-usability capabilities of the formalized requirement specification from the ENTOC formalization approach? This question aims to explore how re-usabe the output data format from the formalization is in case of modifications to the corresponding production cell.

1.4 Limitations

This thesis work will be limited to using the tools developed by other partners in the ENTOC project, hence it will not involve the development of the tools. The thesis work will also be limited to the Volvo use case and shall not involve testing on other use cases.

In regards to requirements, there are many different types of requirements however in this thesis the focus is on describing requirements related to the production system. This means that the focus is on describing the requirements on the various resources e.g. machines, robots, factory layout etc. In addition, the thesis focuses on describing the relation between the product and the production system where operations and relations between operations play a key part since they describe how the product can be assembled.

1.5 Objectives

Volvo being one of the demonstrators in the ENTOC project, chose an existing production cell for its test case. The RFQ for this cell is available, however as is the current practice for most companies, the requirements are in the form of documents without any defined syntax or semantics. Hence, the first objective for this thesis is to analyze the RFQ and determine which sections of the requirements can be formalized by the formalization approach and tool-chain developed in the ENTOC project.

After the analysis of the RFQ, The next objective is to deploy the tool-chain comprising of the various tools developed for the formalization approach on the production cell to evaluate the tool-chain performance in formalizing the requirements specifications.

After the requirements of the existing production cell have been formalized, the next objective is to conceptualize the new production cell to cater for the production of different products, one of the products having variants. A design layout for the production cell is proposed and the scheduling of the production processes and allocation of resources is done.

The next objective is to analyze the requirements for the new cell and apply the tool-chain to formalize the requirements specifications of the designed new cell to test the tool-chain's performance and also to ascertain the feasibility of the new cell design. This objective will also involve using one of the tools in the tool-chain to optimize the production cell in terms of the production capacity and resource allocation to ensure no bottlenecks in the design.

Once the feasibility of the new cell design has been ascertained, the next objective will be to build a virtual model of the production cell. This objective will be to demonstrate how the resulting requirements from the formalization and the available layout data can be used for an efficient, consistent and computer-assisted approach for equipment design, virtual commissioning, and approval of plant functionality.

1.6 Thesis outline

Chapter 2 gives a detailed background of the thesis as well as an explanation of the Volvo test cases on which the thesis is based. It also describes the evaluation criteria used for the evaluation of the formalization approach.

Chapter 3 gives a literature review on requirements specifications and explains their relation to production system design and installation. It also describes the current requirements specification engineering approaches being used at Volvo GTO.
The chapter further explains the need for formalization of the requirements and also explains the proposed approach in the ENTOC project. The chapter also describes the tools that make up the tool-chain developed for the formalization and the application of each tool in the formalization process.

In chapter 4, a description in detail of each of the steps taken in the application of the formalization approach with the aid of the tools on the Volvo test cases is given. The sequence in which the tools are used is further highlighted and the outcome/result from each of the steps is also discussed.
To make the thesis easily readable, the results from each step will be given after each step so the reader can easily follow the sequence of the tool-chain use as its being applied to the test case.

In chapter 5, the the evaluation of the approach is give according to the evaluation criteria described in chapter 2.

2
Thesis Background and Overview

The first step in the engineering process for the installation or updating of a production system is referred to as requirements engineering [1]. This step involves the customer i.e. the company wishing to have a production system installed or updated compiling a requirement specification detailing a list of requirements for the production system. This requirements specification is sent out to a number of companies often referred to as solution providers or vendors. The vendors analyze the requirements and propose various solutions to the requirements of the customer.

Today as is the practice at Volvo, the requirements are written in natural language text with some sections described using images and figures. The requirements are spread out over many different file formats for example Microsoft word documents, Excel sheets and PowerPoint presentations. This makes it difficult to link up the information in these documents. When the requirement specification is sent out to the vendors, there are many possible interpretations of the information contained in these files. This may result in the vendors providing solution proposals that do not match the requirements the customer intended. The vendors also have a challenge as it is time consuming to analyze information from all the different files that make up the requirements specification.

This process can be made efficient for both the customer and the vendor if the requirements sent out are easily interpreted by the vendors. This can be achieved by ensuring the requirements follow a certain structure, i.e., follow a syntax, and that they should not be ambiguous. This structure should be in format that can be interpreted by a computer program as well as be readable by a human reader. This is referred to as formalization of the requirement specification.

A formalization approach to the requirements engineering process has been developed in the ENTOC project. A tool-chain with three tools was developed to aid in the formalization of the requirements. Volvo Trucks being a partner in the project was asked to come up with two test cases to evaluate this formalization approach.

Volvo came up with two test cases, the first test case is a robot cell for the production of a front lid. This robot cell is already up and functional at the Volvo Umeå production plant. In this cell, two panels i.e. the inner and outer panels are assembled to form the front lid. The second case is a modification of the first test case. In the new cell, two products are to be produced that is the front lid as well as a trunk lid. In this case, the two products have two separate process descriptions, however they share some of the resources in the cell. This test case will aim to show how the ENTOC approach handles product variability in terms of resource allocation and optimizing the process sequence.

This thesis demonstrates the application of the developed formalization of requirement specification approach on the two Volvo test cases. It evaluates the performance of the approach as well as the tools developed to aid in the formalization. The evaluation of the approach shall be based on the following criteria;

- **Evaluation Criteria 1 - Requirements coverage.**
  This evaluation will be based on statistics of how many requirements in the specification can be formalized using the approach.

- **Evaluation Criteria 2 - Compatibility and use in other Production system design tools.**
  This evaluation will be based on the ability to use the formalization output in other production system design tools currently being used at Volvo for example PLM systems or production system simulation software.

- **Evaluation Criteria 3 - Re-usability of formalized output data file.**
  This evaluation will be based on the ability to reuse or modify the formalization output file when changes to the production system are implemented and on the ease with which this can be done.

3

Formalization of the Requirements Engineering Specification

This chapter gives an overview on requirements engineering, what it is, what it involves and also what role it plays in the engineering process of designing and installation of production systems. It also highlights what requirements are usually specified in production systems, the different types of requirements as well as the requirement classes considered for this project. The chapter also explains formalization of requirements specifications, its necessity and highlights the importance of formalization and the formalization approach designed in the ENTOC project. The chapter ends with a description of the tool-chain developed for the formalization and an overview of each of the tools that comprise the tool-chain.

3.1 Requirement Engineering and Specification

Figure 3.1 shows an overview of the engineering process commonly deployed for the design of a production system [1]. The figure shows the different steps followed in the engineering process and for each process it defines the outcome or deliverable from that process. In this thesis, emphasis is placed on the first step in the engineering process which is requirements engineering. As shown in the figure, the deliverable for this step is the requirements specification.

Requirements engineering is a cooperative, interactive and incremental process which obtains the development of requirements out of abstract specifications with the goal to determine, analyze, understand and establish requirements [2]. It comprises all the activities and tasks associated with discovering, evaluating, recording, documenting and validating the requirements for a particular project. In this step, requirements are discovered, analyzed, specified and verified.

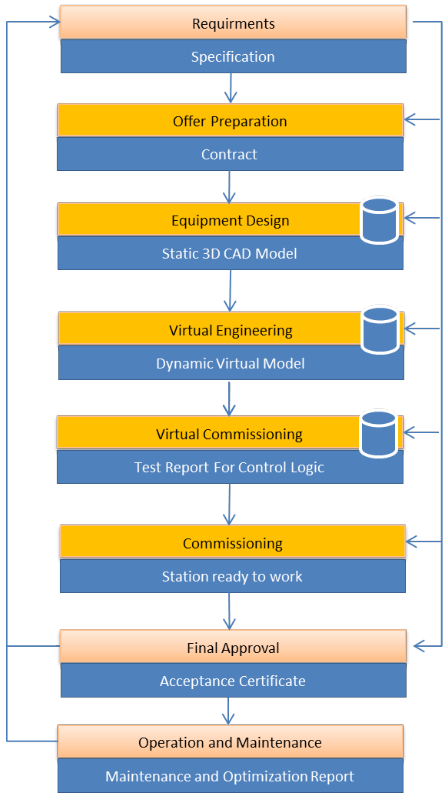


Figure 3.1: Overview of the Engineering process for a production system [1].

The output of the requirements engineering process is the requirements specification. The requirements included in the requirements specification are differentiated into functional and non-functional requirements and have the ability to be varying in abstraction and detail. They are mostly written out of the authors' technical understanding and perspective. A functional requirement will describe a particular behaviour of function of the system when certain conditions are met. Non-functional requirements on the other hand generally specify the system's quality attributes or characteristics [3].
The requirements specification in most cases is a set of files in different file formats for example Microsoft word documents, Excel sheets, PowerPoint presentations as well as other documentation file formats.
The requirement specification for a production system typically contains several requirements which can be grouped into different requirement classes. Some of common requirement classes for production systems are given below;

- A production process specification.
- A rough specification of the plant hierarchy, including main aggregates and specifications of essential aggregate characteristics.
- A specification of the plant layout constraints.
- A detailed time schedule for erecting and commissioning.
- A set of environmental constraints.
- A set of general company constraints
- A set of site specific constraints.

3.1.1 Requirement Engineering Aspects within the ENTOC Project

As stated in the above section, the requirement specification for a production system consists of a number of requirement classes [1]. For the ENTOC project as well as for this thesis, emphasis has been put on selected classes. These classes are process description, layout planning, plant hierarchy, time scheduling for commissioning, and general company constraints. The classes are described in more details below;

- **Process description**
  The production process is described based on the product to be produced and the layout of the plant. The first step is to describe the production process briefly by listing all steps of the process flow chronologically and by depicting all components and resources needed to be able to produce the product in a 2D layout including names and types.
  The next step is to detail the description of each step in the process flow.
  - Which operation to be performed.
  - What resources to be used.
  - How to perform the operations.
  - By whom (resource or operator).
  - Transportation from an operation to another.

  For production systems with product variants, all steps are described for all product variants.
  In summary, the process description class of requirements constitutes requirements related to the processes and sequences of processes that are involved in production.

- **Layout planning**
  Layouts and visualizations are often used as an assistance to illustrate requirements written in natural language and tables. During bidding-processes in many cases the layouts are supplemented by phrases like "proposal for solution" or the supplier is called to design an alternative solution. Usually the layouts include real requirements, but also proposal for solutions, for example in a brown-field project: In summary, this class of requirements includes requirements associated with the resources and the resource layout for the production system.
  - Use-able shop-floor area and headroom.
  - Machine(s) and infrastructure that have to be used.
  - Existing machines at a fixed location.
  - Existing machines, which can be moved freely or to a restricted location.
  - Proposal how to connect machines by conveyor technology, but the supplier is called to design an alternative solution.
  - An exactly defined system (the supplier is not allowed to change the layout).

In summary the layout class of requirements consists of requirements related to the layout of the production system with emphasis on the positioning on the resources.

- **Plant hierarchy**
  One of the processes involved in crafting a production system is to engineer the production system. Before this is done, the production process is roughly known and so one proceeds by splitting the process into various sub-processes and to think about technical components needed to realize each sub-process. In doing so, a plant hierarchy is created whose structure and internal relations between components needs to be maintained. Hence the plant hierarchy class of requirements constitutes those requirements related to the plant structure and relations between the different components that make up the structure.

- **Time schedule for commissioning**
  In this class of specifications, the dates for when the different processes involved in building the production system to when it is commissioned are given. These dates are commonly referred to as milestones and are commonly required by the customers for whom the production system is being built.

- **General company constraints**
  To guarantee that the new production equipment or tool will meet the requirements for example capability and target values, the companies requires specific corporate standards and guidelines to be applied. In the standards and guidelines it is defined, for example how the capability for different application should be measured, and how test inspections should be carried out.
  In summary, this requirements class constitutes requirements related to the company standards that the production system must adhere to.

3.1.2 Problems associated with traditional Requirements Engineering Specialization Approach

This section explains the issues that arise in the traditional requirements approach due to lack of formalization. The issues are explained for each of the requirements classes given in the section above.

- **Process description**
  In section 3.1.1, it was stated that the process description class of requirements constitutes requirements related to the processes and sequences of processes that are involved in production.
  In the current requirements engineering approach without formalization, the process description is made in natural language text and stored in for example Microsoft word documents and other text formats with references made to layout visualizations stored in separate files. In some cases for example for the Volvo Test case, PowerPoint presentations are included in the specification to

show how the processes relate to the product as well as what resources are needed for these processes. Flowcharts are also used to show the sequence of these processes.
The main problem with this approach is that the reader has to browse through different files and file formats to be able to make sense of the whole process description. This can be time consuming and increases the amount of time the vendor or solution provider takes to analyze the requirements.

- **Layout planning**
  Section 3.1.1 stated that this class of requirements includes requirements associated with the resources and the resource layout for the production system. It also stated that in the current approach, this specification usually includes proposals for solutions.
  The problem with this approach is that for the layout's author i.e. the customer as well as the reader i.e. the solution provider/vendor it is time consuming to differentiate between requirements and "proposal for solutions" in these layouts. Another issue with the current approach is since images, figures and visualizations are used to describe the layouts, the readers have to frequently switch between layout images and text to be able to understand the definition of the requirements.
  The other problem with the current approach is that since the process descriptions which are in text format are given in separate documents or files, usually the layout images, figures or visualizations do not have any reference to the process description or in cases where the references are made, it is time consuming for the reader to switch through the two separate files to make sense of the references.

- **Time schedule for commissioning**
  In section 3.1.1, it was stated that this class of requirements comprises of requirements pertaining to the time plan and deadlines for the different processes involved in the engineering process of the production system.
  In the current requirements engineering approach without formalization, it is common practice to provide the time plan as a spreadsheet with the process and the planned time span i.e the start and end times provided. This file however adds to the number of different documents the solution provider has to go through in order to understand the requirement specification. The main issue in this case is the amount of time spent in analyzing the different documents, time which could be saved with formalization.

- **General company constraints**
  As stated in section 3.1.1, this requirements class constitutes requirements related to the company standards that the production system must adhere to. This requirements class also includes a description of the stock of spare parts available for use in the production system. Hence as part of the specification, the vendor provides the solution provider with a list of allowed equipment as well as a list of available spare parts.

The issue with this approach arises from the fact that this information is exchanged or provided in tabular form in spreadsheets or text file format. This makes it cumbersome for the solution provider since he has to go through several spreadsheets to pick out the information and this is time consuming.

3.2 Formalization

As explained in section 3.1, the outcome of the requirements engineering process is a requirements specification. The requirements specification consists of different requirements grouped in different requirement classes. Requirements are usually written in natural language (e.g. English), The main issue with writing requirements in a natural language however is that they cannot serve as inputs for other tools used in the engineering processes for example PLM systems and also leave room for misinterpretation of the requirements. This gives rise to the need for formalization.

Formalization involves the use of a formal language/format which can be understood by a computer program and is also not ambiguous. A formal language or format ensures that a given sentence cannot be understood in different ways. This implies that humans too can benefit from using a formal language/format since the requirement will be fully understood without a prior interpretation from the user [4].
With formalization, the requirement specification is formal in the sense that it follows a specified syntax and the format used can be exchanged between different engineering design tools throughout the whole engineering process. Formalization also enables automatic verification of the conformity of the designed production system to the specified requirements.

The next section presents the formalization approach developed in the ENTOC project and also presents ways in which the approach addresses the problems associated with the current requirements engineering approaches.

3.3 ENTOC Formalization Approach

The main goal of the ENTOC project is the development of an engineering tool chain for efficient and iterative development of smart factories, which provides the possibility to create and use valuable engineering information through the whole development process of those factories. This was the basis on which the formalization approach for the ENTOC Project was developed. The goal of this formalization approach is therefore to create a requirements specification format that can enable data/information exchange between different tools in the engineering process.

This approach is built on existing standards together with a chain of tools developed during the project to aid in the formalization. In the sub-section below, an overview of the standards used is given to give the reader an understanding of how the approach works.

3.3.1 Standards and Concepts used in the ENTOC Formalization Approach

3.3.1.1 Automation Markup Language (AutomationML)

AutomationML was developed as a data exchange format for all production system engineering data relevant for the whole life-cycle of production systems. It provides concepts for system modeling following object-oriented paradigms thus enabling lossless bilateral data exchange and furthermore the development of systems for centralized data management and engineering artifact libraries [5].

This format can be applied for lossless data exchange along various chains of data processing systems (including engineering tools) of all industrial areas and beyond without any limits related to licensing and application costs.

AutomationML is applicable for discrete manufacturing systems, the process industry, in large enterprises and small and medium-sized companies, in basic and detailed engineering as well as virtual commissioning, physical realization, and commissioning. In addition to that, the interaction of different companies is realized, including OEM's, system integrators and component/device vendors.

AutomationML Basic architecture
AutomationML has a lean and distributed file architecture as explained in [6]. It does not define any new file format but combines existing established XML data formats as shown in figure 3.2, which have been proven in use for their specific domain. These data formats are:

- CAEX format [7] for object topologies including hierarchies, properties and relations of objects.
- COLLADA 1.5.0 and 1.4.1 format [8] for geometries and kinematics of objects.
- PLCopen XML 2.0 and 2.0.1 formats [9] for the discrete behavior of objects.

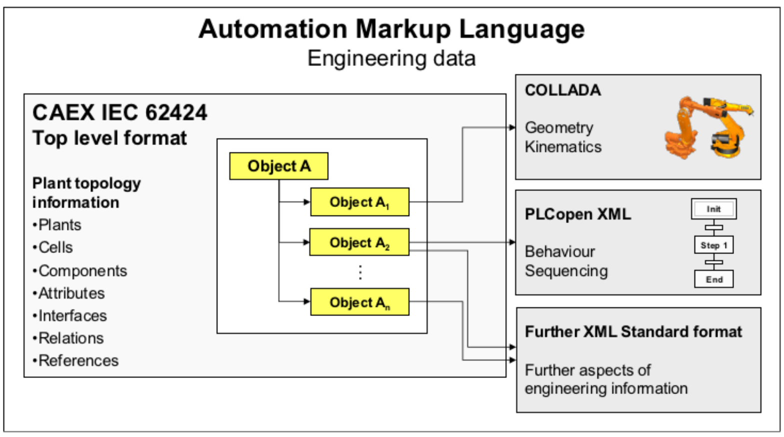


Figure 3.2: AutomationML basic architecture.

- **CAEX Format**
  CAEX forms the core of AutomationML. A CAEX object is a data representation of any asset. It can model physical assets, e.g., a motor, a robot, a tank; or abstract assets like a function block, a model or a folder. CAEX allows linking those objects to systems since every physical or logical system is characterized by internal elements (objects) which may contain further internal elements, and all elements may have interfaces, attributes, and connections with each other. CAEX also allows the modeling of any plant topology, communication topology, process topology, and resource topology.

  The CAEX format provides the following concepts in regards to production systems:

  - Description of instance hierarchies:
    To describe an existing production system, the definition of its system elements is necessary. Those system elements are called InternalElements within the CAEX format. A system element gets a name and also a unique identifier (GUID). Attributes and attribute values can be assigned to the system elements. In addition, it is possible to describe relations between those elements. The concept of CAEX InterfaceClasses is used for that purpose. If the production system is a complex structure of aggregates and sub-aggregates, it can be expressed with CAEX by using an unlimited number of sub-structures of InternalElements (IEs of IEs of IEs . . .). Thus the description contains a hierarchy of system aggregates.

    In addition, the CAEX InstanceHierarchy can be used for the definition of arbitrary hierarchical and networked structures, for example, the description of processes. For this purpose, the first level of the CAEX InstanceHierarchy describes an InternalElement called ProcessHierarchy with corresponding InternalElements constructed below this element. The process hierarchy element can contain InternalElements of possible roles, e.g. Process, State and Transition.
  - Description of roles:
    If a new production system is designed, then at the beginning often a brainstorming about the production process and the rough structure of the system is performed. CAEX may be used to manifest the results of those discussions.In order to track the development of system components, it is necessary to provide names to these components and to define the structure of the plant in terms of aggregation relations. These tasks can be realized by using CAEX instance hierarchies.

    At this level of engineering, it is not possible to determine the values of attributes of the system components. But it is possible and necessary to define the role of the system components within the future production processes. A role may be described by a definition of requirements regarding the attributes of the system components. For that kind of role definition, CAEX introduces the concept of RoleClasses in [10], which supports inheritance concepts known in object-oriented programming languages. The internal elements defined in the first engineering step are linked to RoleClasses. This approach leads to a model for the intended production systems in terms of a defined system structure and role-based requirements statements regarding the system elements.

- Description of equipment types:
  The concept of CAEX SystemUnitClasses can be utilized if a vendor of production system components intends to promote catalogs of their products. These SystemUnitClasses can be used as templates for equipment types. An important engineering step is the assignment of a SystemUnitClass to an InternalElement, which is described within the CAEX standard as the process of copying all sub-structures of the source-SystemUnitClass to the target-InternalElement. The definition of SystemUnitClasses follows object-oriented principles and provides features like an inheritance. One main difference to CAEX RoleClasses is that attributes are not used to create requirement statements but to define promise statements.

■ **COLLADA 1.5.0 and 1.4.1 Format**
COLLADA stands for COLLAborative Design Activity. It was developed by the KHRONOS association under the leadership of Sony as an intermediate format within the scope of digital content creation in the gaming industry.
It was designed to enable the representation of 3D objects within 3D scenes covering all relevant visual, kinematic, and dynamic properties needed for object animation and simulation.
COLLADA is an XML-based data format with a modular structure enabling the definition of libraries of visual and kinematic elements. It can contain libraries for the representation of geometries, materials, lights, cameras, visual scenes, kinematic models, kinematic scenes, and others as explained in [11].

■ **PLCopen XML 2.0 and 2.0.1 Format**
Logic information is an important aspect for raw system planning, electrical design, HMI development, PLC and robot control programming, for simulation purposes, and virtual commissioning. To support the different phases in the iterative production system engineering process covering different levels of detail, AutomationML needs to be able to store logic information from different tools and disciplines. AutomationML offers this functionality as explained in [12].

The AutomationML format with its properties as explained above is hence the data exchange format that was chosen for the formalization approach in ENTOC. To supplement its use to ensure a larger coverage of the requirements, another standard called the BoilerPlate was chosen and its described in the section below.

3.3.1.2 Boilerplates

Usually, automotive production systems' requirements are defined in natural language. These requirements don't have a particular structure because they described in many large documents and it is ambiguous to understand the composition. Sometimes many requirements specified in one sentence. However, the separation of requirements is a pre-condition to building traceability concepts or hierarchical requirements structures. Currently, the explanation of different types of requirements differs in the used words, the grammar, and level of formal abstraction.

The boilerplate according to [18] predefines a sentence to describe logical requirements, events, behavior, etc. The boilerplate has placeholders to explain the context and specifics of its boilerplate instance. Instances of different boilerplates can be combined. Every complete trace from a starting word to an ending word is an individual boilerplate belongs to one of these groups listed below.

- **Logical requirements**
  Logical requirements are used to define the logical relations. These relations could be a comparison of an object with a condition or behavior of a system. The following example explains the structure of these Instances type. "If <system> <process verb> <object>".
- **Event-based requirements**
  Event-based requirements are used to describe the requirements that express time variant between the occurrence of events. An incident could be a defined event or an action of a system. As an example,''as soon as <system><Process Verb><Object>".
- **Time conditioned requirements**
  Time conditioned requirements are used to describe requirements with a dependence of time duration. The time duration is defined by entering or leaving a state, or by process activities performed by an actor or system. An example illustrates the structure. ''As long as <system> is <in the state>".
- **Object description requirements**
  The object description requirements are used to describe the properties of an object. a property defined by value and expression operator (e.g., equal to, greater than, less than, etc.). A boilerplate that describes a property of an object, it contains the object itself, the required property, and the subject of the requirement.
- **Safety and security requirements**
  It used to describe safety requirements.This boilerplate requires a condition that must be fulfilled before an object acts.

3.3.2 Description of the Approach

In this approach, several concepts and language constructs have been developed. These language constructs can be used independently from each other but in this approach, they have been developed to be used complementarily. The language constructs have been described in low level syntax elements provided by the AutomationML standard described in section 3.3.1 which makes it possible to use one generic parsing algorithm for usage of all concepts for the different tools aiding the formalization process. The developed language constructs also make it possible to describe important aspects regarding the product, production system resources, processes as well as timelines for the creation of the production system.

The ENTOC approach aims at covering a number of aspects involved in the production system development process however this thesis work covers just some of these aspects and these are; Project Planning, Production Resource Planning, Process-Driven Production Planning, PPR-Model-Based Production Planning and finally Requirement Specification based on Natural Language Boilerplates.

Engineering Aspect	Coverage by ENTOC Approach
Production Resource Planning	Definition of a resource structure and attributes in AutomationML with the taraVRBbuilder tool.
Process Driven Production Planning	Definition of a process structure and process sequence in AutomationML using the boilerplate tool.
PPR Model-Based	Modeling of PPR structure using connectors and links in AutomationML using the boilerplate tool.
Product Variability Handling	Optimization of process sequence for different process description each corresponding to a different product using the PPR_AnalyZ3r tool.
Project Planning	Assigning milestones, plans , result types for the different resources in the system using the Milestone planning tool.

Figure 3.3: Summary of how the ENTOC approach covers the different Engineering aspects.

A detailed description of how the approach covers each of these aspects is given below;

- **Project Planning**
  The engineering and commissioning of a production system is called a project in the following ways. It involves time, resources and efforts. The time when the system should be ready for production is a key non-functional requirement of a customer of such a system. Customer and supplier of the production system often define intermediate time points to monitor the project progress especially within large projects. Those time points are called mile-stones of the project. Within large projects the supplier engages sub-suppliers resulting

in the definition of partial milestones.

The goal of the ENTOC approach in regards to mile-stone planning is to provide formal representations of mile-stone plans as non-technical requirements to be included into final offer as base for contract designs. A derived goal is to provide possibilities to track intended results at mile-stone dates during the project.
Figure 3.4 below shows the formal representation of the mile-stone plan according to the ENTOC approach in a conceptual visualization.

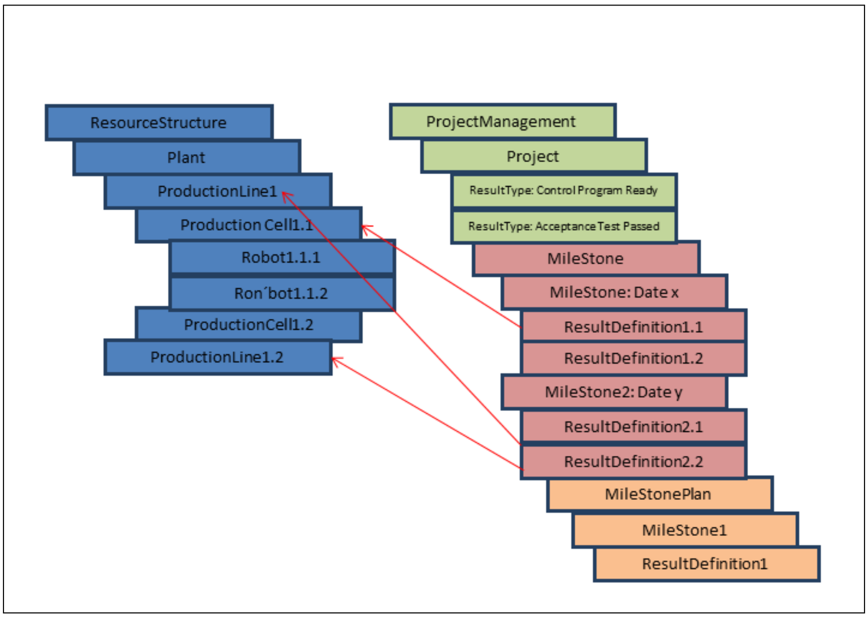


Figure 3.4: Conceptual visualization of the milestone mappings using the ENTOC formalization approach.

In the approach, the resource structure (marked in blue in figure 3.4) is defined by a hierarchical tree of the equipment. This will be available in AutomationML format as an export from one on the tools developed for the project. An additional tree of information defines the project management information. The project declaration part is colored green in figure 3.4. It contains definitions of result types, for example "Acceptance Test Passed". This list of result types can easily be extended per project. Those result types have to be related to parts of or the whole production system. Thus special nodes called "Result-Definition" interlink the result types with elements of the resource structure within the first-level mile-stone plan (colored red in the figure 3.4). Any kind of those result definitions are attached to mile-stones, which carry information about due date and time. Multiple milestones are assembled to form complete mile-stone plans.
An advanced and optional concept is to define sub-mile-stone plans (e.g. the orange colored sub- mile-stone plan in Figure 3.4) to define mile-stone activities in more detail.

Figure 3.5 shows an example of the formalization representation in an AutomationML format. In the example, resource structure consists of equipment for cockpit installation in a car production context. The raw design contains three stations for part supply, body positioning and cockpit mounting. This resource structure is referenced by a kind of project plan, which is implemented as mile-stone plan. The current plan contains definitions of "control program is ready" and "acceptance test has passed" as result types for milestones. The example shows how to use them for a result definition of mile-stone 001. This result definition contains references from resource "station 1" to "control program is ready", which translates to the definition "when mile-stone 001 is due, then the control program for station 001 has to be ready."

Figure 3.5: Example showing a milestones plan definition in AutomationML using the ENTOC approach.

To achieve this formalization, the MileStone Planner tool has been developed and its functionality is explained further in section 3.3.3

- **Production Resource Planning**
  During the design of a production system, the designer has to consider the equipment required for the production as well as the layout of the production system. In the requirements engineering phase, the designer has to specify the requirements on the particular resources as well as the geometry and restrictions on the layout.
  In the ENTOC formalization approach with the AutomationML format as the data exchange format of choice, these specifications are formalized by defining a resource structure as an instance in InstanceHierarchy with the resources forming the InternalElements of this structure. The requirements associated with the different resources are then added to each of the resources as at-

tributes.

To formalize the modular nature of some of the resources, the hierarchical tree will still be applied with the components of a particular resource being represented with a parent-child relationship with regards to the internal elements.

Figure 3.6 shows this formalization as implemented in the AutomationML format. In the example the resource structure for the production system is defined and also the attributes of one robot are defined.

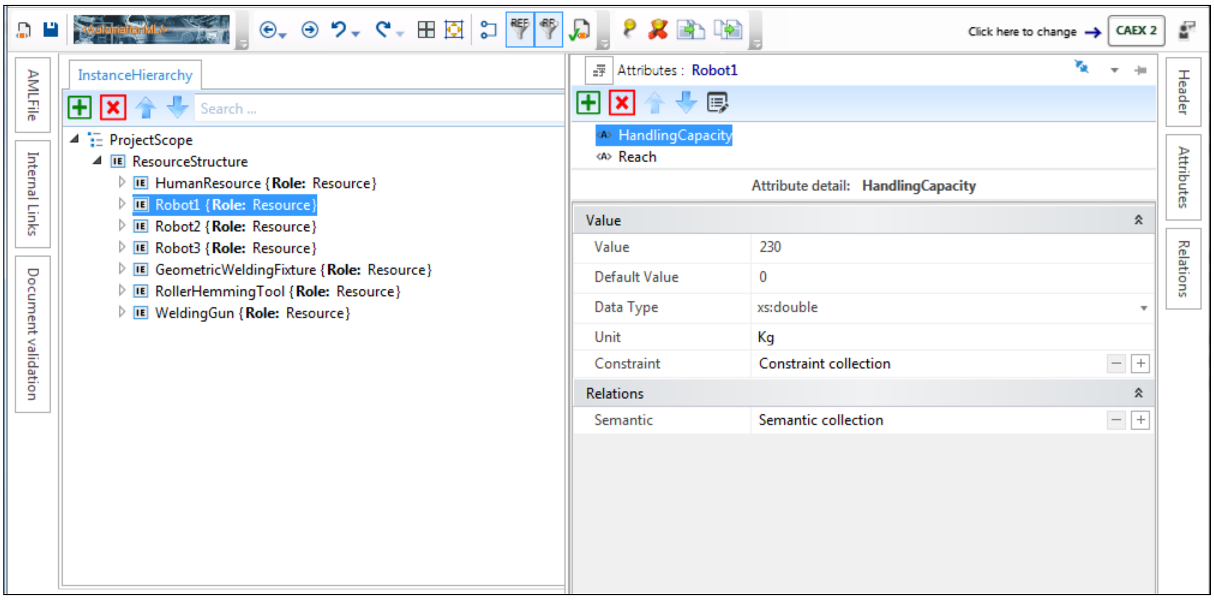


Figure 3.6: Example showing Resource Planning in AutomationML as formalized using the ENTOC Approach.

To achieve this formalization, the taraVRbuilder tool has been developed and more information about this tool is given in section 3.3.3.

- **Process-Driven Production Planning**

  A core task in the design of a production system is the process planning which usually results in the production process description. In this task, the sequence of the production process steps is described from the first process to the final process.

  In the ENTOC formalized approach, process description is formalized in AutomationML format by defining a Process structure as an instance InstanceHierarchy with the processes forming the internalElements. The requirements for each process are added as attributes of the corresponding InternalElement.

  To formalize the sequence of the processes, Interfaces with names such as “ProcessFollowsProcess” or “ProcessPreceedsProccess“ are defined for each process’ internalElement. Links are used to show the sequence or order in which the processes are carried out. These links connect the interfaces of the processes following each other in the sequence.

Figure 3.7 shows an example of the process description in AutomationML formalized with this ENTOC approach. The example also shows the use of InternalLinks and Interfaces to show the sequence formalization. In the example, the requirements for the StructuralGluing process are shown formalized as attributes for the StructuralGluing InternalElement.

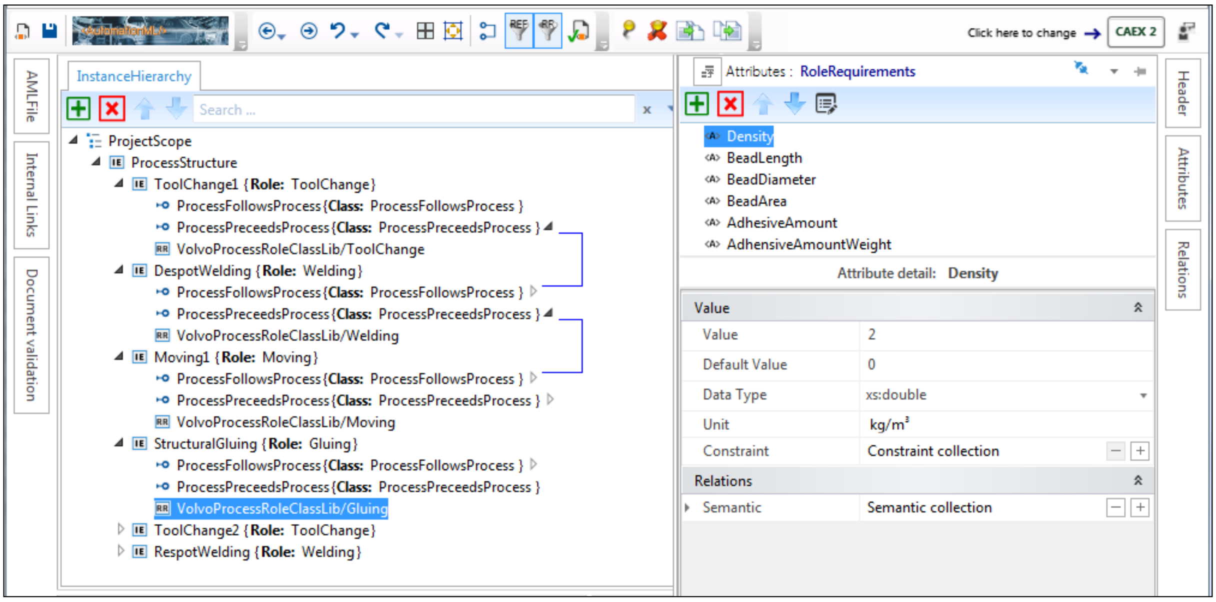


Figure 3.7: Example showing process plan formalization using the ENTOC Approach.

- **PPR-Model-Based Production Planning**
  With the rising complexity, increasing variant diversity and the goal of shorter time-to-market the importance of simultaneous engineering is as high as it never was. This aspect implicates challenges in every phase and discipline of the product development process. One of the most time consuming challenges is the transfer of data from mere concepts into requirement specifications. One of the major causes of this challenge is the interdependence of the disciplines involved i.e. Process planning needs information about the product , resource planning needs information about the process description. To counter this challenge, the product-process-resource (PPR) model is required hence the need for PPR model based production planning.

  The ENTOC formalization approach addresses PPR model based planning by formalizing it using AutomationML functionality. In the approach, product, process and resource structures are modelled as instanceHierachies with the corresponding objects defined below them as internalElements. For each PPR relation, Interfaces are defined for the associated product, resource and process. These Interfaces are given names for example, “ProcessUsesResource”, “ProductusedbyProcess”, ”ProcessRequiresResource” etc. To show the PPR relationship, internalLinks connect the interfaces of the respective objects in the PPR model.

Figure 3.8 below shows an example with this formalization done in AutomationML. In the example, the PPR relationship between the Despot Welding Process, the resources required for this process and the products involved for this process are shown as well as the internalLinks and PPR interfaces to complete the PPR relationship.

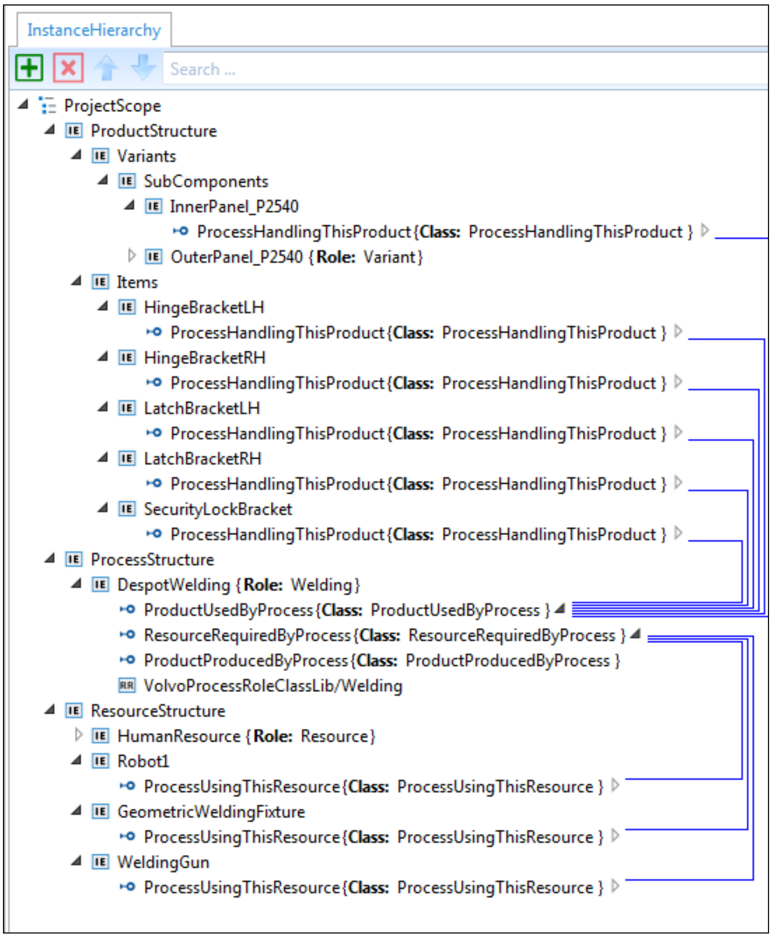


Figure 3.8: Example showing the PPR relationship in AutomationML as formalized using the ENTOC Approach.

- **Requirement Specification based on Natural Language Boilerplates**
  Requirements based on natural language are human readable but often ambiguous and hence interpreted differently by different stakeholders The main problem is this ambiguity, which often results in misunderstandings between the stakeholders on the customer side that define and extend requirements (customer, OEM, general contractor) and the stakeholders on the contractor side that develop solutions for the production plant based on the understanding they get from the requirements (general contractor, contractor, provider of material handling systems, PLC programmer). These misunderstandings result in the development of production plants, which do not satisfy all of the customer's intended requirements, because these production plants are built

to satisfy wrongly understood requirements.

The ENTOC formalization approach addresses these issues with natural language boilerplates. In this approach, boilerplates have been generated and are composed of boilerplate elements arranged in a specific sequence. Different boilerplate elements have been defined for example an element called “Subject” can be a resource or a process in the production system. A boilerplate’s specific sequence of specific boilerplate elements carries a specific semantic meaning. The semantic meaning follows conventions of a spoken language but because each boilerplate has only one specific semantic meaning it is uniquely determined.

The formalization using boilerplates is expressed in AutomationML by defining the different elements of the boilerplates using InternalElements. These elements are then connected to one another using InternalLinks to define the entire boilerplate representation of the requirement.
Figure 3.9 shows an example of this formalization as done in AutomationML. In this example, the requirement expressed in natural language is "Welding programme shall be with adaptive regulation". It is added to the AutomationML using a specific boilerplate.

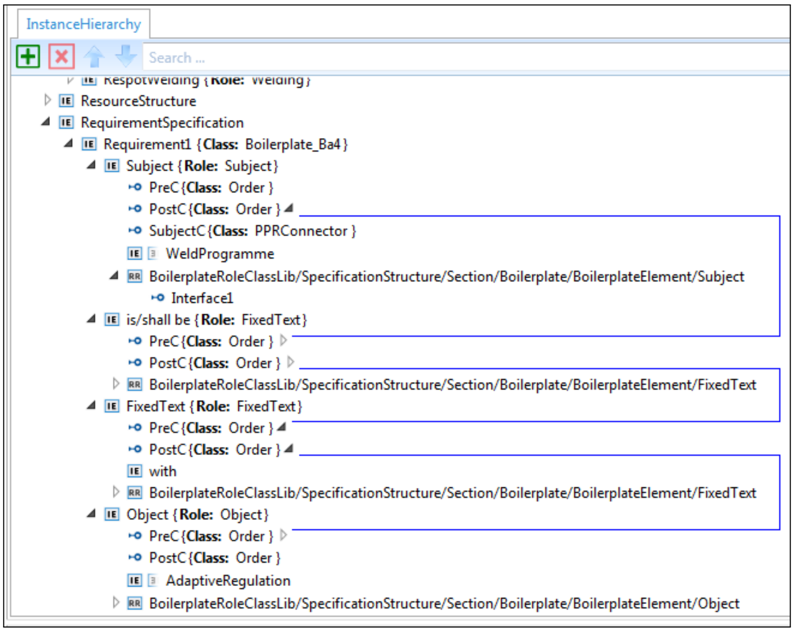


Figure 3.9: Example showing the formalization of a natural language requirement using boilerplates in AutomationML.

- **Product Variability Handling**
  The difference between the products can be created through different variants. These variants could be a hardware or software all the variants that have the same functionality or nature grouped together under the name variant group (VG). Besides, form these variants groups can get many product configurations (C), but not all of them valid product according to configurations roles.

Variants Group (VG)	Variant V	Variant Name	Group Cardinality
Frame VG1	V1	Frame rigid	Choose exactly one
	V2	Frame tractor	
Cab VG2	V3	Cab VI	Choose exactly one
	V4	Cab V2	
Accessories VG3	V5	Lower light bar	Choose at least one
	V6	Head lamp protector	
	V7	Wind deflector	

Table 3.1: Description of the variants and constraints used in the example.

Each valid product has own Bill of Material (BOM), and each of valid product needs one or more process to be assembled. In the table shown below explain an example for the relation between the variants groups and the variants also shows the configuration roles (Group Cardinality) which shows the number of variants that should be chosen from that specific variant group for each valid product.

In the example below shows, two different configurations C1 and C2 both define a product but the valid product should fulfill the configuration role.

C1 : V1, V3, V4, V5, V7
C2 : V1, V4, V5

The product invalid in the configuration C1 because it does not fulfill the Group Cardinality V3 and V4 belong to the same group variant VG2 and the configuration role in that group say choose just one variant. The configuration C2 is a valid product since satisfying the roles. Additional constraints have been considered in the table shown below.

Additional Constraints
(V1 and V3) OR (V1 and V4)
(V1 and V6) OR (V4 and V7)

After adding more restrictions the configurations C1 and C2 become invalid. The configuration C3 shown below fulfills both the cardinality constraints and the additional constraints.

C3 : V1, V3, V5, V6

- Job Shop Problem
  Each valid product needs one or more operations to be produced and this operation in a particular sequence. Moreover, each operation needs a re-

source or more. Scheduling the operations is a common problem and it is called JobShop problem. The problem is how to schedule the operations and the available resources to produce a valid product and minimize the makespan. The example shown below explains the problem.
Example:
Consider a production line with three products being produced. Each product goes through a sequence of processes and the production line consists of three resources. Each process uses one resource and is represented by (Name of resource, Duration of the process). The process sequences are listed below.
Product 1: (Resource1, 4), (Resource2, 2), (Resource3, 2)
Product 2: (Resource1, 2), (Resource2, 4), (Resource3, 2)
Product 3: (Resource2, 4), (Resource3, 3)

As shown above, each product needs one or more process to be produced and all the processes share the same resources such as Product1 use Resource 1 for 2 time units and Product2 use the same Resource for 1 time unit. The solution of the job shop problem is managing the resources to be used for all the processes without overlapping and minimize the makespan as shown in the figure 3.10.

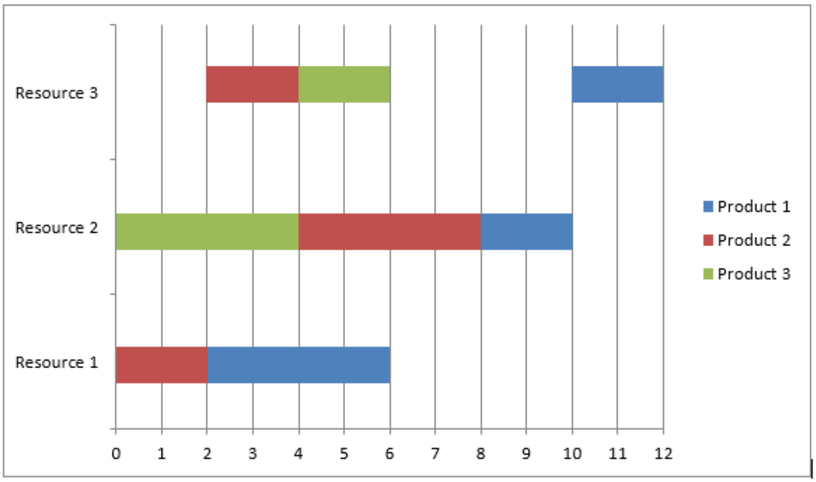


Figure 3.10: Process Scheduling.

3.3.3 Tool-Chain used for the ENTOC Formalization Approach

In section 3.3.2 above, the ENTOC requirement specification approach has been described. As part of the project, several tools which are suited to create and validate the requirements specification were developed. As stated earlier, the goal of this thesis work is to evaluate the performance of the approach together with these tools on the Volvo test case. In this section, a description of the tools is given as well as a description of the tool-chain to show how the tools work together to realize the developed formalization approach.

The following tools were developed;

- An extension of the 3D modeling tool taraVRbuilder, which exports the plant structure in an AutomationML format.

- A tool for requirements specification for technical equipment.
  The tool uses pre-defined language elements to construct specification statements. Those language elements are e.g. verbs, nouns, attributes and other elements of written/spoken language. The tool is called Boilerplate based Natural Language Requirements Tool (BRT).

- A tool for analyzing the feasibility of resource and process structures in order to produce well specified product variants. The tool is called PPR AnalyZ3r.

- A tool to specify time related project information in order to design, manufacture and install production equipment. The tool is called Milestone Planning Tool.

- **taraVRbuilder for Resource Planning**
  taraVRbuilder is a rapid prototyping software for generating layouts of virtual dynamic production and logistics systems. It can be used for planning purposes in context of Digital Factory and Industry 4.0 scenarios. It provides a comprehensive library of more than 500 object types, which can be used in a drag and drop manner for the planning of factories and production systems [19]. The software can also be used to define products associated with the production system. Figure 3.11 below shows a screenshot of the taraVRbuilder user interface.

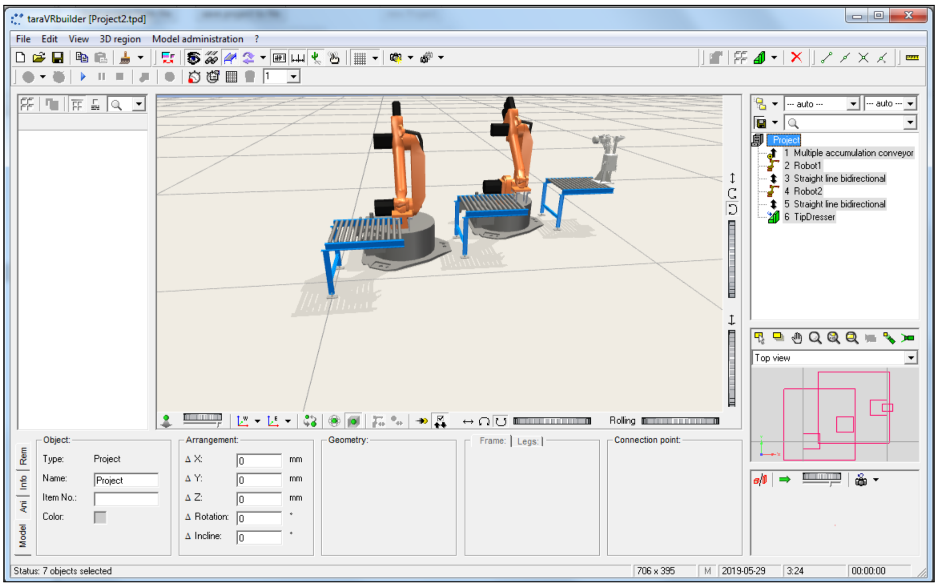


Figure 3.11: User interface of the taraVRbuilder tool.

Within the ENTOC formalization approach, the taraVRbuilder is used to design the basic structure of the production system. It is also used to assign requirements pertaining to the resources in the production system.

Section 3.3.1 described the AutomationML format which is the data exchange format chosen for the ENTOC formalization approach. The taraVRbuilder includes the functionality of importing an AutomationML RoleClassLibrary. The requirements on the resources are included as attributes in the role class library and then imported in the taraVRbuilder after which they are assigned to the resources in the model.

The software then exports the production system structure together with the requirements into an AutomationML file. The export includes two Instance-Hierachies and these are the ProductStructure as well as the ResourceStructure with the resources as InternalElements together with the assigned requirements as attributes. An example of the exported AutomationML structure is shown in figure 3.12 below,

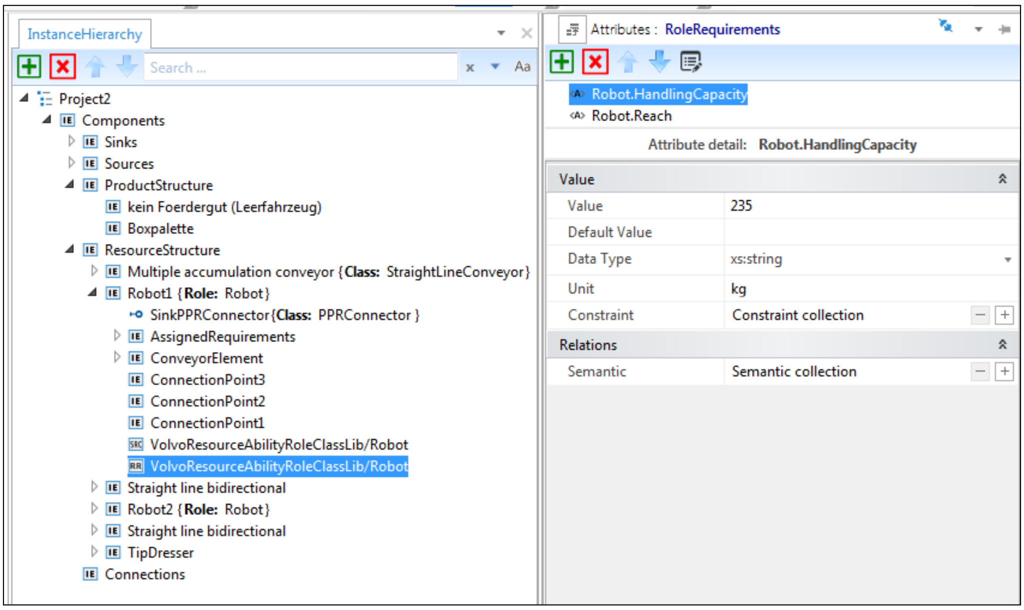


Figure 3.12: Example of the AutomationML structure exported from the taraVR-builder tool.

An outline of how the tool is applied to aid the formalization approach is described below:

- The production system designer defines the AutomationML RoleClassLibrary with resource requirements.

- The designer uses the taraVRbuilder to design the structure of the production system by aligning the resources as per the layout. In this step the designer can use the standard objects in the taraVRbuilder library or can import CAD models for the resources not given in the library.

- The designer then imports the RoleClassLibrary from the first step and assigns the requirements to the particular resources for which they are specified.

- The designer then exports the AutomationML file.

A further description of these steps is given in the methodology section where the approach is applied on the Volvo test case.

The exported AutomationML file is then used as an input to the next tool in the tool-chain that is the Boilerplate based Natural Language tool. This tool is described below.

■ **Boilerplate based Natural Language Requirements Tool for Requirements Specification**
Boilerplate based Natural Language Requirements Tool is a tool developed by TWT Gmbh and was developed around the Boilerplate standard described in section 3.3.1. The overall purpose of this tool is to allow the user to formalize the requirements but in a manner that they are still human readable so that the requirements can be reasoned about. The requirements are stored in AutomationML format. The input of this tool is an AutomationML file which in regards to the tool-chain is the export of the taraVRbuiler tool described above.
The tool supports the ENTOC formalization approach through the following functionlity;

- The tool reads the ProductStructure and ResourceStructure InstanceHierarchies from the AutomationML file input.

- The production system designer uses the tool to add the ProcessStructure InstanceHierarchy and then adds the processes as per the process description. The tool adds the processes as internalElements to the AutomationML file as well as the process requirements. These requirements are added to the process internalElements as attributes by the tool.

- The tool is used to generate the internal links between the products, resources and processes in order to reflect the PPR model of the production system as well as the sequence of the processes.

- The tool imports the BoilerplateSystemUnitLibrabry developed for the ENTOC project containing the Boilerplates templates.

- The tool enables the user to formulate the requirements using the boilerplate templates hence formalizing them.

- The tool also displays the Instance Hierarchies as well as the added requirements on the tool Interface.

- The tool exports an AutomationML file with all the InstanceHierarchies, Internal PPR links and process sequence links as well as the formalized requirements formulated from the Boilerplates.

The figure 3.13 below shows the tool's user interface.

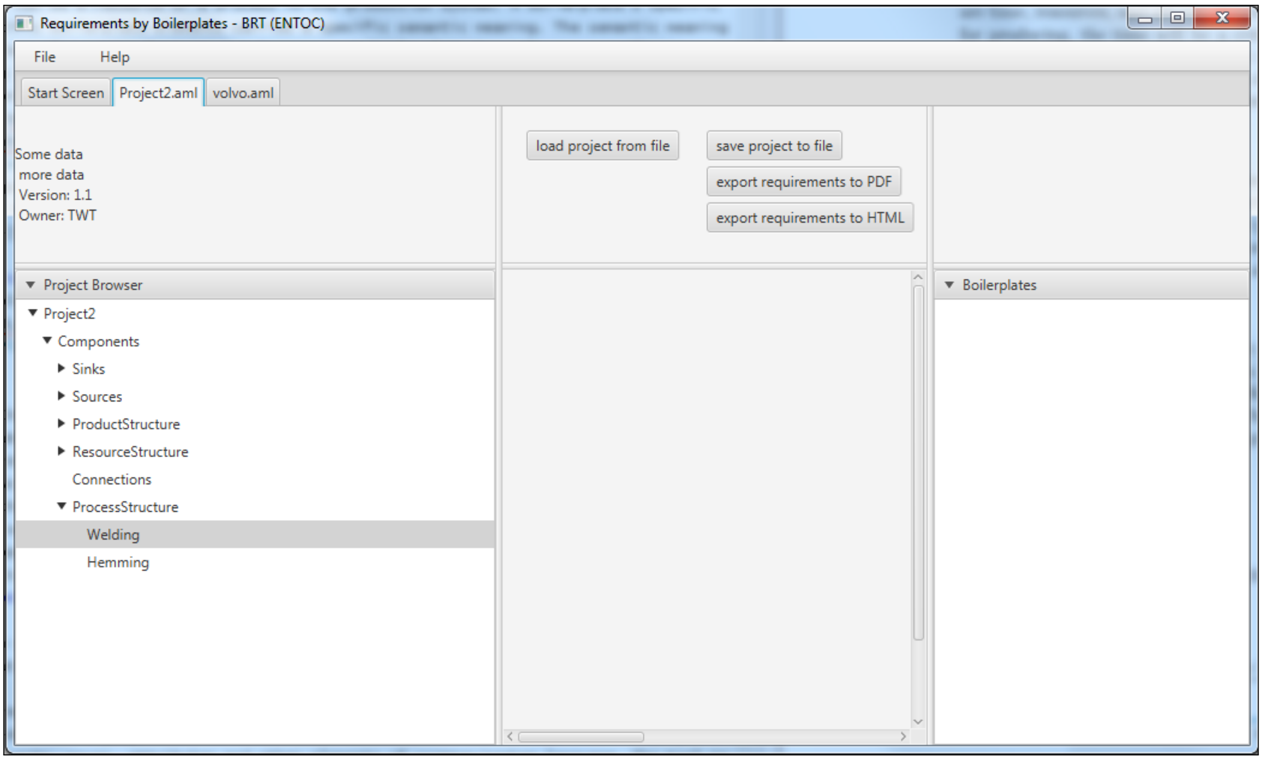


Figure 3.13: User interface of the Boilerplate tool.

The export from this tool is then used as a input for the next tool in the chain. This tool is described below;

- **PPR_AnalyZ3r for Product Variability Handling**
  The PPR_AnalyZ3r tool developed by Chalmers University which is one of the partners in the ENTOC project. It is a testing and optimizing tool that can read an AutomationML file then check the relations between the product, process, and resources to detect any conflicts whilst checking the feasibility of the PPR model. The relations are evaluated according to three factors as shown below.
  1. The logical aspect
     In the optimization part changing the processes sequence needed to optimize the productivity and checking the logical relations between the process, resource, and product gives feedback if the change is possible. Besides, each product variant needs specific items for example product A implies items a_1,and a_2. Hence the tool checks the relations between the product variants and the items needed to produce those variants.

2. The scheduling constraints
   The AutomationML file as input for this tool contains many restrictions and the scheduling constraints are part of these restrictions. The tool optimizes the scheduling and minimizes the process duration as much as possible according to these constraints and gives more information about the bottlenecks.

3. The resource requirements
   The resources have some attributes that describe the functionality and the abilities for each resource. The tool checks the attributes and the required resources for each process then work on matching the available resources with the required process, aiming to minimize the cost function for a certain process.

Currently, the analyzing tool does not have GUI, so the Boilerplate tool can call the analyzer and feed it with AutomationML file after adding the requirements and specifications. When the analysis is finished, the tool will print all the information regarding the optimized process as well as the list of product variants/items and, the list of processes as shown in figure 3.14 .

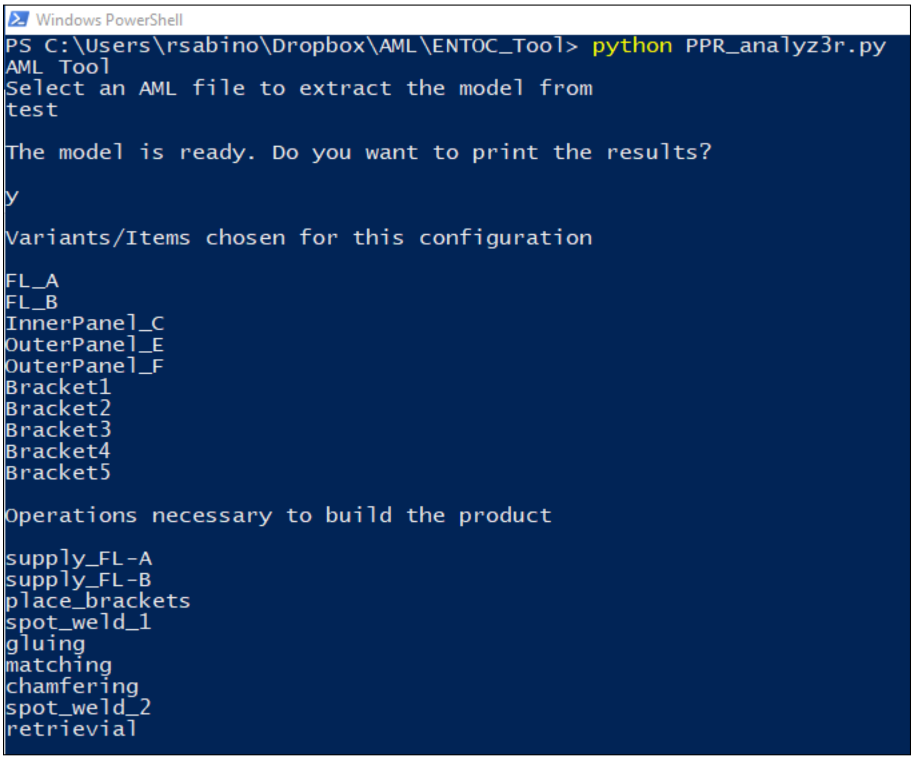


Figure 3.14: Tool output showing the variants to be produced and the corresponding processes required.

As mentioned previously, matching the resource with the process is part of the optimization. So the tool matches the available resources with the necessary processes needed. After that, the tool decides which resources are eligible for the task and then selects the ones that optimize the cost function. It will then display the process sequence with the scheduling times and the processes list with the matched resources. In addition, it offers the option to print a Gantt chart with the process sequence as shown in figures 3.15 and, 3.16.

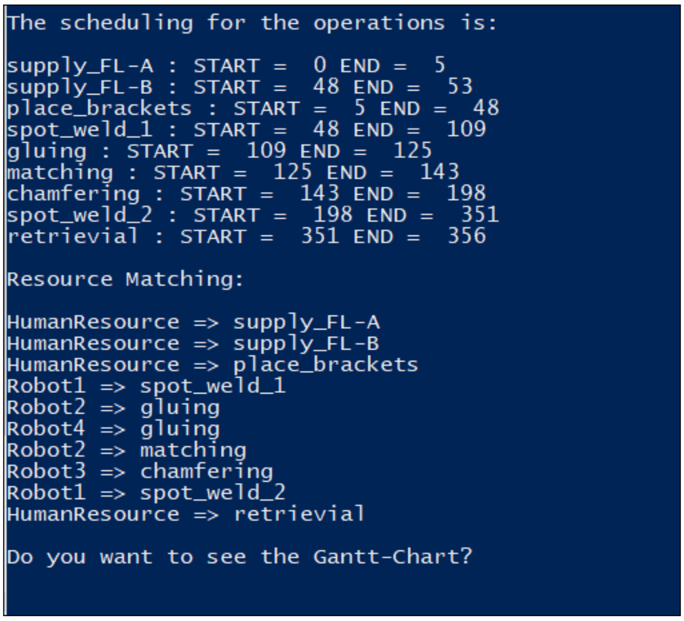


Figure 3.15: Tool output showing optimal scheduling and resource matching.

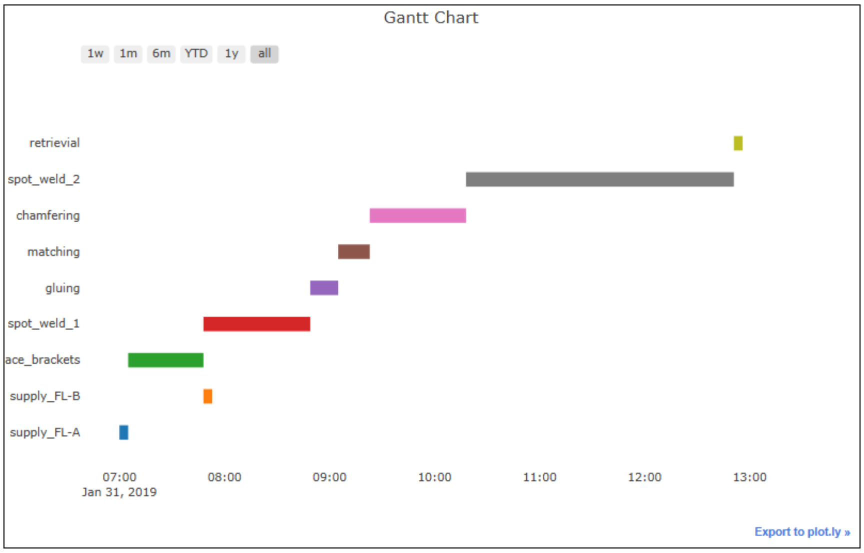


Figure 3.16: Scheduling Gantt Chart produced by the tool.

If the analyzer detects renders the solution unfeasible because of any of the constraints not adhered to, it will display a warning message and will point on the subset of constraints making the model unfeasible as shown in figure 3.17.

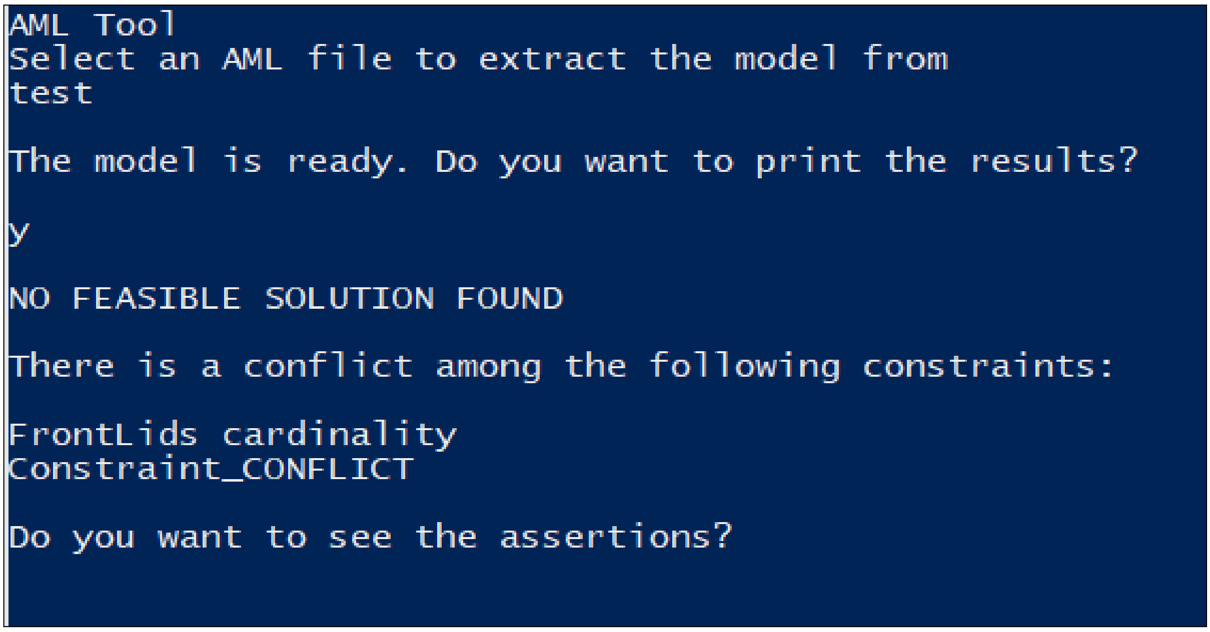


Figure 3.17: An example of a conflict identified due to in-feasibility.

- **Milestone Planning Tool for Project Planning**
  The Milestone Planning Tool enables creation of project time plans in the form of milestone plan and then attaches them to the AutomationML file. The critical dependencies in the engineering and commissioning of a production system are time, resources, and efforts. When the production system should be ready for producing, the time will be a critical non-functional requirement for the customer and the supplier. For that reason, both the customer and supplier should define a time points to keep checking the project progress, especially within large projects.Those time points are called milestones of the project.

  Thus a milestone plan combines delivery time points with objects to be delivered. In the early phase of the project, the delivery objects are related to the equipment but are not the equipment parts themselves. Those delivery objects are for example engineering data about the equipment (PLC or robot control programs, CAD data, etc.), but can also be manuals or other documents.

  The Milestone tool task is getting the AutomationML file as input then define the time points and add them as a milestone plan to the same file. This planning tool developed based on Web-technologies. In the left side of the start page shows a list of options as AutomationML folder,Projects,Milestone plans ,and Milestones as shown in figure 3.18.

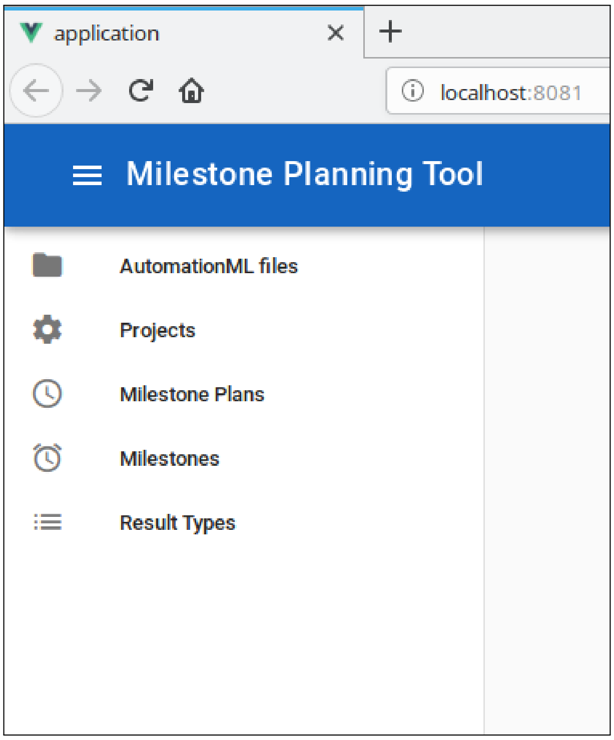


Figure 3.18: Start window of the Milestone Planning Tool.

The AutomationML folder in the left side of the start page contains all the AutomationML files uploaded to the tool and allows to upload or delete the files from the folder as shown in figure 3.19.

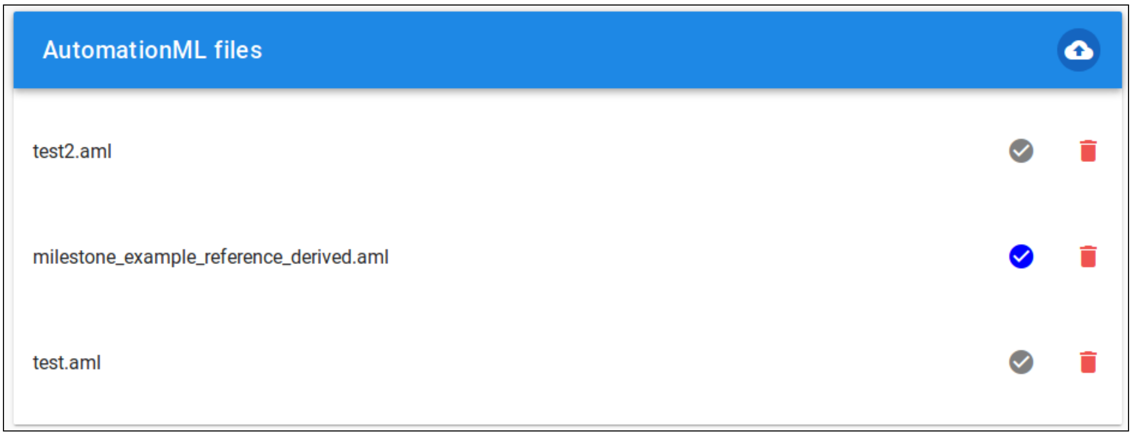


Figure 3.19: Selection of AutomationML files.

The Milestones option in the left side displays all the milestones times and allows to edit the milestone properties as illustrated in figure 3.20 .

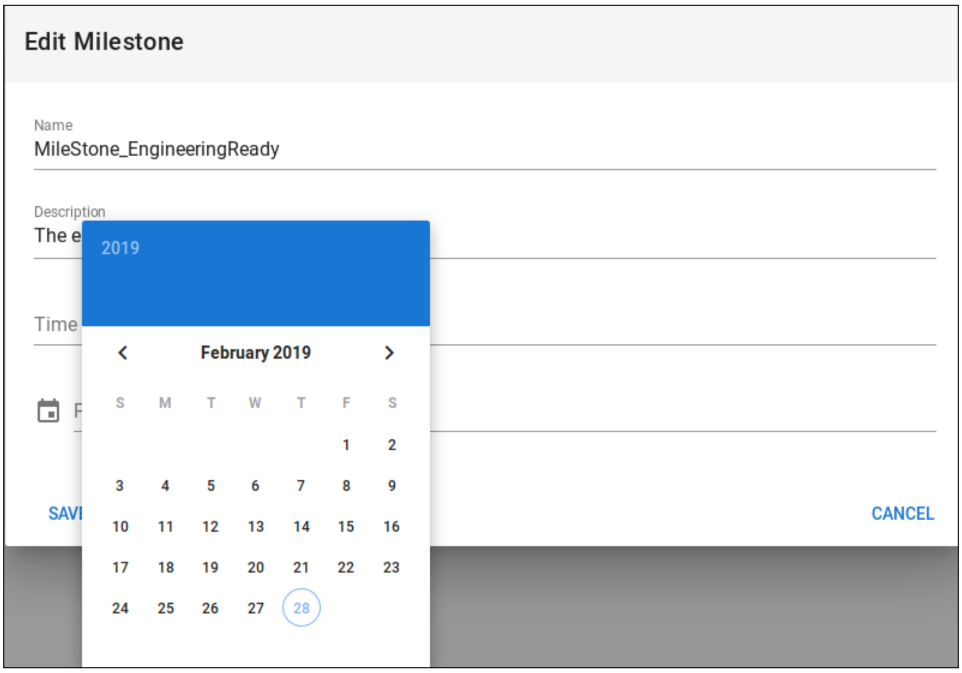


Figure 3.20: Editing of milestone properties in the Milestone Planning Tool.

3.3.4 ENTOC Formalization Approach Solutions to Existing Requirements Engineering Problems

In section 3.1.2, the problems associated with the current requirements engineering process due to lack of formalization were presented. In that section, the problems were outlined for each of the requirements classes that are being considered for the ENTOC Project. In this section, we describe how the ENTOC formalization approach tackles and solves some of these problems. In this section, the solutions are also outlined for each of the requirements classes being considered in the project.

- **Process description**
  The main problem with this class of requirements when not formalized was that the reader of the requirement specification had to read through a number of files to make sense of the process description due to the cross referencing between these files. With the ENTOC approach, this problem is counted by using a single AutomationML file to define the entire process description i.e. the process sequence, the required resources, the associated product for each process as well as the associated requirements for each process. With a single AutomationML file, the cross referencing between different file is eliminated. This helps save time for the vendor who the requirement specification is meant for and also eliminates the possibility of misinterpretation of the process description.

- **Layout planning**
  In this class of requirements, one of the problems that was identified was that

the difficulty in differentiating between a requirement and solution proposals in the same layout specification. The ENTOC approach addresses this issue by offering the ability to add extra information to the layout objects for example attributes can be used to attach requirements information to the resources in the layout.

The other issue was that layout specification always make references to other files for example the process description to give the reader a clear understanding of the layout and its relationship to the process description. This often is time consuming for the reader. With the ENTOC approach, this cross referencing is eliminated through the use of a single AutomationML file that includes all the necessary information.

- **Time schedule for commissioning**
  The main issue associated with lack of formalization for this class of requirements was the distribution information over a number of files anf file formats for example, the time schedules and plans are usually defined in Microsoft excel sheet while gate condition and definitions may be defined in a word document meaning the user has to go back and forth to understand the information regarding the timeplan. With the ENTOC approach, with the help of the Milestone planner tool, the time plans are put together with the rest of the specification in a single AutomationML file. This helps save time and its easier for the reader to understand the time plans.

- **General company constraints**
  This class if requirements as described in section 3.1.1 consists of generic company requirements pertaining to company standards. These are usually several documents with cross references in them. The same issue arises due to lack of formalization and this issue is the time consumption in reading through all these documents. The other issue however is also with this kind of data distribution, automated testing is rather difficult and in some cases even impossible. The ENTOC approach addresses these issues through the introduction of a standardized language in form of boilerplates which can be added to the AutomationML containing the rest of the requirements. This brings about the possibility for automated testing since the AutomationML is machine readable. Having most or all the company standard requirements in a single file also saves time in extracting this information.

4
Application of the ENTOC Formalization Approach to the Volvo Test Cases

Volvo as part of the ENTOC Project partners was chosen as a demonstrator for the ENTOC Formalization approach. Volvo designed two test cases, and in this chapter, we describe the steps taken in this thesis work to apply the ENTOC approach in the formalization of the requirement specification for these test cases.

4.1 Volvo Test Case One - Front Lid Cell

This Volvo test case is based on existing production cell which is a robotic production cell for the P2545 Front lid for one of the Volvo truck variants. The production cell is an assembly cell where the P2545 Front Lid is assembled by merging together the Inner Panel and Outer Panels together with 5 Brackets. It consists of three robots, three fixtures, a hemming bed and a gluing station. The assembly consists of mainly three processes and these are welding, gluing and hemming.
Figure 4.1 below shows a screen capture of the production cell simulation model.

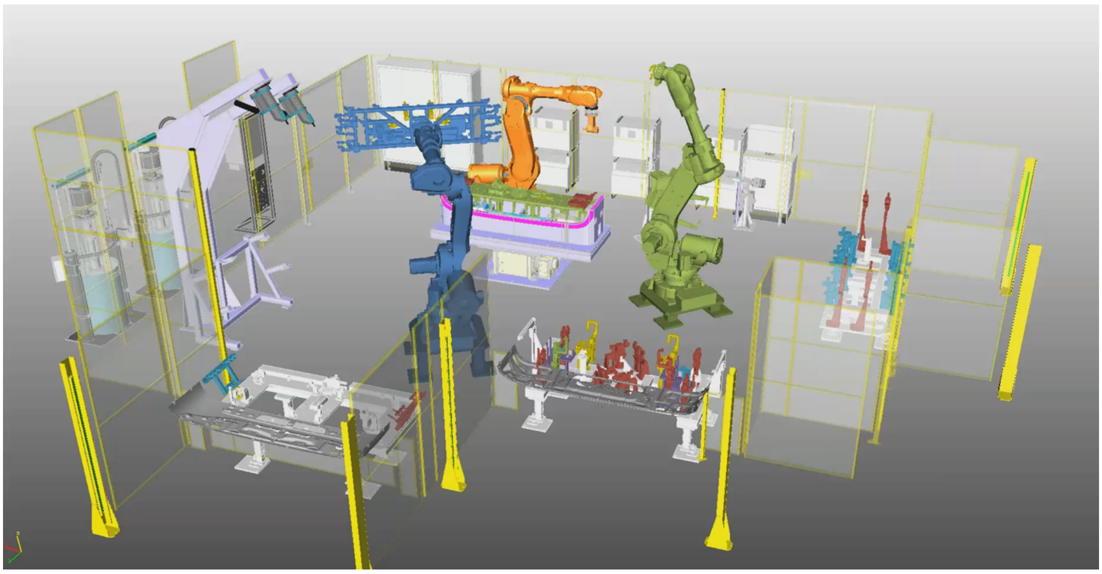


Figure 4.1: Simulation model of the production cell used for the Volvo test case.

The following steps were taken in applying the ENTOC approach with the help of the developed tool-chain to formalize the requirement specification of the test case. They are described further in the next sections.

- Analyzing the requirement specification of the production cell.
  The Requirements specification for this particular cell was provided in the form of Request for Quotation. This analysis was done to determine what sections of the requirement specification can be formalized and which ones cannot be formalized using the ENTOC approach and tools.

- Applying the ENTOC approach using the tools to formalize those sections of the specification which can be formalized as determined from the previous step.

4.1.1 Analysis of the Request For Quotation (RFQ)

This production cell is already operational hence its Request for Quotation(RFQ) is available. The RFQ is a complete set of documents that will be completed when a contractor has been selected and the contract signed. It also includes documentation about changes agreed upon during the contract implementation.

For this thesis work, the RFQ was treated as the unformalized requirement specification to be formalized using the ENTOC approach with the help of the tools.
Figure 4.2 below shows a screen capture of the RFQ folder for the Production cell.

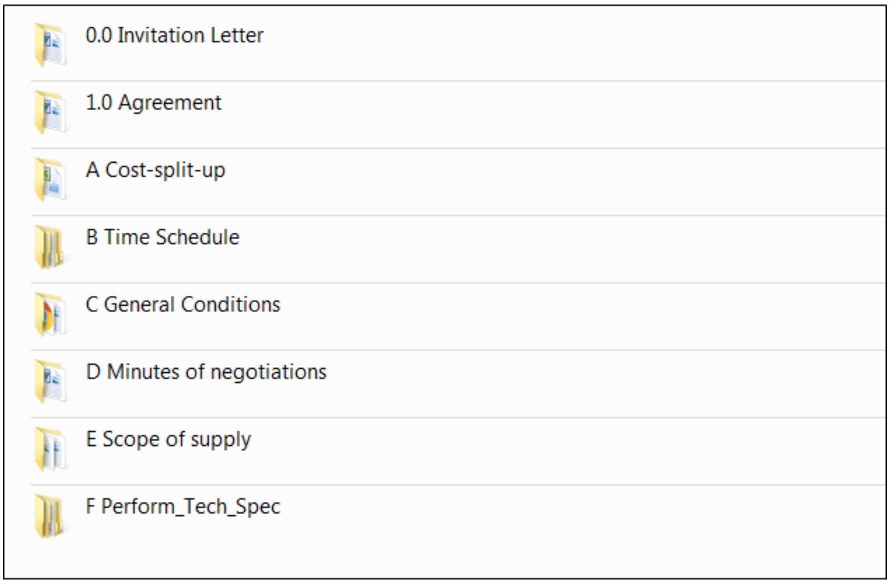


Figure 4.2: A screen capture of the production cell RFQ Folder.

As can be seen in figure 4.2 the RFQ contains a number of folders each folder containing a set of documents with different information regarding the design and requirements of the production cell.
In this step of the thesis work, we analyzed the information contained in each of

these sets of documents to determine which pieces of information can be formalized using the ENTOC approach.

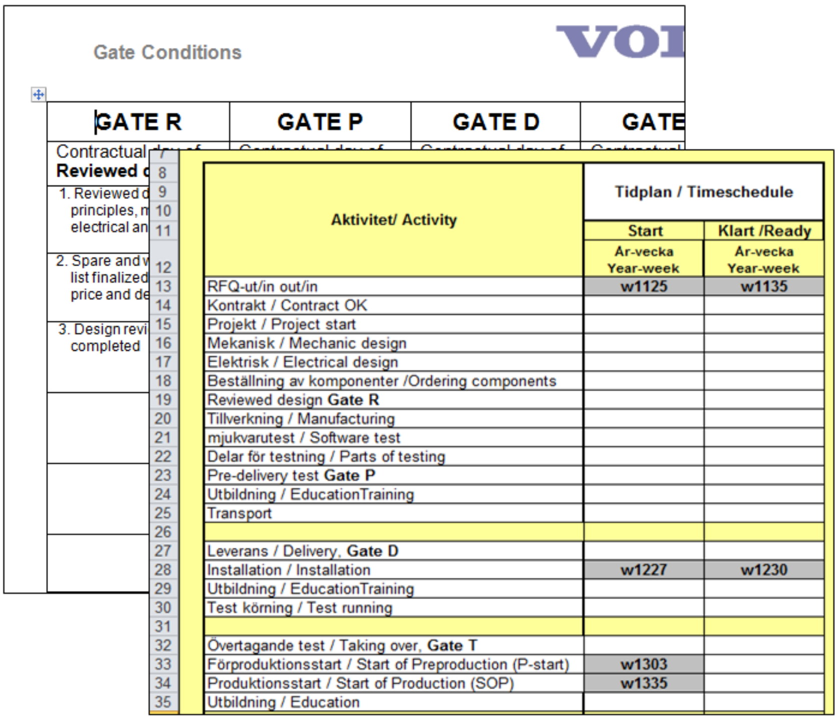
Below we describe the information contained in each set of documents and also give a conclusion on the result of our analysis on as to whether this information can be formalized using the ENTOC approach or not. For cases where the information can be formalized, we describe with what tool in the ENTOC tool-chain this information can be formalized. We also mention the format of the files in these sets of documents to highlight the diversity of the file formats that are used when the requirement specification is not formalized.

- **Invitation Letter**
  This document contains an invitation to submit an offer. It is in Microsoft Word format.
  From the analysis, this information is not necessarily a requirement and hence could not be formalized using the ENTOC approach.

- **Agreement**
  This document is an agreement to be signed between Volvo which is the customer and the chosen vendor or contractor. It is also in miscrosoft word format.
  From the analysis, it was concluded that this information is not a requirement and hence cannot be formalized using the ENTOC approach.

- **Cost Split up**
  The cost split is a Microsoft Excel sheet with a list of resources to be included in the purchase and the corresponding prices.
  The conclusion from the analysis was that this information gives some details on the production cell equipment but not enough information to derive any requirements from it. Thus, it cannot be formalized using the ENTOC approach.

- **Time Schedule**
  This set of documents contains a word document describing the Gate conditions in regards to the production cell installation and use. It also contains an excel sheet with the time plan of the production cell installation and operation described in terms of the Gate conditions.
  Figure 4.3 below shows a screenshot of some of the files inside the time plan specification.



	Aktivitet/ Activity	Tidplan / Timeschedule	
		Start	Klart /Ready
		Ar-vecka Year-week	Ar-vecka Year-week
13	RFQ-ut/in out/in	w1125	w1135
14	Kontrakt / Contract OK		
15	Projekt / Project start		
16	Mekanisk / Mechanic design		
17	Elektrisk / Electrical design		
18	Beställning av komponenter /Ordering components		
19	Reviewed design **Gate R**		
20	Tillverkning / Manufacturing		
21	mjukvarutest / Software test		
22	Delar för testning / Parts of testing		
23	Pre-delivery test **Gate P**		
24	Utbildning / EducationTraining		
25	Transport		
26			
27	Leverans / Delivery, **Gate D**		
28	Installation / Installation	w1227	w1230
29	Utbildning / EducationTraining		
30	Test körning / Test running		
31			
32	Övertagande test / Taking over, **Gate T**		
33	Förproduktionsstart / Start of Preproduction (P-start)	w1303	
34	Produktionsstart / Start of Production (SOP)	w1335	
35	Utbildning / Education		

Figure 4.3: A screen capture of the files inside the Time plan Specification.

The analysis of this information shows that the information/requirements in the time plan can be formalized using the ENTOC approach with the aid of the MileStonePlanner Tool.
The activities described in these documents are treated as milestones and using the tool are assigned to the different resources they correspond to as shown in section 3.3.3.

- **General Conditions**
  This set of documents consists of contractual agreements between Volvo and the solution suppliers and does not contain any requirement specifications in relation to the production cell.
  The analysis of this information concluded that it can be added to the AutomationML file using the boilerplate tool with the natural language boilerplates. However since no requirements are present, it is ignored.

- **Scope of Supply**
  The information contained in this set of documents gives an overview of the planned system, overall targets/performance, obligations to follow in design and installations, cross reference to all important documents and some practical things for changes and responsibilities.
  Many requirements are outlined in these documents and from the analysis of these documents, most of these requirements can be formalized with the EN-

TOC approach using the Boilerplate Tool.
These requirements are in a natural language format and hence using the Boilerplate tool can be structured into various boiler plates and then formalized.
Figure 4.4 below shows a screenshot of some of the files inside the Scope of supply specification.

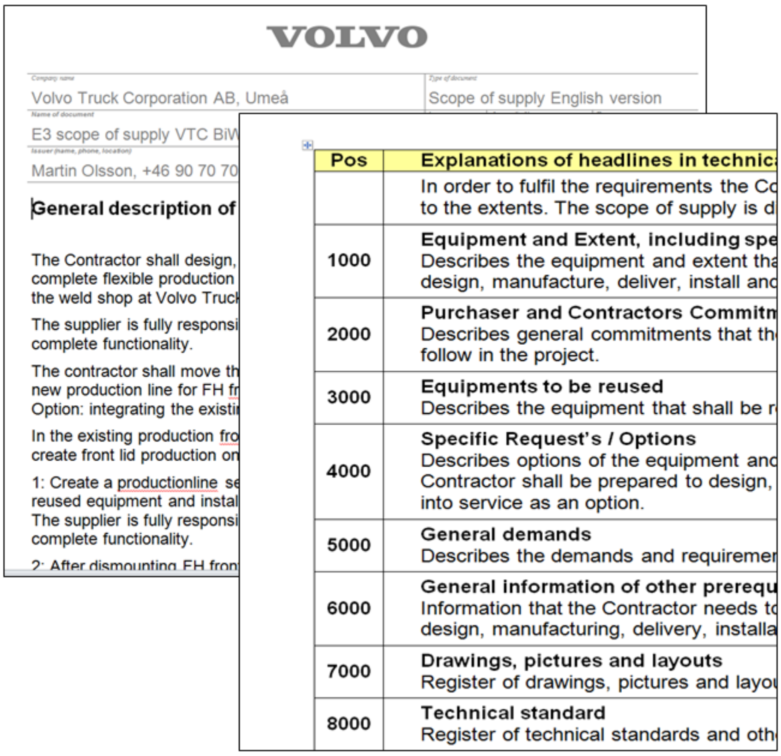


Figure 4.4: A screen capture of the files inside the Scope of Supply Specification.

- ■ **Performance and Technical Specifications**
  This set of documents contains case specific requirements that describe what is to be produced, how, where, how much, as well as the quality. Models of the equipment and components are also be specified. These requirement specification documents are divided into the following;

  - ○ **Layout/Resources**
    In this set of documents, different file formats are used and these are; An image is used to show the cell layout with the resources used for production. CAD drawings are also provided to give a more detailed description of the layout with the geometry of the cell included.
    From the analysis of these documents, conclusion was made that these requirements can be formalized using the ENTOC approach with the help of the TaraVRbuilder tool.
    Figure 4.5 below shows a screenshot of some of the files inside the Layout

specification.

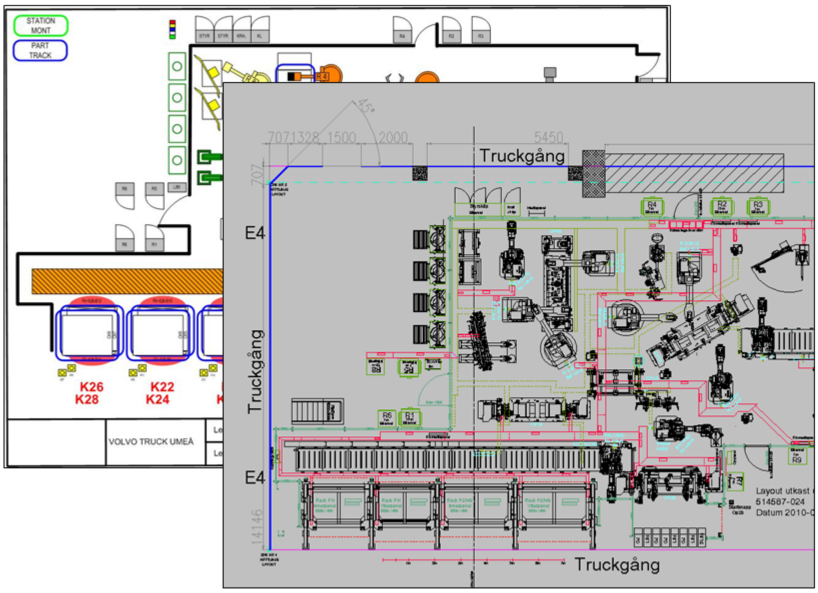


Figure 4.5: A screen capture of the sample documents inside the Layout Specification.

- **Product Description**
  This set of documents describes the product to be produced in detail plus the item/components used to assemble the tool. In this specification, different formats are also used for example, CAD models are used to describe the product and word files are used to give a description of the components used to assemble the product.
  These specifications were analyzed and the conclusion was that they could be formalized using the ENTOC approach by using the taraVRbuilder as shown in section 4.1.2.1.

- **Process Description**
  This set of documents specifies the sequence of procedures that are taken for the assembly of the product. It also details the resources used for each process.
  The specification is in different file formats for example word documents are used to describe the processes and image files show the location in the cell where these processes are carried out. Powerpoint presentations are used to describe the process sequence and also show the flow of the product through the cell.
  On analyzing this specification, it was concluded that it can be formalized using the ENTOC approach. To facilitate this formalization, the Boilerplate tool will be used as shown in section 4.1.2.2.

Figure 4.6 below shows a screenshot of some of the files inside the Process Description specification.

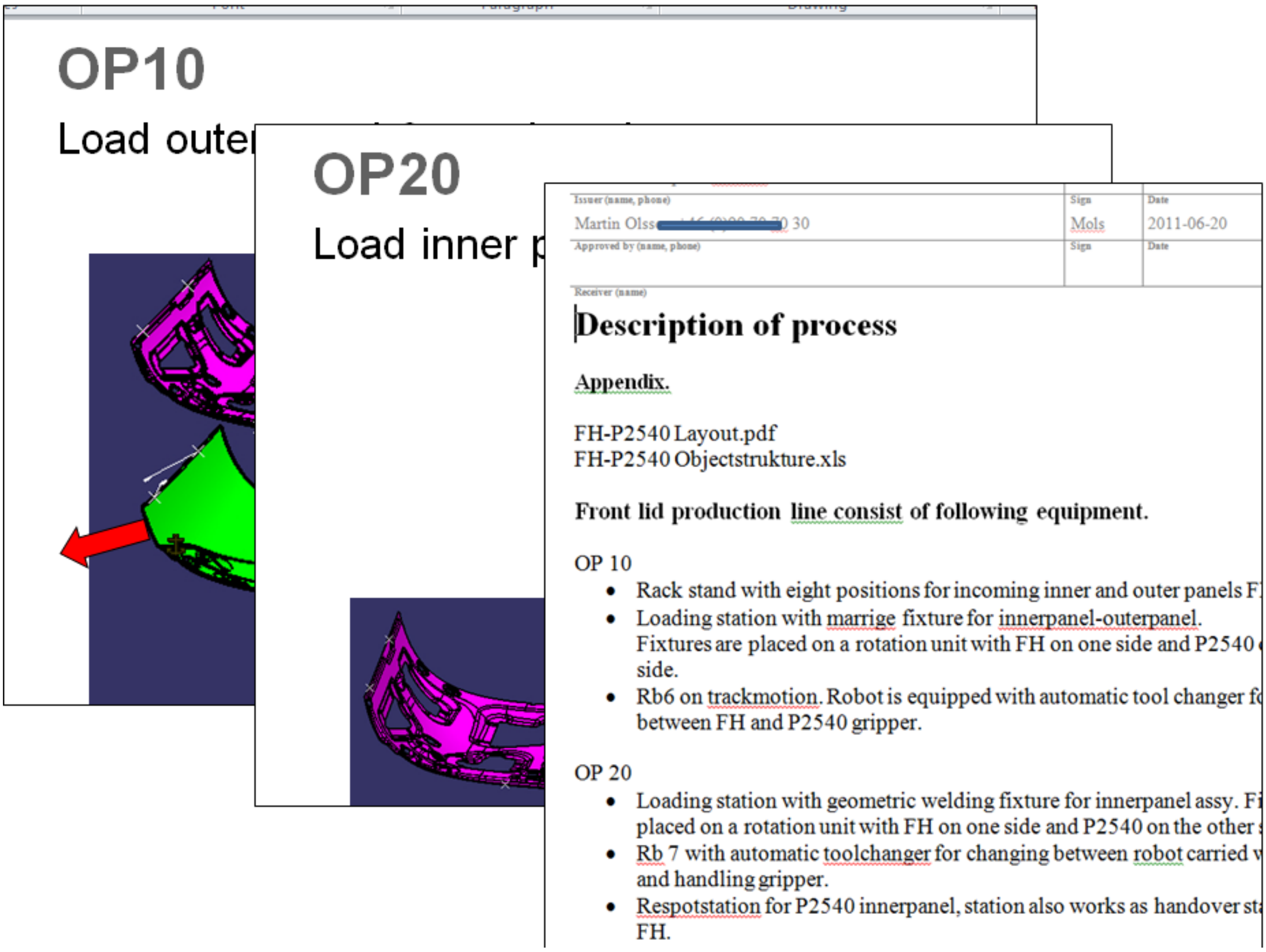


Figure 4.6: A screen capture of the files inside the Process Description Specification.

4.1.2 Formalization of the Test Case's Requirement Specification using the ENTOC Approach and Tool-Chain

From the analysis, it was identified that some of the requirements could be formalized using the ENTOC approach and tools, in this section a description of how these requirements were formalized is presented and how the tools were used to aid in the formalization is also explained. An overview of the workflow deployed for the formalization is given in figure 4.7 below.

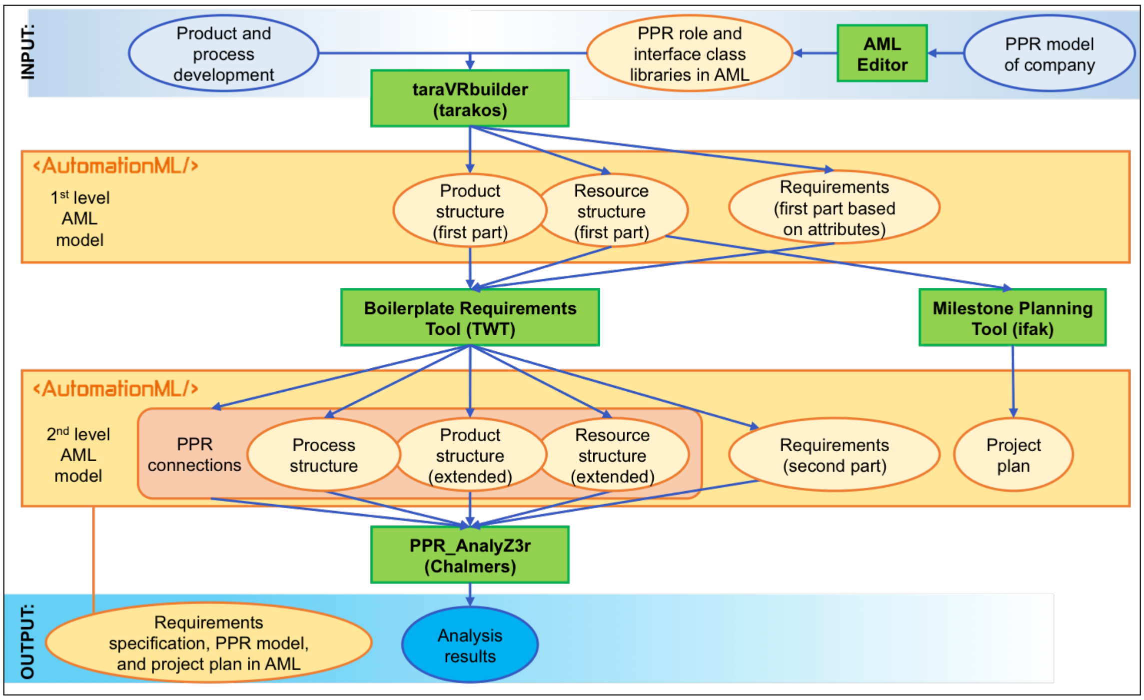


Figure 4.7: Workflow deployed in the formalization of the requirements.

The steps taken in the formalization are summarized in the figure 4.8 below.

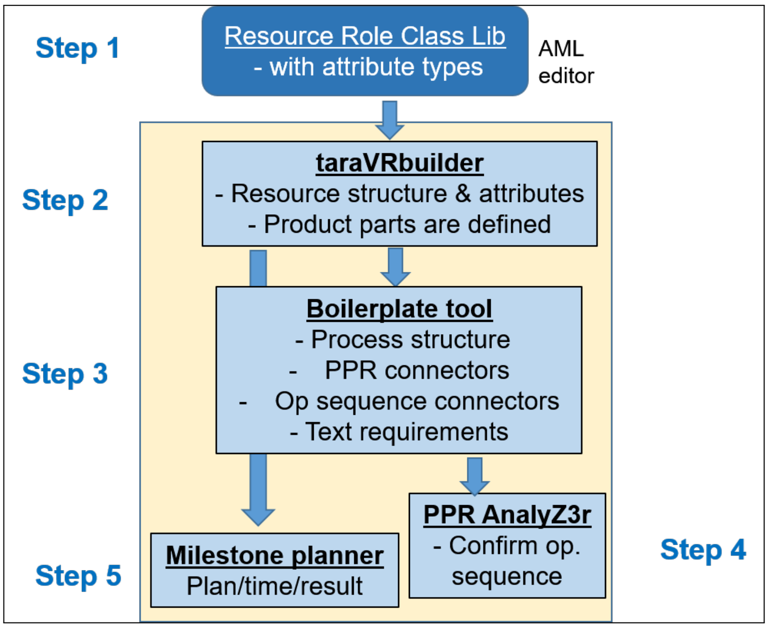


Figure 4.8: Steps taken in the formalization of the requirements in the RFQ.

4.1.2.1 Formalization of the Layout and Product Specification

Figure 4.11 below shows the design layout of the production cell. It shows the resources used in production and also the product parts going into the production cell and at what locations they are fed into the production cell.

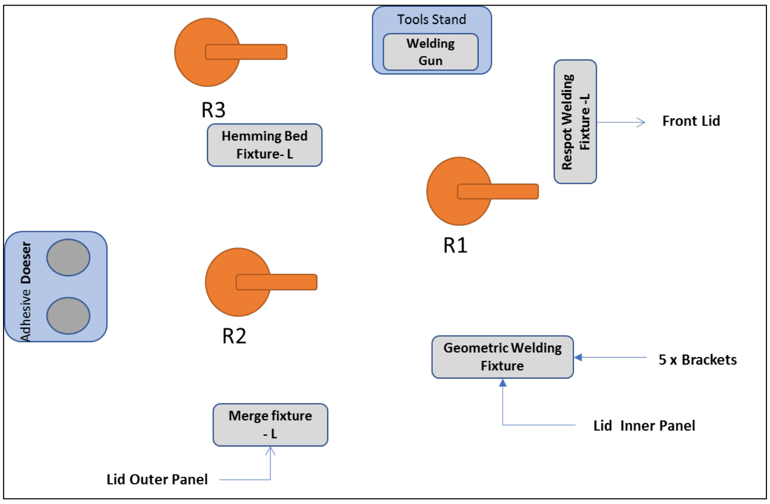


Figure 4.9: Design layout of the production cell.

To formalize the layout specification, the taraVRbuilder tool was used. Firstly, a RoleClass library called the VolvoResourceAbilityLib was designed in AutomationML and the requirements on the different resources in the layout were added as attributes. Figure 4.10 below shows the RoleClass library as it was defined in AutomationML. In this figure, the requirements on the Adhesive doser system are shown. Figures showing the abilities of the other resources are shown in the appendix.

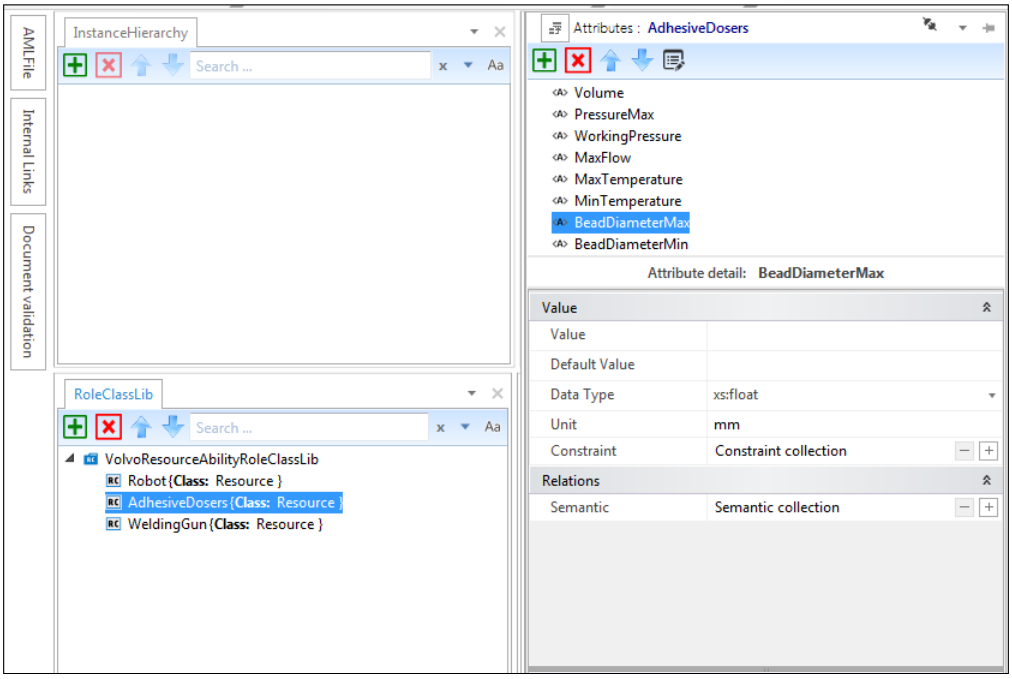


Figure 4.10: Role class library defining the requirements on the resources used in the cell.

The next step was to design the layout in taraVRbuilder. For this step, CAD models were imported from the Volvo CAD library into the taraVRBuilder project and the layout was defined as shown below.The product parts used for the assembly were also defined.

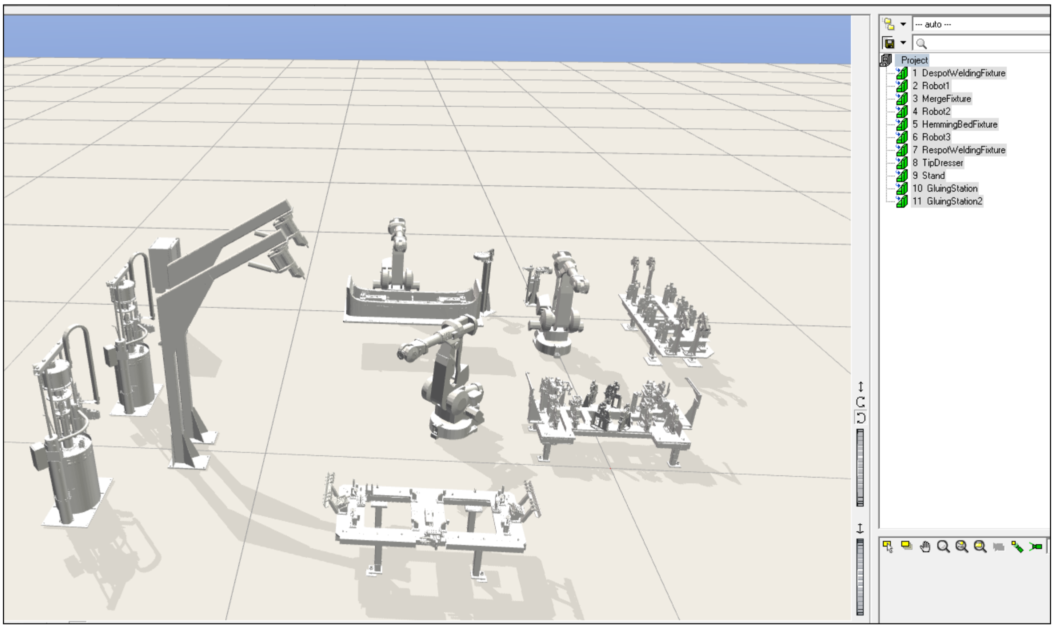


Figure 4.11: Design layout of the production cell in taraVRBuilder showing the resources in the production cell.

The Roleclass library was then imported into taraVRbuilder and the requirements on the resources were defined for each model as shown in the figure below;

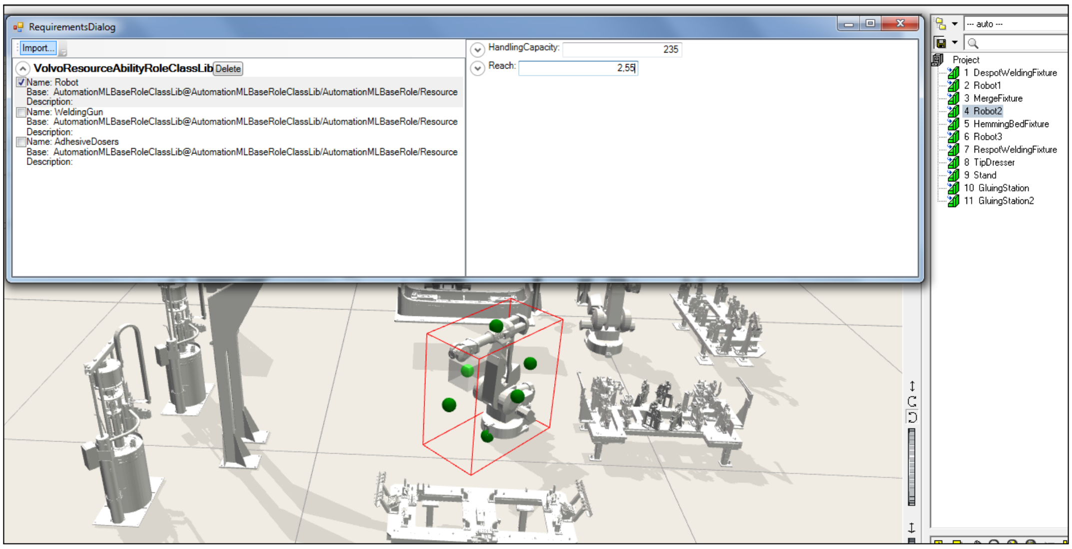


Figure 4.12: Addition of the resource requirements using the taraVRbuilder tool.

An AutomationML file was then exported. In the AutomationML file, the Product-Structure and the Resource structure are defined and for each of the resources, the associated requirements are aslo defined. The exported AutomationML file is shown in figure 4.13 below.

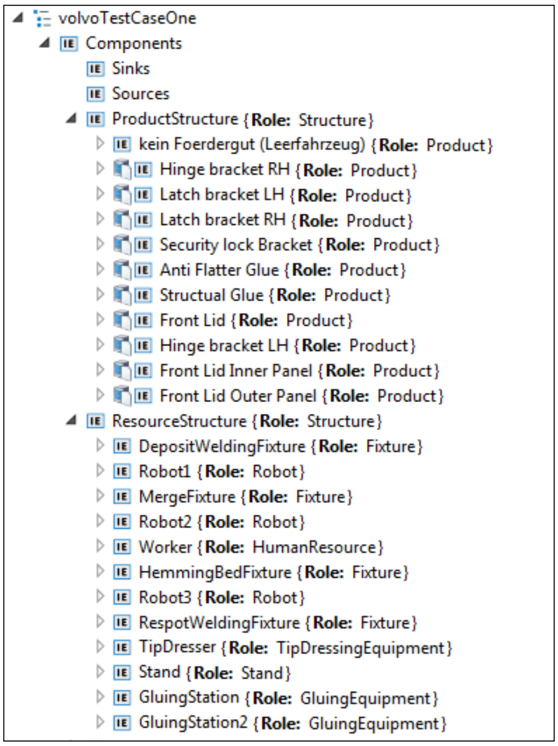


Figure 4.13: AutomationML file exported from the taraVRbuilder showing the Product and Resource structures.

4.1.2.2 Formalization of the Process Specification

Figure 4.14 shows a summary of the process description of the production cell. For each process, the resources required for the process as well as the part of the product being processed is shown.

To formalize the process description requirements as well as to show the PPR model of the production cell, the Boilerplate tool was used. The AutomationML file exported from the taraVRbuilder was imported into the boilerplate tool.
The ProcessStructure instanceHierachy was then added using the tool and the processes defined below the process structure. For each of the processes, attributes were defined to attach the requirements specified in the specification as shown in figure 4.16.
Order connector Interfaces i.e ProcessPreceedsProcess and ProcessFollowsProcess were then added and connected using internalLinks to show the sequence of the processes.
PPR connector interfaces i.e ProcessUsesResource, ProcessUsesProduct, ResourceUsedbyProcess and ProductUsedByProcess were then added. InternalLinks were then used to connect these interfaces to complete the PPR Model of the cell.This is done using the tool as shown in figure 4.17.

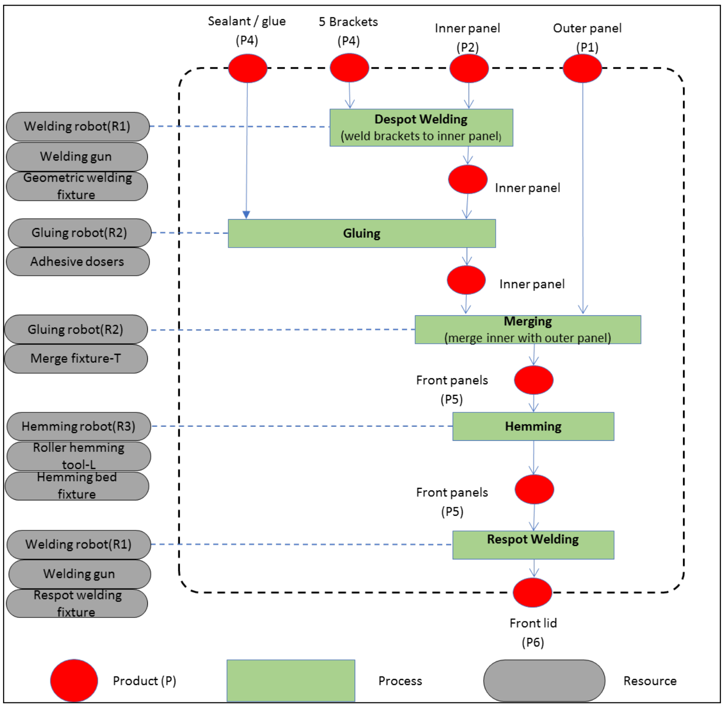


Figure 4.14: Process Description for the Front lid cell production.

Figure 4.15 below shows the process of adding the ProcessStructure instance hierarchy together with the order and PPR interfaces for the sequencing and PPR modelling respectively using the Boilerplate Tool.

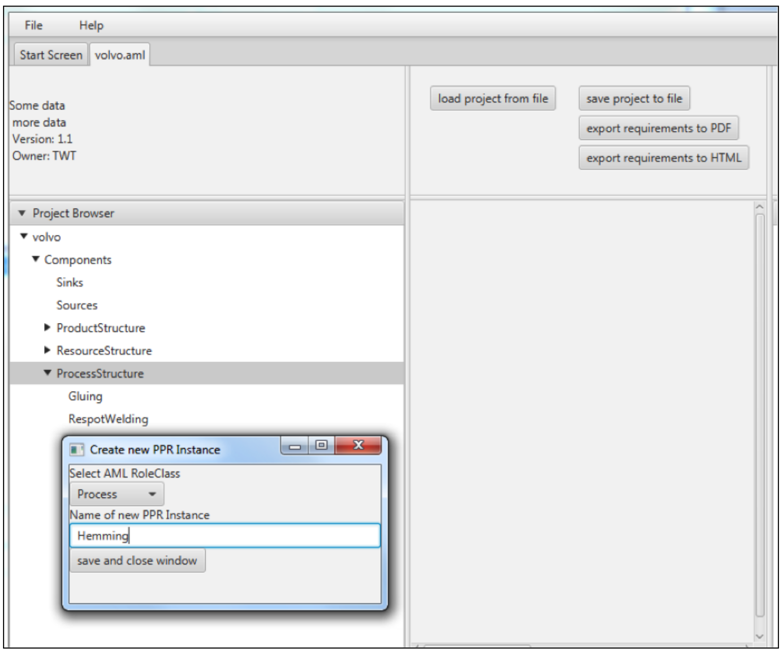


Figure 4.15: Addition of the Process structure and the specified process requirements using the boilerplate tool.

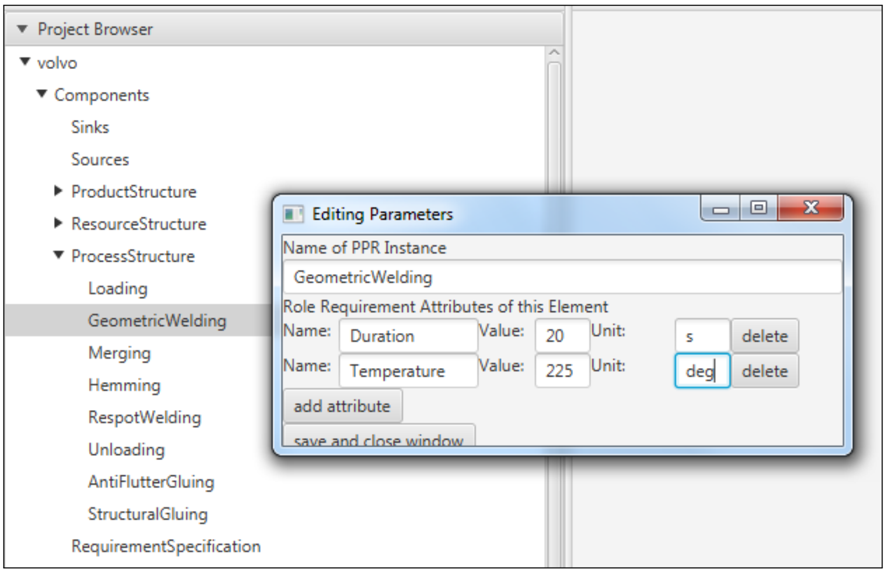


Figure 4.16: Addition of the process requirements as attributes of the process.

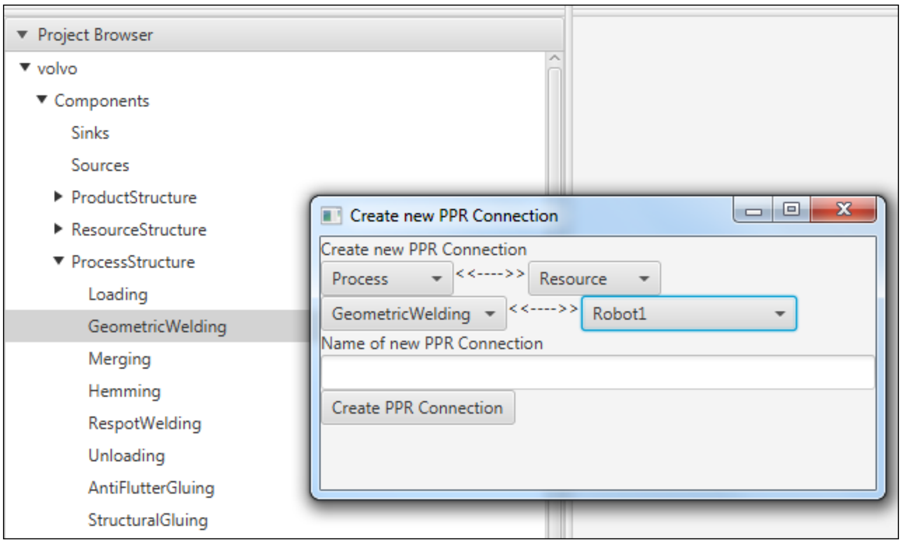


Figure 4.17: Addition of the PPR and order connectors for PPR modelling and process sequencing.

After defining the process description i.e the process sequencing and the process requirements together with the PPR definition, the AutomationML file was exported and is shown below in figure 4.18.
In the figure, the left hand side shows the process sequencing using the order interfaces i.e. ProcessFollowsProcess and ProcessPreceedsProcess linked with internalLinks to show the precedence.The right hand side shows the PPR connectors and internal links to show the PPR modelling of this test case.

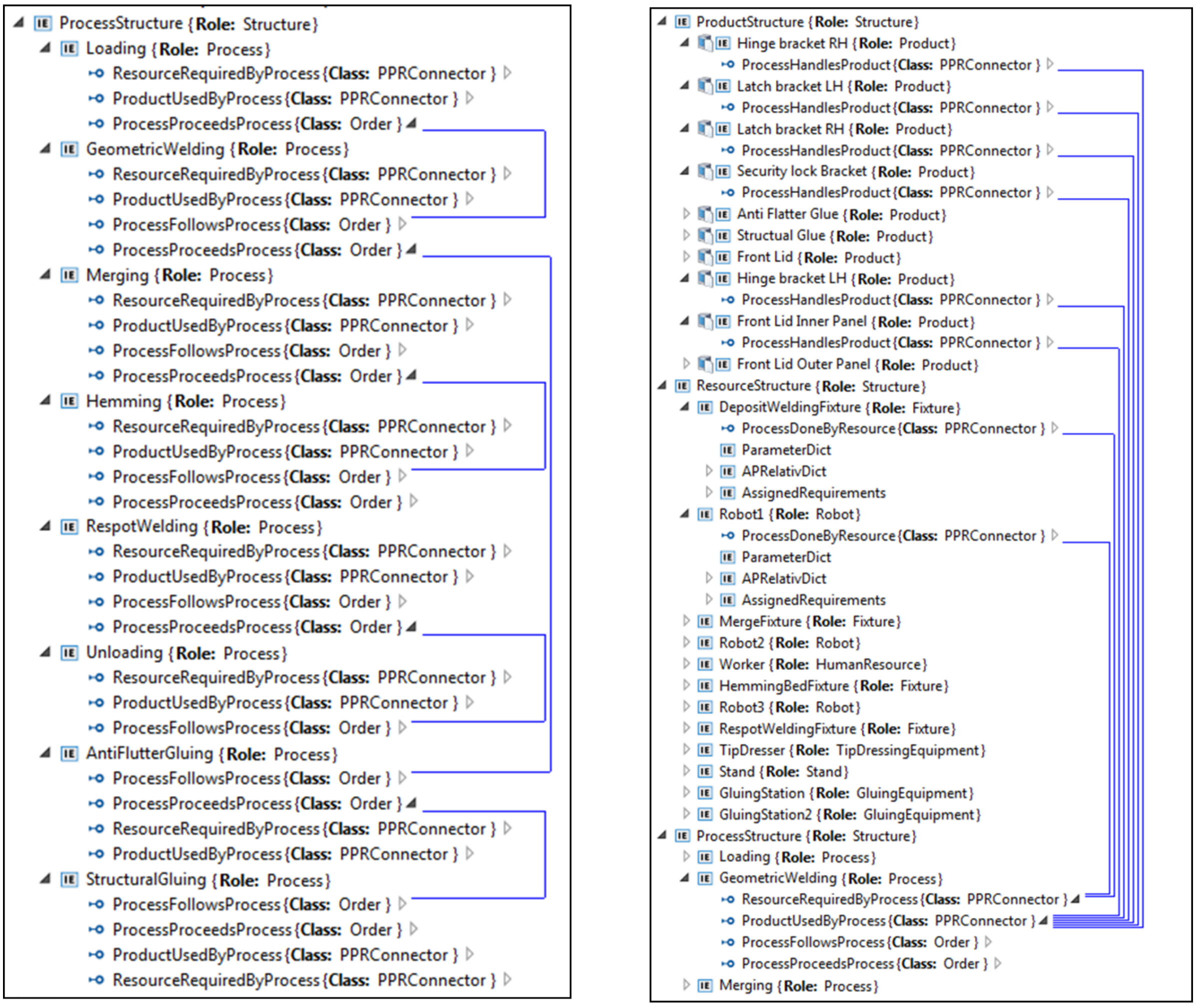


Figure 4.18: AutomationML file exported from the Boilerplate tool showing the added ProcessStructure and the PPR modelling of the production cell.

4.1.2.3 Formalization of the Scope of Supply and Volvo Standards Specification

From analysis of the RFQ, it was concluded that most of the requirements in the Scope of supply were non functional requirements expressed in natural language form. Boiler plates were generated using the Boilerplate system unit library to express these requirements in a formalized form. Figure 4.19 below shows the Boilerplate system unit library used to generate the boilerplates used for this formalization. Figure 4.20 shows addition of the requirements using the Boilerplates.

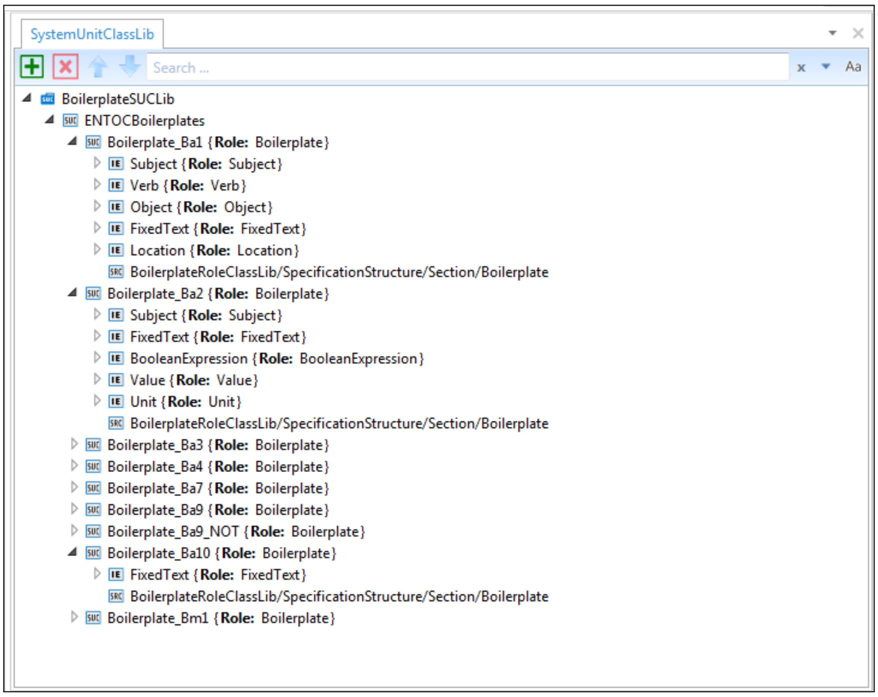


Figure 4.19: Boilerplate system unit library used to generate the boiler plates for formalization of the non functional requirements in the scope of supply.

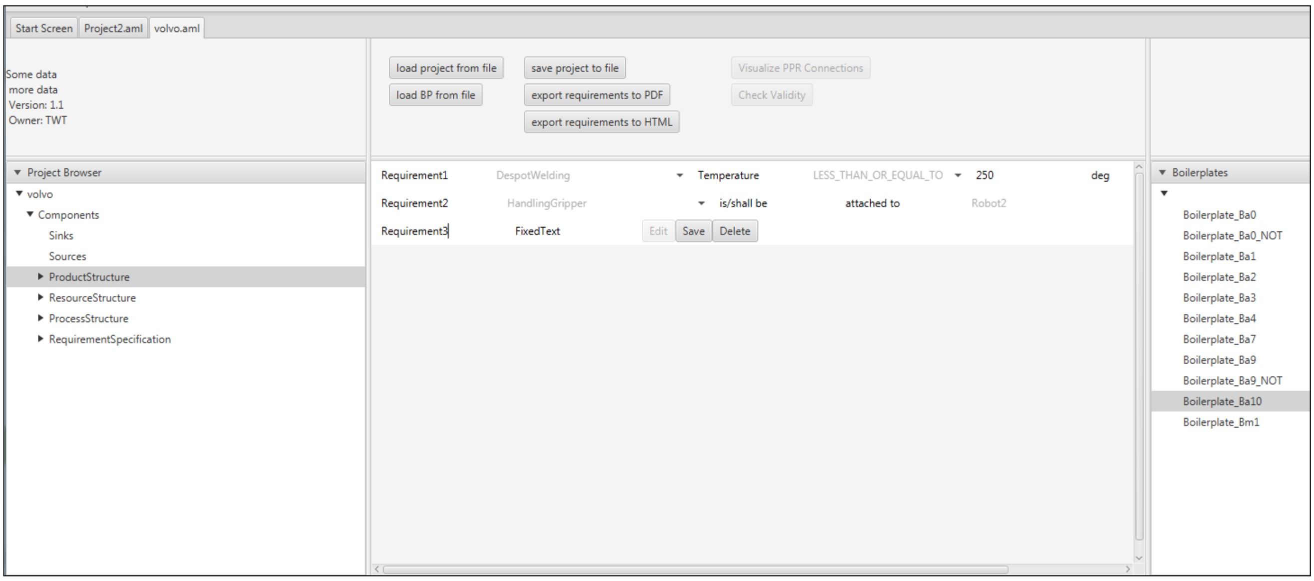


Figure 4.20: Addition of the Boiler plates for the requirements specified in the Scope of Supply.

Figure 4.20 above shows the addition of the requirements formalized using the generated boiler plates with the aid of the Boilerplate tool. The boiler plates are generated from the existing objects in the AutomationML file i.e. the resources, the product and the processes.

Once the requirements were added using the generated boiler plates, the AutomationML file was exported. The output file showed the additional hierarchy that is the RequirementsSpecification hierarchy whose internal elements were the formalized requirements. The requirements are structured just like the Boilerplates used to generated them.
Figure 4.21 below shows some of the requirements formalized for this test case as generated using the boiler plates.

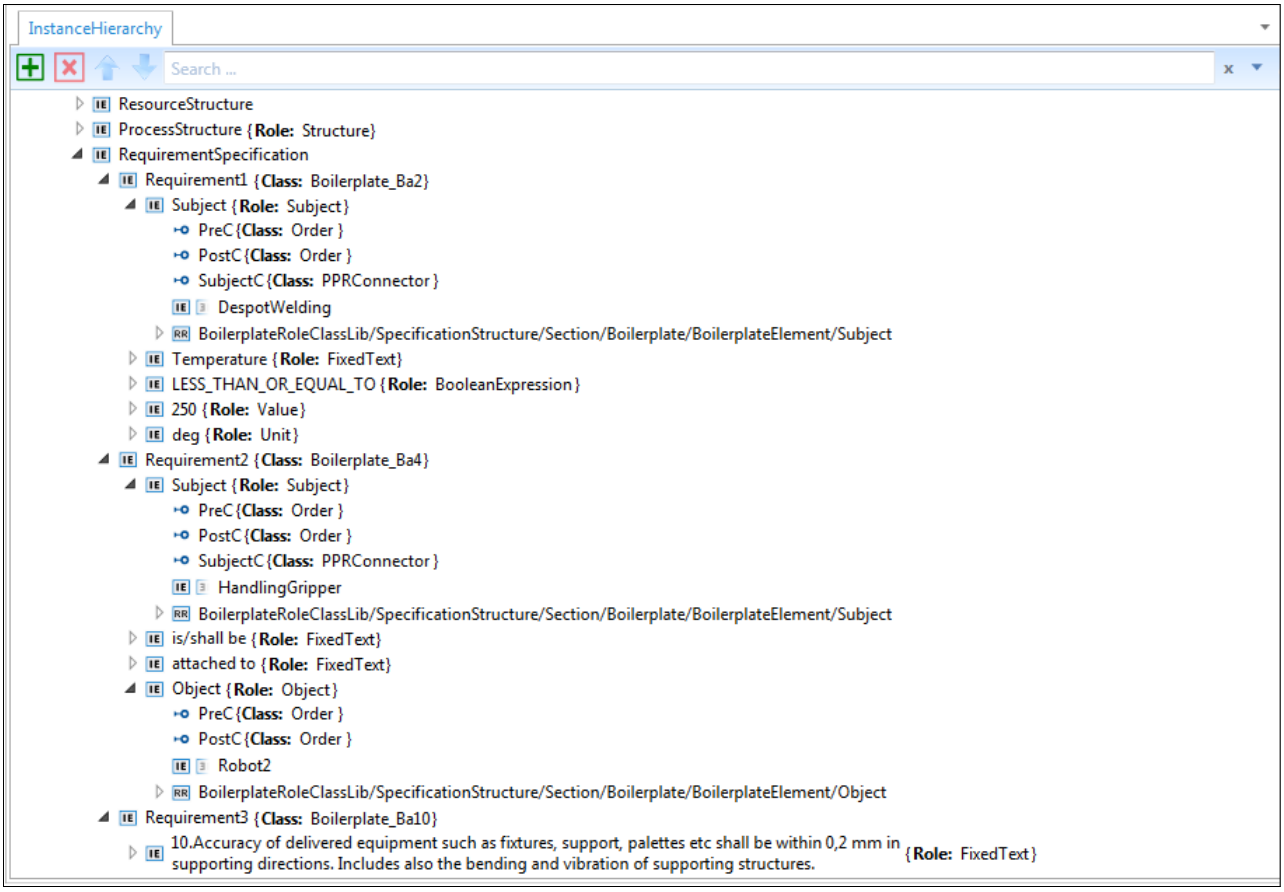


Figure 4.21: Exported AutomationML file showing some of the requirements added using boiler plates.

4.1.2.4 Analysis of the Production cell feasibility using the PPR_AnalyZ3r Tool

To analyze the feasibility and allocation of resources of the production cell, the AUtomationML file exported form the Boilerplate tool was used in the PPR_AnalyZ3r tool.
The PPR_Analyzer tool is a command line based tool and had not graphical interface at the time this thesis was done. Hence the results are generated in PDF form. For this test case, the PPR_AnalyZ3r produced the results shown in figures 4.22-4.23 below.

```
Variants/Items chosen for this configuration

Hinge bracket RH
Latch bracket LH
Latch bracket RH
Security lock Bracket
Anti Flatter Glue
Structual Glue
Front Lid
Hinge bracket LH
Front Lid Inner Panel
Front Lid Outer Panel

Operations necessary to build the product

LoadingPR
GeometricWeldingPR
MergingPR
HemmingPR
RespotWeldingPR
UnloadingPR
AntiFlutterGluingPR
StructuralGluingPR

The scheduling for the operations is:

LoadingPR: START = 0 END = 16
GeometricWeldingPR: START = 16 END = 56
MergingPR: START = 109 END = 119
HemmingPR: START = 119 END = 178
RespotWeldingPR: START = 178 END = 340
UnloadingPR: START = 340 END = 356
AntiFlutterGluingPR: START = 56 END = 82
StructuralGluingPR: START = 82 END = 109

Resource Matching:

Worker=>LoadingPR
DepositWeldingFixture=>LoadingPR
MergeFixture=>LoadingPR
Robot1=>GeometricWeldingPR
DepositWeldingFixture=>GeometricWeldingPR
Robot2=>MergingPR
MergeFixture=>MergingPR
Robot3=>HemmingPR
HemmingBedFixture=>HemmingPR
Robot1=>RespotWeldingPR
RespotWeldingFixture=>RespotWeldingPR
Worker=>UnloadingPR
RespotWeldingFixture=>UnloadingPR
Robot2=>AntiFlutterGluingPR
```

Figure 4.22: Tool output showing optimal scheduling and resource matching.

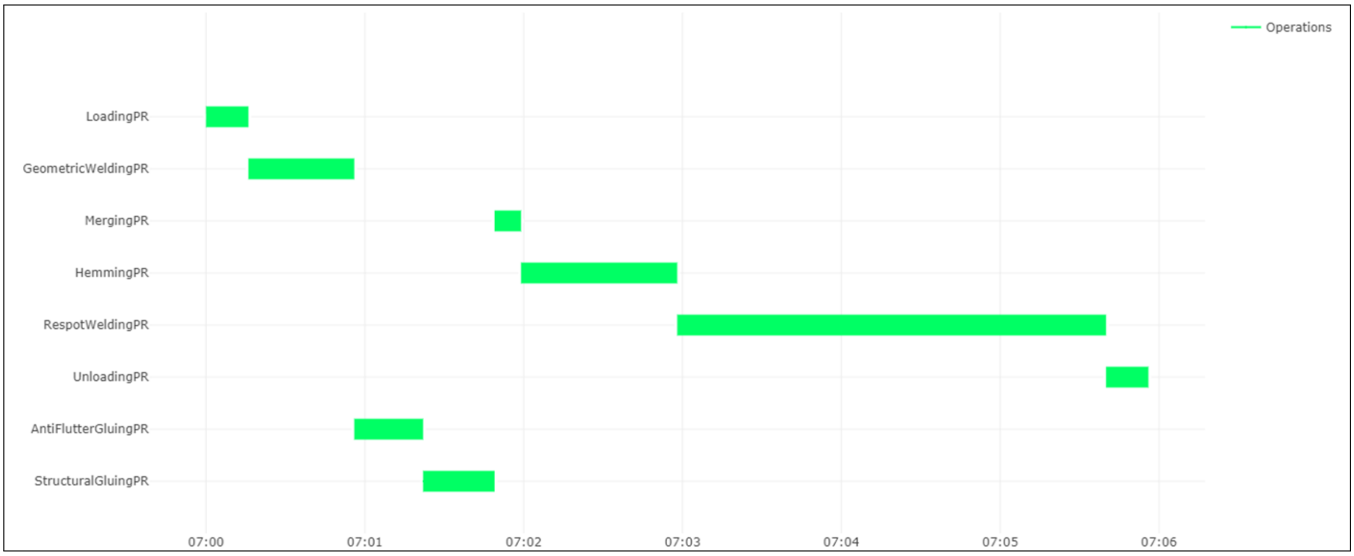


Figure 4.23: Gantt Chart produced by the tool showing the optimal scheduling of operations.

Figure 4.22 shows the optimal scheduling and resource matching for this production cell. The scheduling coincides with the process description and the reason for this is because since this is an already operational production cell, the process sequence and scheduling as well as the resource allocation has already been defined in the requirements hence no optimization is actually being done by the tool. This however is not the case for a green field scenario as seen in test case two.
Figure 4.23 shows the Gantt chart produced by the tool with the scheduling of the different processes that make up the production in this test case. As can be seen, the process sequence is that each process begins after the one before is finished and there are no parallel processes.The reason for this is that only one product is being processed in the cell and so the sequence flow will be one process after the other.

With the feasiblity check done, the formalization of the requirements for this cell was completed. However the milestone planning was not done for this particular test case since no milestones were set as it is already an exisiting production cell with every piece of equipment already installed.

4.2 Test Case Two - Front and Trunk Lid Assembly Cell

This test case is based on a proposed production cell for the assembly of both the P2545 front lid from test case one and the P2545 luggage/trunk lid. For the assembly of the front lid, the same equipment and process description is maintained as in the first test case. However, for the assembly of the trunk lid, additional equipment is to be added. A clinching station, an additional hemming bed, a fourth robot as well as a deposit fixture are to be added to the production cell from test case one to enable the production of the trunk lid. Resources in the cell are to be shared in the assembly of these two products. The resource sharing must be coordinated to ensure that both products are assembled concurrently.

The ENTOC approach was applied on the this test case and the same steps were taken as in the first test case. This steps are described in the sections below;

4.2.1 Analysis of the Request For Quotation (RFQ)

Since the test case is based on a proposed production cell which is a modification of the cell in the first test case, the RFQ is essentially the same with just a few modifications in some sections. These sections are the product description, the layout description, the process description, the milestone plan specification as well as some additions to the scope of supply.

4.2.2 Formalization of the Requirement Specification using the ENTOC Approach and Tool-Chain

4.2.2.1 Formalization of the Layout and Product Specification

Figure 4.24 shows the design layout of the new production cell. It shows the resources used in production and also the product parts going into the production cell and at what locations they are fed into the production cell.

The taraVRBuilder tool was used for the formalization as in the second test case. The Role class library was defined and the requirements specified for the different resources in the cell same as the first test case.

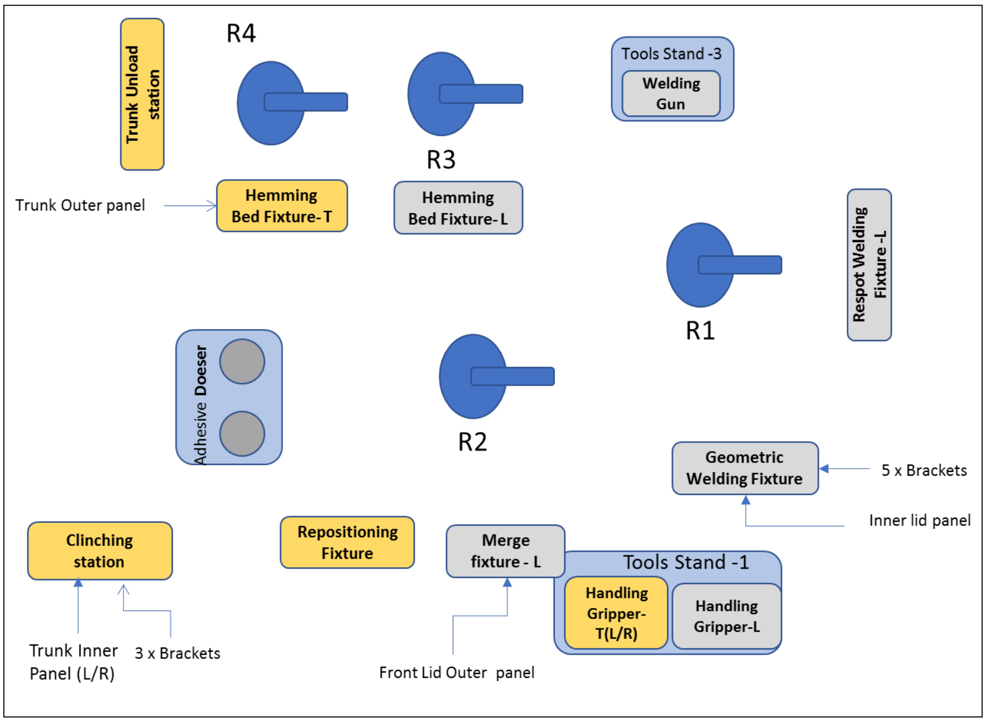


Figure 4.24: Design layout of the new production cell.

The AutomationML file exported from the taraVRBuilder is shown in figure 4.25 below. The figure shows the product and resource structures added by the taraVR-builder tool.

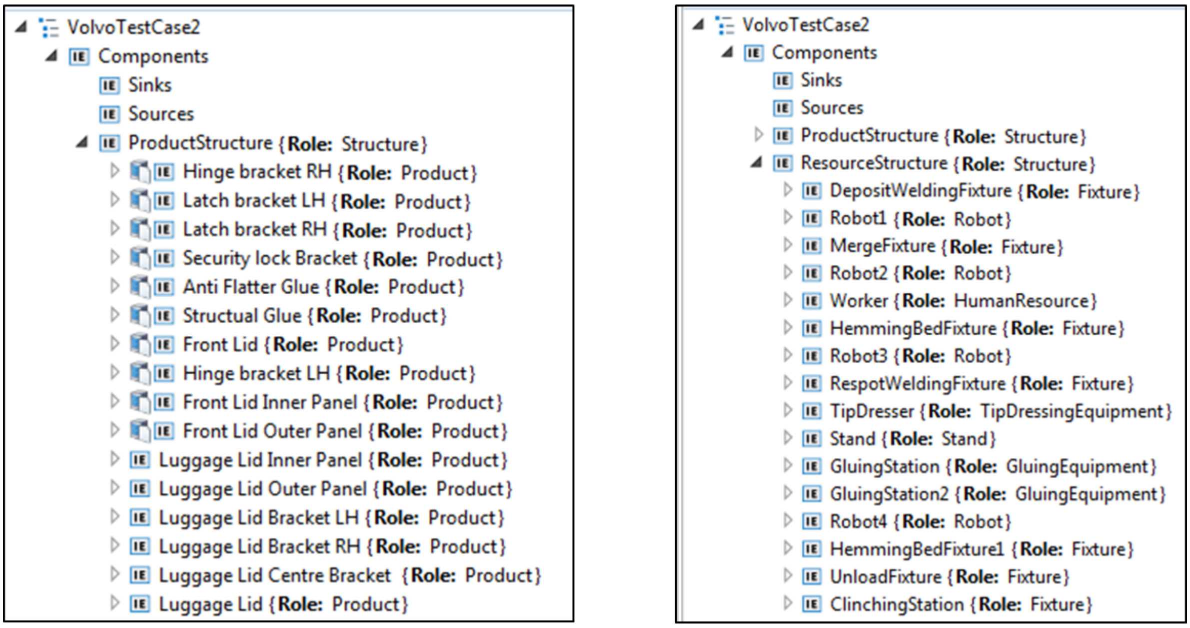


Figure 4.25: The resource and product structures generated by the TaraVR-Builder.

4.2.2.2 Formalization of the Process Specification

The production cell used in this test case as mentioned earlier is used for the assembly of two products i.e. the front lid from the first test case as well as the trunk lid. The process description for the front lid is maintained as in the first case and the same resources/equipment are used for the assembly of that cell. The process description for the trunk lid is shown in figure 4.26 below.

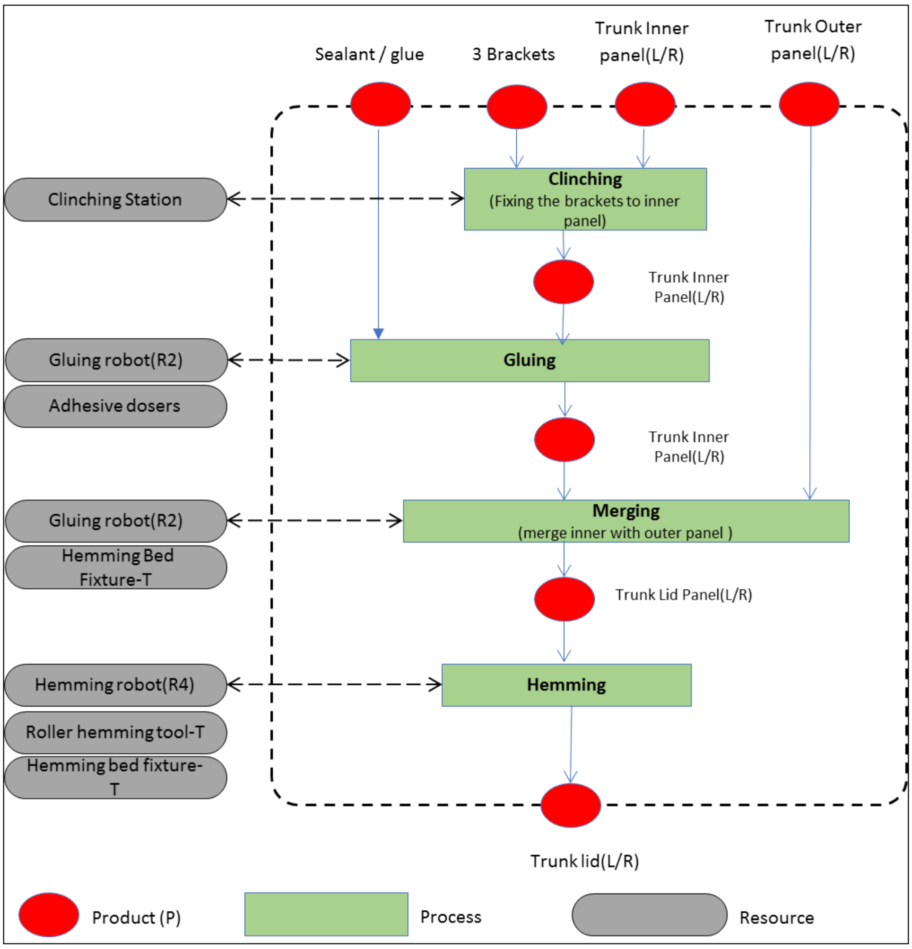


Figure 4.26: Process Description for the trunk lid production.

The AutomationML file exported from the taraVRbuilder tool is then imported into the Boilerplate tool. The process structure, the process sequence as well as the PPR modeling is then implemented using the Boilerplate tool. The process structure is created as two independent process structures each corresponding to one of the two products being assembled.
Resources are shared for the two process sequences and coordination is neccessary and this will be optimized using the PP_AnalyZ3r tool as shown in section 4.2.2.3. Figure 4.27 below shows the AutomationML file resulting from the addition of the process structure and the PPR modelling by the boilerplate tool.

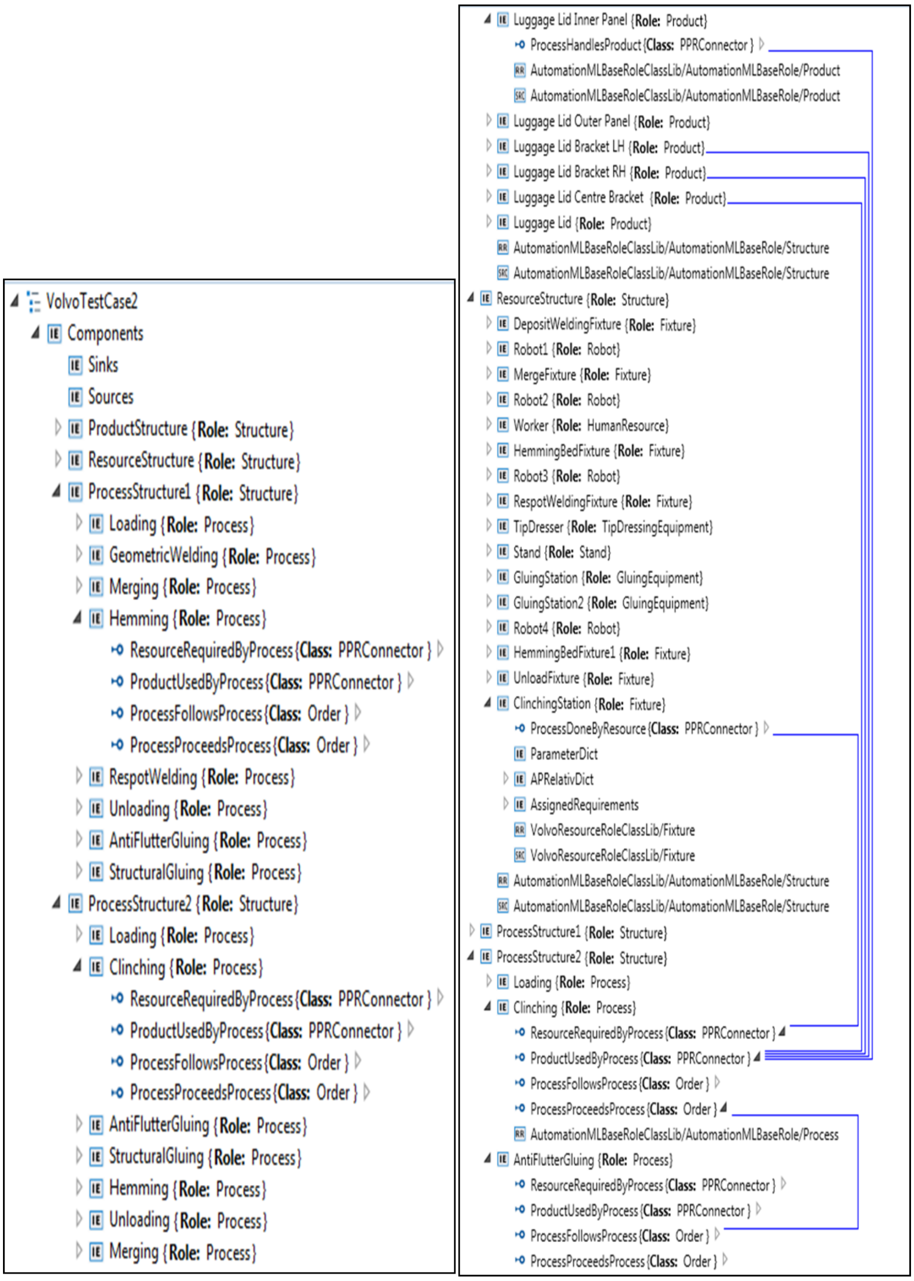


Figure 4.27: The exported file from the Boilerplate tool showing the added process structures and the PPR modelling of the production cell.

4.2.2.3 Analysis of the Production cell feasibility using the PPR_AnalyZ3r Tool

After the PPR modelling of the production cell was done and reflected in the AutomationML file exported from the boilerplate tool, the feasibility and resource allocation was analyzed using the PPR_AnalyZ3r tool to ensure that the sequencing of operations and the allocation of resources did not result in a deadlock and that both products could be produced without any clashes in the operations. This analysis produced a Gantt chart sowing the sequencing of the operations for the two products. This gantt chart is shown in figure 4.28 below. The Gantt chart shows that both sets of operations are carried out without any clashes and both products can be produced in the same cell without any deadlocks resulting from operations competing for a given resource.

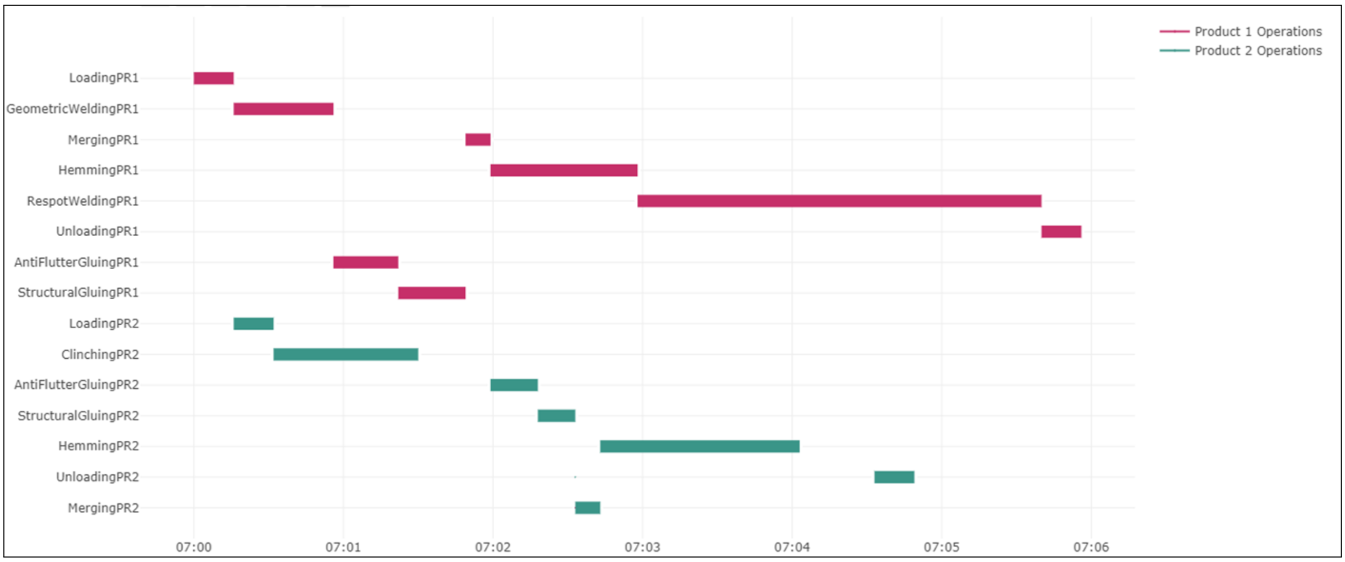


Figure 4.28: The exported file from the Boilerplate tool showing the added process structures and the PPR modelling of the production cell.

With this production cell, the functionality of the PPR_analyZ3r tool was exhibited and its usefulness in producing an optimal sequence of operations taking into account the two sets of operations vying for the same available resources was highlighted. With a feasible sequence generated, the production cell was then proved to be feasible.

5
Evaluation of the ENTOC Formalization Approach on the Volvo Test Cases

In this chapter, the results from the evaluation of the ENTOC requirements formalization approach performance on the Volvo test case is presented. The evaluation is done based on how the formalization performs in regards to criteria described in section 2.

5.1 Evaluation criteria 1 - Requirements coverage

This evaluation was based on statistics on how many requirements in the RFQ specification were formalized using the approach with the aid of the tools.
The RFQ was composed of different sets of requirements as explained in the chapter above. The statistics are hence broken down based on this classification of requirements.

- Invitation Letter.
  No requirements are specified in this document hence no statistics can be provided.

- Agreement.
  No statistics could be provided for this since no requirements are contained in this set of documents.

- Cost Split up.
  For this set of documents, no requirements are specified hence no statistics could be drawn out.

- Time Schedule.
  For this set of documents, requirements regarding time plans are stated. With the Milestone planning tool, the time plans were represented with milestones. 100% of the requirements in this section could be formalized using the milestone planning tool by defining the time plans as milestones and result types.

- Scope of supply.

The scope of supply section of the RFQ was a 21 page document consisting of 250 requirements all of which were non-functional requirements. 18 of these requirements were pertaining to the resources in the layout and were covered using the taraVRbuilder tool. Another two of these requirements were pertaining to the resources however they were covered using the Boilerplate tool. The remaining 230 requirements were comments on the general expected performance of the production cell and were therefore covered using boiler plates with the help of the boilerplate tool.
Hence 100% of of the requirements in the scope of supply section were digitalized using the ENTOC formalization approach however a small percentage of these were formalized in both syntax and semantics.

- Performance technical specifications.
  This set of requirements consists of 17 pages, five of this pages are requirements in text form while the remaining 12 pages are made of of graphics. In total 84 functional requirements and two non functional requirements are available.
  20 of these requirements pertain to the layout and resources of the production cell and were hence covered using the taraVRbuilder tool while 63 of the requirements were related to the process description and were covered using the boilerplate tool. Three of these requirements were comments on the functionality of the cell and for these boiler plates were used to cover them.

The figure 5.2 below summarizes the statistics on how many and how the requirements in the Volvo requirements specifications were covered using the approach.

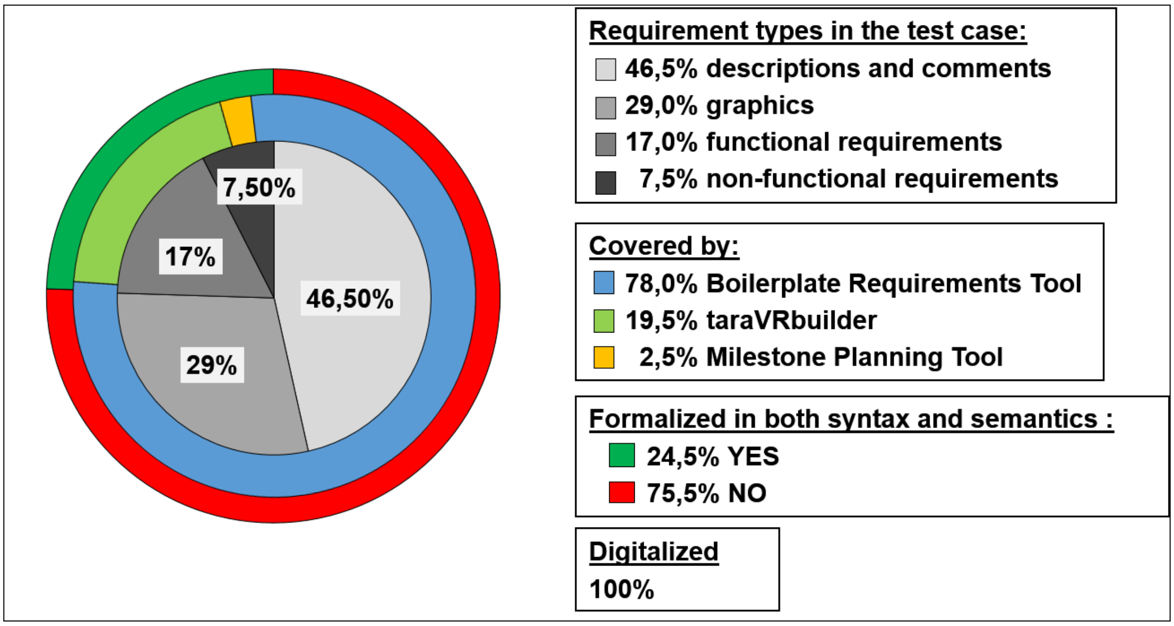


Figure 5.1: Figure showing the requirements coverage statistics for the Volvo test case one.

5.1.1 Answer to the Research Question 1

This evaluation provides the answer to the first research question from section 1.3 which was;
To what extent does the ENTOC formalization approach and tools cover the requirement specification of the Volvo test cases?
The ENTOC approach with the use of the tools digitalizes 100% of the requirements in the Volvo test case. This means that 100% of the requirements can be added to the AutomationML file which is the data exchange format used. However, 24.5% of all requirements can be formalized in both syntax and semantics.

5.2 Evaluation criteria 2 - Compatibility and use in other Production system design tools

This evaluation was based on the ability to use the formalization output in other production system design tools for example PLM systems or production system simulation software.

The AutomationML file output from the tool-chain can be interpreted by three of the tools in the chain i.e the boilerplate tool, the milestone tool and the PPR_analyZ3r. The taraVRbuilder tool does not have the capability to import AutomationML files. As it currently stands, the AutomationML output file from the formalization can not be interpreted or used in any other known engineering tools used in production system design at Volvo Trucks. However the goal of the project was to create a proof of concept which can then be demonstrated to other engineering tool developers with the hope of enabling adaptability of the tools to interprete the formalization output.

5.3 Evaluation criteria 3 - Re-usability of formalized output data file

This evaluation was based on the ability to reuse or modify the formalization output file when changes to the production system are implemented and on the ease with which this can be done.
The first tool in the chain i.e the taraVRbuilder tool which is used for formalizing the layout, resource and product requirements does not support re-usability since it has no functionality to import an existing AutomationML file. However the rest of the tools in the chain i.e the boilerplate tool, the milestone planning, tool allow for re-usability of the AutomationML file. This enables any modification in the production system or in the requirements to be edited in the formalized AutomationML output. A summary of the import and export capabilities of the tools is given below;

ENTOC Tool	AutomationML file import capability	AutomationML export capability
TaraVRbuilder	No	Yes
Boilerplate Tool	Yes	Yes
Milestone planning tool	Yes	Yes
PPR_AnalyZ3r	Yes	No

Figure 5.2: Summary of the AutomationML import and export capabilities of the tools.

5.3.1 Answer to the Research Question 2

This evaluation provides the answer to the second research question from section 1.3 which was;
What are the re-usability capabilities of the formalized requirement specification from the ENTOC formalization approach?
The formalized specification can be re-used to some extent. The Boilerplate tool and the milestone planning tool are the only tools which can make any modifications to the specification in AutomationML format. Using these tools, a specification can be imported and requirements added or modified to correspond to any modifications made in the production system.

6
Conclusion

The main purpose of this thesis as explained at the beginning of this report was to apply the formalization approach with the help of the tools developed during the ENTOC project on two Volvo test cases and then evaluate its performance on the test case based on a specified evaluation criteria.

The formalization was focused mainly on requirements pertaining to the cell layout, resource and product description as well as the Process description and general cell text-based requirements.

Based on the analysis of the requirements and the formalization steps done in this thesis, the following conclusions on the ENTOC formalization approach were drawn;

- The approach digitalizes a large percentage of the test case requirements. As analyzed, 100% of the requirements can be stored in the intended data exchange format which is AutomationML. However a small percentage of the requirements 24.5% is formalized in terms of both the syntax and semantics.

- The output file from the formalization approach is reusable and can be updated to reflect changes in the process description. However changes in the layout can not be updated due to the inability of the TaraVRBuilder tool (the tool used for layout formalization) to import an AutomationML file.

- The output data file from the formalization can be exchanged within the tools developed to aid the formalization however the file format is not compatible with any traditional or commonly used engineering tools or applications.

6.1 Suggestions for Future Work

With the formalization approach developed and the output file from the formalization defined, further research and investigation is needed on how to incorporate the output file into other traditional engineering tools such as PLM systems, simulation tools among others. Further research is also needed to cover other aspects of the production system such as the control engineering information and the production system geometry and kinematics.

6.2 Ethical and Sustainability Aspects

As engineers, we are part of a broader society that is affected by the results of our work. This thesis work being part of a broader project i.e. ENTOC, which if successful and adopted for industrial use will have both ethical and sustainability impacts on the environment and on society. With this in mind, the ethical and sustainability aspects are focused on the entire project and not just the thesis work. Formalization of the requirements specification has both positive and negative impacts in regards to the ethics and environment as outlined below;

- It speeds up the engineering process since it eliminates misinterpretation leading to faster ramp up times of production cells. This then reduces the strain on workers and potentially increases profits for the production company.
- It helps do away with paper based sharing of the requirements specification since the digitalized requirement specification can only be shared electronically. The digitalized requirements specification can therefore be regarded more environmental friendly.

On the other hand, there is also a negative impact of formalizing the requirements specification as explained below;

- Formalization of the requirements specification is a tool for optimizing the engineering process which in some cases may lead to work redistribution. As a result some workers previously tasked with compiling the traditional requirement specification may be laid off.

Bibliography

[1] ENTOC Deliverable 2.1, “Analysis of the target requirements engineering approach,” April 2017

[2] T. Gilz, “Modellbasierte virtuelle Produktentwicklung – Requirements Engineering und Requirements Management, pages 53-75,” Berlin, Germany:Springer-Vieweg, 2017

[3] U.Eriksson, “The difference between functional and non-functional requirements,” [Online]. Available: https://reqtest.com/requirements-blog/understanding-the-difference-between-functional-and-non-functional-requirements/., Accessed on: 23/05/2019.

[4] P. Florent, J. Yang and G. Mohamed, “A Formal Framework for the Formalization of Informal Requirements. The International Journal of Soft Computing and Software Engineering. p14-27”, Paris, France:HAL Archieves, Aug 2013.

[5] D. Rainer, J. Michael and P. Johanna, “AutomationML Brochure,” [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1544706233-automationml.pdf, Accessed on: 23/05/2019.

[6] AutomationML Consortium, “Whitepaper AutomationML Part 1 - Architecture and general requirements,” [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1485867599-AML_Whitepaper_Architecture_V2.0.0.zip, Accessed on: 23/05/2019.

[7] IEC 62424:2016, “Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCECAE tools,” [Online]. Available: https://webstore.iec.ch/publication/25442, Accessed on: 23/05/2019.

[8] ISO/PAS 17506:2012, “Industrial automation systems and integration – COLLADA digital asset schema specification for 3D visualization of industrial data,” [Online]. Available: https://www.iso.org/standard/59902.html, Accessed on: 23/05/2019.

[9] AutomationML Consortium, “Whitepaper AutomationML Part 4: AutomationML Logic,” [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1490340053-AML_Whitepaper_Logic_V1.5.0.zip, Accessed on: 23/05/2019.

[10] AutomationML Consortium, “Whitepaper AutomationML Part 2 – Role class libraries,” [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1485868210-AML_Whitepaper_RoleClassLibraries_V2.0.0.zip, Accessed on: 23/05/2019.

[11] AutomationML Consortium, "AutomationML Whitepaper Part 3 – Geometry and Kinematics," [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1489067636-AML_Whitepaper_GeometryKinematics_V2.0.0.zip, Accessed on: 23/05/2019.

[12] AutomationML Consortium, "Whitepaper AutomationML Part 4: AutomationML Logic," [Online]. Available: https://www.automationml.org/o.red/uploads/dateien/1490340053-AML_Whitepaper_Logic_V1.5.0.zip, Accessed on: 23/05/2019.

[13] A. Ebrahimi, K. Åkesson, P. Johansson and Thomas Lezama, "Automated analysis of interdependencies between product platforms and assembly operations," 6th CIRP Conference on Assembly Technologies and Systems (CATS), p1-3, 2016.

[14] Jobshop Problem:2019, "Process Scheduling Job shop problem," [Online]. Available: https://developers.google.com/optimization/scheduling/job−shop, Accessed on: 07/06/2019.

[15] B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman,and K. Åkesson, "Sequence planning for integrated product, process and automation design," Automation Science and Engineering, IEEE Transactions, vol. 7, no. 4, Oct 2010.

[16] M. Navaei, L. Javad, and H. ElMaraghy. "Grouping part/product variants based on networked operations sequence," Journal of Manufacturing Systems 38, P 63-76, 2016.

[17] S. Roselli, K. Bengtsson, and K. Åkesson, "SMT solvers for job-shop scheduling problems: Models comparison and performance evaluation," in Congress of CASE, the Portuguese Operational Research Society, 2017.

[18] ENTOC Deliverable 2.2, "Language for formalized requirements engineering," April 2017.

[19] Tarakos GmbH, "Functional specifications for the taraVRbuilder," [Online]. Available:http://www.tarakos.de/images/pdf/Functional-specifications_taraVRbuilder_2018.pdf, Accessed on: 07/06/2019.