



October 1th 2013

Lean Engineering Design of Rail Interlocking Systems with RailML

Thesis Report Master TIL – Mark Bosschaart – 1371851

Prof.dr.ir. L.A. Tavasszy	-	TU Delft, TBM, TLO
Dr. R.M.P. Goverde	-	TU Delft, Citg, T&P
Dr.ir. E. Quaglietta	-	TU Delft, Citg, T&P
Dr. W.W.A. Beelaerts van Blokland	-	TU Delft, 3me, TEL
Dr.ir. B.J.R. Janssen	-	Siemens NL, Rail Automation

Foreword

This report is a master thesis project of Transport, Infrastructure & Logistics (TIL) at Delft University of Technology. The TIL master program has a multi-disciplinary character with a study program that mainly covers courses at Mechanical Engineering, Civil Engineering and Technology, Policy & Management; to a lesser extent also at Aerospace Engineering and Applied Mathematics. The TIL program knows four specializations of which this thesis touches the 'Transport & Supply Engineering' specialization. The specialization, and therefore the master thesis project, aims to apply transport technologies and planning methodologies in the context of supply chain optimization. One of those transport technologies is rail.

During the 2012 High Speed Rail Congress in Philadelphia, I met Mr. Chlebowski, Siemens' vice president of sales & projects locomotives. This contact resulted in a trip of the Dispuut Verkeer, the TIL students' association, to Siemens' test track at Wildenrath in Germany. I was positively surprised by the company Siemens and approached mister Mink, head of rolling stock in the Netherlands, for the possibilities of a graduation internship. Unfortunately, this request did not result in a project offer. A class in railway traffic management given by guest lecturer Bob Janssen of Siemens, provided me with a new chance to ask for a graduation internship; this time with success. He proposed to study the lean engineering of rail interlocking systems. I favored the character of the project most and foremost due to a close relation with the study program and because of my personal interest in the railways. As of March 2013, I started with the project that lies before you.

The report intends to enrich the knowledge of interlocking engineers at Siemens, signaling experts at the Dutch infrastructure provider Prorail, developers of the RailML group and process chain analysts. The result provides reason for Siemens and Prorail to continue with RailML, it provides the RailML group with a cornerstone for the formalization of interlocking and gives process analyst a first look into the effects of the lean management approach in engineering design processes. During a RailML congress in Paris of September 2013, industrial parties and infrastructure managers of the RailML interlocking group indicated to study the developed formalization in order to apply it in the future.

Readers that have a little amount of time, are advised to consider the summary, the conclusion to the main research question (section 6.2) and the recommendations. Readers having interest in the lean engineering design transformation methodology should consider chapters 3 and 5. Readers that have interest in the RailML interlocking formalization should especially consider chapter 4. Readers that mostly have interest in the performance of RailML, should consider chapter 5.

Special thanks in the first place go to Bob Janssen and Siemens who provided me with the possibility to execute

this graduation internship. Second, I would like to thank my graduation committee for their devotion and advices. Third, I want to thank the involved people from Prorail, and especially Sander Dragt, for providing data, answering my questions and allowing me to interview various involved people at Prorail. Fourth, I would like to thank the associated people at Siemens in Braunschweig for providing various data and information. Fifth, I would also like to thank the other employees of the Rail Automation department at Siemens the Netherlands who were very welcoming and also provided the necessary data and information.

May this report lead to a more attractive railway industry!

The Hague, October 1th 2013,



Mark Bosschaart

Summary

An interlocking system forms a vital part of a railway signaling system because it ensures the safe movement of trains. For one by controlling movable track elements like switches and for another by controlling signals.

The engineering design of interlocking systems poses challenges to both engineers at e.g. Siemens and the infrastructure manager, e.g. Prorail in the Netherlands. Challenges arise from the cumbersome, labor intensive, conservative, ambiguous and failsafe nature of the interlocking engineering design processes.

Best practices in lean production systems raise Siemens' interest in a lean engineering approach, i.e. reduce the 'waste' that does not contribute value to the final rail product. The development of RailML, an open source IT tool that aims to standardize the exchange of data in a variety of rail processes, may prove the lean catalyst. The lean philosophy has however seldom been applied in engineering design processes. Therefore, this research aims to provide insight in the drivers of the engineering design chain that decrease development time, increase data reliability and make application more flexible, while ensuring (fail-)safe systems. For that purpose, this thesis investigates the next research question:

To which extent can RailML potentially enable a leaner engineering design of rail interlocking systems?

This study focuses on the Dutch railway network. Furthermore, this study only considers engineering design processes and RailML as the data exchange tool.

Five sub research questions lead to an answer:

Which methodology leads to a lean transformation of rail interlocking systems' engineering design?

Best practices about lean transformations lead to a methodology that commences with defining complexity into standardized and varying requirements of the final design. A future lean structure of the chain follows from mitigating 'waste' and achieving lean transformation directions. The more standardized the requirements, the higher the potential for an IT tool like RailML to mitigate 'waste'. Case specific transformation directions include 'waste' prioritization, process standardization, open source IT tools, a modular process and information sharing. A model study allows to quantify the improvement of a strategy like RailML on the status quo and benchmark the lean effect. Four criteria count: costs, time, interdependencies and ambiguities.

How would a future lean interlocking engineering design change the status quo?

The status quo contains twenty processes that struggle with non value added work in progress, many (non machine readable) data transfers, ambiguous interlocking design from scratch, high fixed costs, many validation processes and low productivity rates. A lean engineering design process mitigates most of that 'waste' by concentrating on five main processes: project definition, data preparation, interlocking area design, interlocking engineering and a dry test. Realistically, varying requirements and completely new interlocking areas prevent full automation of the engineering design chain.

How should interlocking be implemented into RailML for the purposes desired by Prorail?

Prorail proposed to use and provided the files of the Santpoort Noord interlocking area as a case study to formalize the interlocking in RailML. A RailML interlocking formalization from the perspective of a train requesting a route, results in a complete, concise and RailML group coherent interlocking formalization that allows to capture all necessary rail elements for (Siemens') interlocking engineering and the two-block three-aspect system ATB-

EG. The RailML structure contains two main sub elements: signals and segments. The signal element captures aspect dependencies on the basis of entrance speed profiles, signal aspect and target signal combinations. The segments element covers route relevant elements. In the case of ETCS level 3, the interlocking formalization might ultimately become superfluous.

How does RailML change the interlocking engineering design chain when implemented as the rail data exchange tool for railway signaling systems in the Netherlands?

RailML mainly improves the design stage by machine readable transfers, elimination of interlocking area design from scratch and automates test protocol development. At the engineering stage, RailML eliminates the data preparation process and a part of interlocking IT engineering. These improvements do however assume the existence of a single data source in the chain e.g. a GIS. RailML mainly lacks the ability to achieve a modular engineering design, continuous process flow, improved project management and lean perfectionism / learning effect.

What benefits does the application of the RailML formalization have during the engineering design of an interlocking area over the traditional and lean process structure?

RailML reduces Siemens' costs between 17%-27% and covers 21%-35% of a lean cost reduction, depending on project complexity. RailML improves productivity by 26%-47% although this only covers 6%-14% of a lean improvement. RailML relatively improves transaction costs but needs a larger average of non value added work-in-progress to achieve so. RailML reduces the tier rejection rate by 47%-77%, which is 50%-81% lean. RailML worsens the non value added work-in-progress for high complexity projects. RailML shows deviating lead time performance: a reduction of 2%-25% which is 2%-64% lean. RailML reduces the current amount of design rules by 55%, which is 85% lean.

In general, RailML improves the performance of the interlocking engineering design chain on almost every performance metric, for each project complexity. RailML covers about 43% of the lean state for a medium complexity interlocking area as Santpoort Noord. Academically, the proposed interlocking formalization excels in its way of incorporating signal aspect dependencies, minimal amount of data (barely any redundancy) and alignment with both industry and infrastructure managers. In fact, on the RailML congress of September 2013, the RailML group decided to study the implementation of the developed formalization for different signaling systems than the Dutch and different engineering processes than Siemens'. The lean methodology leads to a scientific benchmark of a strategy's, e.g. RailML's, lean performance. The methodology however needs a substantial amount of data and does not seem to work as well for high complexity projects as simpler ones.

The author mainly recommends Siemens and Prorail to continue with the development of RailML, to start an interlocking engineering design pilot project with RailML and to enlarge the degree in which interlocking areas can be standardized.

From an academic standpoint, the author mainly recommends to study whether the interlocking formalization works for different signaling systems, infrastructure managers and interlocking engineers, to quantify the effects on the safety case, to investigate effects of more detailed data, to find design modeling techniques that work on a micro level and to test the hypothesis that high complexity engineering design projects suit less for a lean transformation.

Table of Contents

Foreword.....	iii
Summary	v
1. Introduction.....	1
2. Problem & Research Statement.....	3
2.1 The General Interlocking Context	3
2.2 Rail Interlocking	4
2.3 System Limitations of Engineering and Designing Interlocking Systems.....	5
2.4 Lean Engineering Design	6
2.5 RailML	7
2.6 The Interlocking Engineering Design System in a Nutshell	8
2.7 Research Questions.....	9
2.8 Academic and Business Contribution of this Thesis.....	11
3. A Lean Benchmark for Engineering Design Processes	13
3.1 The Lean Transformation Methodology for Engineering Design Processes.....	13
3.2 Prorail’s Requirements for Interlocking Systems	20
3.3 The Current State of the Interlocking Engineering Design Process.....	24
3.4 The Lean State of the Interlocking Engineering Design Process	32
4. RailML Development for the Purposes of Interlocking Engineering Design.....	37
4.1 The Development of RailML v2.2 for Application in the Dutch Railway Network	37
4.2 The RailML Interlocking Formalization	42
4.3 The Effects of the New Train Control System ETCS on the Interlocking Formalization	51
5. RailML’s Improvement of the Interlocking Engineering Design Process.....	57
5.1 RailML’s Degree of Fit in the Interlocking Engineering Design Chain.....	57
5.2 The RailML Performance Model	63
5.3 The Lean Performance of RailML	71
5.4 Discussion of Results.....	79
6. Conclusion & Recommendations	83
6.1 Conclusions on Sub Research Questions.....	83
6.2 Conclusion on Main Research Question.....	85
6.3 Recommendations	86

Appendix A	- RailML Interlocking UML by RailML group	89
Appendix B	- System Analysis	90
Appendix C	- Literature Review on Lean Transformations	93
Appendix D	- Prorail's Lean Design Performance	101
Appendix E	- Siemens' Lean Engineering Performance	104
Appendix F	- Interview Summaries	120
Appendix G	- Sptn Site Analysis	129
Appendix H	- Element Complexity Analysis	134
Appendix I	- Sptn in RailML	136
Appendix J	- Sptn Interlocking in RailML	141
Appendix K	- RailML's Data Exchange Coverage	145
Appendix L	- ARENA Model Setup	147
Literature	173
Abbreviations	177

1. Introduction

Siemens designs and produces various solutions required for today's railway operation. Their railway product portfolio includes rolling stock, traction systems, communication equipment, tunnel equipment and signaling systems (Siemens, 2013b). Railway signaling systems form an important part of Siemens' Infrastructure & Logistics sector portfolio, generating at least 40% of total profit which equals to about 17% of the total signaling market (Sheahan & Jones, 2012; Siemens, 2012). Railway signaling systems ensure the safe guidance of trains. The desired level of safety and management of train flows follows from six types of equipment: track-free detection, interlocking, signaling, train protection, train describers and traffic control support (Goverde, 2012d).

An important strategy of the Siemens Infrastructure & Logistics sector focuses on expanding its rail signaling market share as illustrated with their recent acquisition of Invensys (Siemens, 2013a). Besides acquisitions, Siemens tries to increase its market share by structurally selling more signaling equipment to rail infrastructure managers. The Dutch market for interlocking systems has high potential to achieve that goal. First, because the Dutch railway network does not have many Siemens interlocking systems. In addition, many of the current interlocking systems need a replacement in the coming decade because of age and a new train control system ETCS (Buseyne, Wego, Vaux, & Secci, 2011; Mansveld, 2013). Prorail, the Dutch rail infrastructure manager, selects which party may design, install and/or maintain a new interlocking system by means of a tender.

The engineering design process of interlocking systems poses challenges for Siemens to place a competitive bid due to historical, technological and value chain management reasons. Historically, infrastructure managers favor familiar systems to minimize operational surprises and simplify maintenance. Technically, an interlocking system needs flawless integration with other railway signaling equipment in place. A combination of various signaling equipment suppliers and country specific requirements, increases development complexity even further. Challenges in the value chain arise because it contains many parties, interlocking area design always starts from scratch and parties do not exchange their data in a machine readable format. A party like Siemens can adapt their products and processes to meet infrastructure managers' needs, but it comes at the cost of longer design time, a more opaque system hierarchy, higher probability of mistakes and more expensive systems/lower profit margins. Both Siemens and the infrastructure manager clearly do not benefit from this situation. In addition, today's safety concerns might remain (Bundes, 2011, 2012; Inspectie Leefomgeving en Transport, 2012; Kuijper & Hendriks, 2011; Reemst, 2007; van Vliet, 2004a, 2004b).

A potential improvement in the engineering design process comes from the open source data exchange tool RailML (RailML.org, 2013b). RailML initially intended to align the various data sources of rail infrastructure elements for the purposes of modeling. Currently, various industrial rail parties, infrastructure managers and academic institutions see potential to also support the interlocking engineering design process with RailML; mostly out of discontent with the current process (Wisotzki & Muehlhause, 2013). RailML may shorten data preparation time, catalyze track data digitization and standardize data between all parties involved in railway signaling. As a part of that idea, the initiators would like to include interlocking in RailML as well. They eventually aim to make the engineering design process leaner, i.e. reduce the 'waste' that does not contribute value to the final rail product (Ohno, 1988c; Womack & Jones, 2003a; Womack, Jones, & Roos, 1990). Therefore, this research aims to provide insight in the drivers of the engineering design chain that decrease development time, increase data reliability and make application more flexible, while ensuring (fail)-safe systems. For that purpose, this thesis investigates the next research question:

To which extent can RailML potentially enable a leaner engineering design of rail interlocking systems?

An answer results from a literature review, (statistical) data analysis, conceptual process modeling, XML modeling and discrete event simulation.

This thesis analyzes the research question specifically for the Dutch railway network managed by Prorail. Furthermore, although more options exist, RailML is considered as the data exchange tool because of its strong performance and support (Linder & Grimm, 2012). Besides, this research investigates only the interlocking engineering design of railway signaling; not production.

The next chapters introduce the topic and the research questions. Then, this study develops a lean engineering design process to serve as benchmark. Accordingly, development of a RailML interlocking formalization provides insight in an engineering design process with RailML. Finally, this thesis models and benchmarks the performance of a RailML state against the lean state.

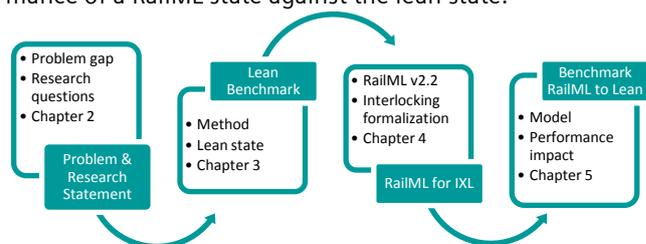


Figure 1: The main research structure

2. Problem & Research Statement

This chapter elaborates on the underlying grounds that make Siemens desire a research project about the implementation and effects of RailML. This starts with the context and the consideration of the engineering design chain. Furthermore, a literature review elaborates on the key aspects of lean, interlocking and RailML. This leads to an overall description of the system boundaries and the research questions that this study needs to address.

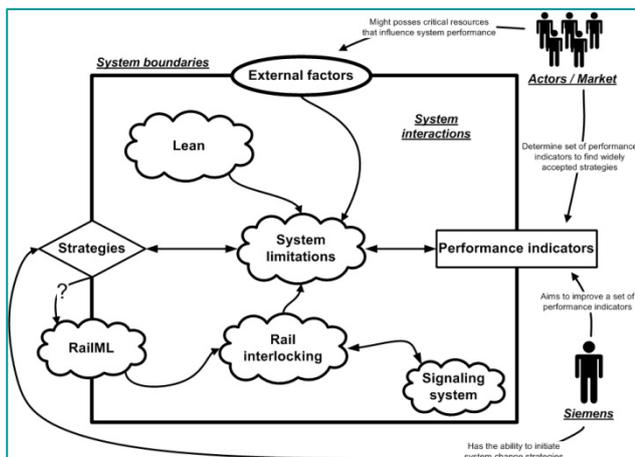


Figure 2: A system diagram visualizes the boundaries / scope of a research topic (de Haan, 2009). The system diagram shows the measures a problem owner, in this case Siemens, can take, which performance indicators he wants to improve and which non-influencable factors he has to take into account.

2.1 The General Interlocking Context

The thesis subject arises in a mixed context of signaling system development, the market between system developers and Siemens' operations. This section states the most important characteristics to provide insight in the dynamics surrounding the subject.

The Railway Signaling System

Rail operation requires a signaling system to guide the safe movement of trains. Three rail operation characteristics demand a signaling system: (1) multiple trains share the same track, (2) trains cannot move easily across track obstacles and (3) rail's low rolling resistance causes trains to have such a long braking distance, that on-sight driving is nearly impossible without collisions (Goverde, 2012). For that purpose, a signaling system runs through a sequence of processes each time that a train requests a section of track (Figure 3 and Figure 4). The signaling loop first detects a train's position. Then, on the basis of the track layout and trains' destinations from traffic control, the system knows where all trains are located and heading to. Second, the system can interlock movable track elements as switches to ensure a train reaches its destination while not causing a collision. Third, the movement authority of a section of track needs to be communicated to a

train via trackside signals or onboard cab communication. Fourth, a protection system needs to ensure that a train will not violate speed and train position regulations. After this fourth step, the cycle starts again for a successive section of track.

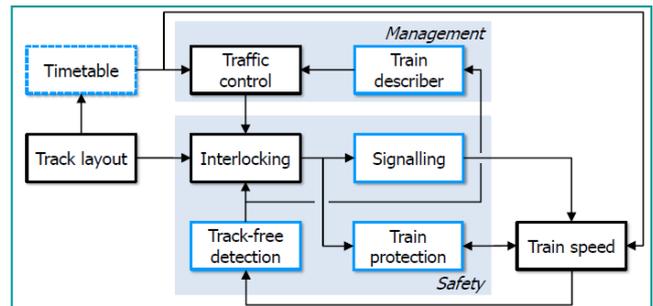


Figure 3: Railway signaling process structure according to Goverde (2012d).

The signal system requires different connected physical (e.g. signals) and digital (e.g. track topology) assets to execute the process loop. Over time, these systems are developed by a variety of organizations like Siemens, where usually a public body or rail service operator takes care of infrastructure management (e.g. Prorail, DB Netz). This leads to a wide variety of signaling systems which functionality is the same, but their architecture is not. Not just between countries or operators, but also at one operator alone like in the Netherlands. This is problematic because the systems need to flawlessly interact in order to ensure interoperability. Furthermore, a signaling system needs to allow for easy modifications when changing the rail infrastructure. Especially because the lifetime of a railway signaling system is long with about 20 to 30 years.

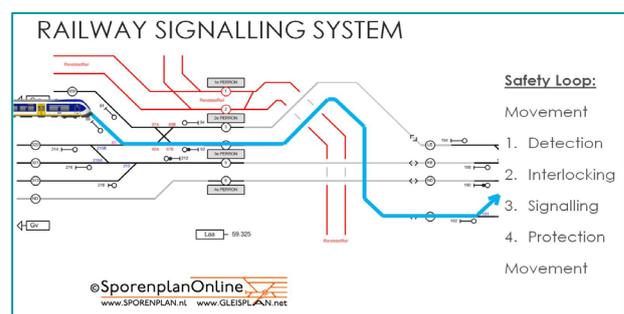


Figure 4: The safety loop as in Figure 3 illustrated for the blue route in the interlocking area of Den Haag Laan van NOI. The track topology originates from SporenplanOnline (2010).

The Railway Signaling System Market

construction of signaling systems requires very specific knowledge that is only well-understood by a few organizations; amplified by the complexity of many different rail systems that need to interact. Therefore, the development of signaling systems is a business of a few incumbents of which Siemens is one. Besides design, engineering and

production, maintenance requires swift action, especially in highly occupied railway networks like in Western-Europe. An infrastructure manager can only deliver swift maintenance with detailed knowledge about the system architecture. An infrastructure manager or engineering agency can therefore not meet the Dutch service level, if they can at all. The producer of the railway signaling system thus often maintains the system as well.

Infrastructure managers on the other hand, impose quite a conservative demand for signaling systems as they desire familiar systems. They demand familiar systems for reasons of safety. Infrastructure managers rarely accept innovations and as a result, few varieties of signaling systems like interlocking exist and the main architecture, ATB in the Netherlands, dates back to 1950. EU regulation however seems to break the conservative nature by demanding an interoperable signaling system architecture across the EU: ERTMS (European Commission, 2013).

In sum, the market has a relatively high entrance barrier based on service and capital, few substitutes, little innovation, high investment costs and few competition (almost an oligopoly). This creates incentives for the buyer, the infrastructure manager, to demand and look for price-cuts. Furthermore, they try to involve more parties in the industry that increase competition. A recent example is Movares that introduces plc's as interlocking system (Blaauboer, 2012).

Siemens' Railway Signaling Systems

The German engineering and electronics firm Siemens produces a variety of railway equipment since 1881: 34 years after Werner von Siemens and Johan Georg Halske founded it in 1847. The rail product portfolio includes rail signaling systems, traction systems, communication equipment and devices, tunnel equipment, passenger information systems and rolling stock. In other words: basically any rail product with the exception of track's substructure and the superstructure.

Siemens' rail signaling systems largely cover the signaling processes and rail management systems as shown in Figure 3 and Figure 4. More specific, Siemens develops level crossing technology, track-free detection mechanisms, interlocking systems, automatic train protection and other components as switch control systems and signals (Siemens, 2013b).

2.2 Rail Interlocking

Rail interlocking literally implies a chain of movable track elements positioned in a desired way to ensure the safe movement of a train (Theeg, Maschek, & Nasedkin, 2009). Movable track elements include those elements that obstruct train movement by track or by appearance in the clearance profile (Theeg & Lykov, 2009). In the Netherlands, common track influencing movable track elements include switches, derailing switches (or catch points) and movable bridges. Common clearance profile obstructing movable track elements include derailleurs, movable bufferstops and water surge barriers. The interlocking system analyzes and controls the dependency between those elements and evaluates track occupancy to safeguard a route. Theeg, Maschek, et al. (2009) and Goverde

(2012c) explain that the interlocking system mainly comprises seven functions for that purpose:

1. track clear acknowledgement from detection system;
2. locking of the route's movable track elements;
3. route holding;
4. flank protection;
5. conflicting route prevention;
6. overlap protection;
7. special functions like the prevention of electric powered vehicles on non-electrified tracks.

The next elaborates on these seven interlocking functions. Railways usually detect a track section's clearance by means of point-based axle counters or line-based track circuits. The Dutch railway network mostly uses track circuits with the exception of the HSL-Zuid and some regional lines. A clear track forms the fundamental requirement for train movement unless a train may operate on sight as is the case with shunting or coupling.

In the Netherlands, electro-mechanic relays still (inter)lock a lot of movable track elements at track sections (Theeg, Nasedkin, et al., 2009). Prorail however equips most new interlocking areas with electronic interlocking systems. The benefit is easy integration with today's ICT systems and probably reduced system complexity as well. Electronic interlocking's main disadvantage comprises the more difficult control of the failure modus. Assuming binary communication, a relay will in the case of an error always return in the same position and thus provide the same signal (0 or 1). In contrast, a computer / channel may show both in the case of an error. Therefore, an electronic interlocking requires a safe and redundant system setup to recognize a system error and take a failsafe action. A safe electronic interlocking system contains at least two hardware channels / computers (two-out-of-two architecture) to process the input detection data from the rail infrastructure. Three channels / computers (two-out-of-three architecture) results in a safe and redundant electronic interlocking system. Redundancy provides a two-out-of-three architecture with the advantage to not shut the system down with one deviating processing result.

Siemens developed the electronic interlocking "Sicheres Mikrocomputer System von Siemens – Welt" (SIMIS W) for the international market. The SIMIS W architecture positions itself between traffic management's interface and the physical infrastructure of a.o. wires, signals and train detectors (Siemens AG, 2013). SIMIS W contains a functional layer responsible for overhead management that controls the interlocking configuration like topology with data derived from the interlocking. The functional layer also provides the logic to translate and verify data between the control layer and the actual interlocking. The logic therefore takes care of interlocking status updates and interlocking assignments. Besides a functional layer, SIMIS W contains an operational logic layer with an area control component at its hart. That component searches, links and controls the relevant movable track elements to activate a route by using the topological data from the overhead management process. The result is a locked route available for a train to take (Figure 5).

The route holding functionality takes care of locking and releasing the movable track elements to accommodate a

train passage. First of all, all elements on a route are locked at the same time; before a train enters that block section. Then, the elements must remain locked when a train passes them or when the block section's signal turns red to announce train presence. In the Netherlands, the release of movable track elements occurs after a train reached the end of the block section (usually the next main signal) or parts of that block section. In the case of a station stop, overlap into the succeeding block section is incorporated.

Flank protection ensures that trains will not be hit by objects from diverging or incoming tracks. To that respect, two scenarios exist: shunting movements on sight or un-fixed rolling stock that enters an active route. One flank protection measure changes movable track elements such that flank movements cannot enter the block section in question. This is enforced by either derailling the rolling stock on the flank or leading it to a different track. When a rail element on a neighboring track does not provide this possibility, the branch of succeeding tracks will be sought until the measure is water tight. Trains can therefore almost impossibly collide, although installation costs will be high. A second way keeps signals on the flanks at stop/red. Although not always very effective since the Dutch ATB, with the exception of ATB-Vv, does not enforce a stop below 40km/h. Even with a safety system that does enforce a stop like ATB-Vv or ETCS, rolling stock can still slide away and enter the set route. Railways use regulation as a third flank protection possibility. In the Netherlands, regulation for example forbids shunting on sight (on main track managed by Prorail). Although a cheap measure to implement, a failure in rule enforcement could have dramatic consequences.

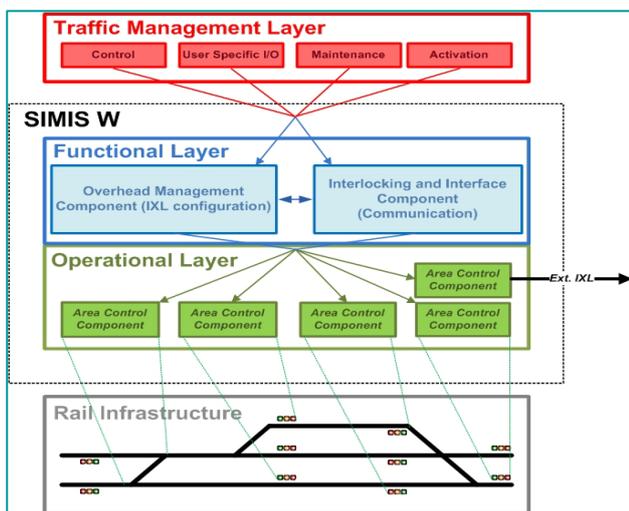


Figure 5: The system architecture of SIMIS W. Area Control Components directly control specific rail elements in the interlocking area. An Area Control Component may also communicate with neighboring interlocking systems. A functional layer connects the Area Control Components and enables access from traffic management applications.

The prevention of conflicting routes starts with a route formation principle. Dutch interlocking areas use both tabular and topological interlocking systems. Tabular interlocking relies on matrices that contain all possible routes with corresponding conflicting routes, switch positions

and flank measures for each possible route. Topological interlocking defines for an interlocking area the set of rail elements and their interdependencies which are based on rules that guarantee safe train movements on the track. Especially in large interlocking areas, the latter method reduces engineering complexity significantly. Either way, both exclude the possibility of a train conflict that could arise when two trains merge/remain on the same track in opposite direction, a train crosses another train's route, two trains 'kiss' each other on a switch complex and with two consecutive trains.

The sixth function of interlocking, overlap protection, concerns the track clear distance up to a signal before an interlocking may grant movement authority. Usually, that distance equals a (number of) section(s) of track between two signals in the same direction: a block section. Overlap protection first extends the safety distance between two consecutive trains to somewhat more than a single block section. A locked block section will not be directly freed when a train leaves the block, though after some overlap distance as a safety margin. Second, overlap protection aims to correct for braking mistakes of the driver. For that purpose, signals in the Dutch railway network are positioned at some distance before a point of danger. In the Netherlands, the interlocking does not require additional locking requirements for this second overlap goal.

This section raises the following knowledge gaps:

1. Which sequence of processes does an interlocking system generally execute to ensure the seven functions of interlocking?

The interlocking system needs input data to allow for the seven functions. A representation of the core data follows from the way an interlocking processes a route request and how it gets engineered.

2. What effects will the future Dutch signaling system ETCS have on the RailML interlocking process approach?

ETCS challenges industry by its intricate and radically different engineering approach over traditional train control systems and as such demand dedicated IT systems (Janssen, 2012).

2.3 System Limitations of Engineering and Designing Interlocking Systems

This section clarifies the gap between current development process and the desired one.

Current Interlocking Engineering and Design Process

The development of an interlocking system usually starts from a specific need to either renew a current system or provide a new built track with signaling system equipment. In the Netherlands, these requests could come from either the ministry of Infrastructure & Environment or a rail operator. The Dutch infrastructure manager Prorail then defines the local rail infrastructure functionality together with other parties like the Dutch Railways and local governments. Prorail then assigns an engineering agency to develop solution strategies that comply with the infrastructure requirements. Prorail selects one strategy for which an engineering agency designs the signaling envi-

ronment in detail. The design a.o. includes interlocking requirements and relevant documents. The main interlocking documents encompass an RVTO (rail design), an OBE (track layout characteristics), an OS (overview of signal aspects) an SVA (additional signaling requirements) and an OR (track circuitry). The OS, OBE and OR concern technical drawings; the RVTO and SVA guidelines come in report form. Another engineering or industrial firm, e.g. Siemens, will then start engineering the IT of the interlocking. Siemens extracts and digitizes the necessary data from Prorail's files into an Element Verbindungs Plan (EVP). An EVP feeds the tools that automatically result in the code for an electronic interlocking system (IXL). When the IT engineering process finishes, both Siemens and Prorail test the interlocking before it actually gets built. After manufacturing, Siemens and Prorail will execute three verification stages to ensure that the hardware, software and interaction with other interlocking goes according to plan. Finally, the system gets installed and tested in real life to validate it against the project definition. Figure 6 shows the entire interlocking process.

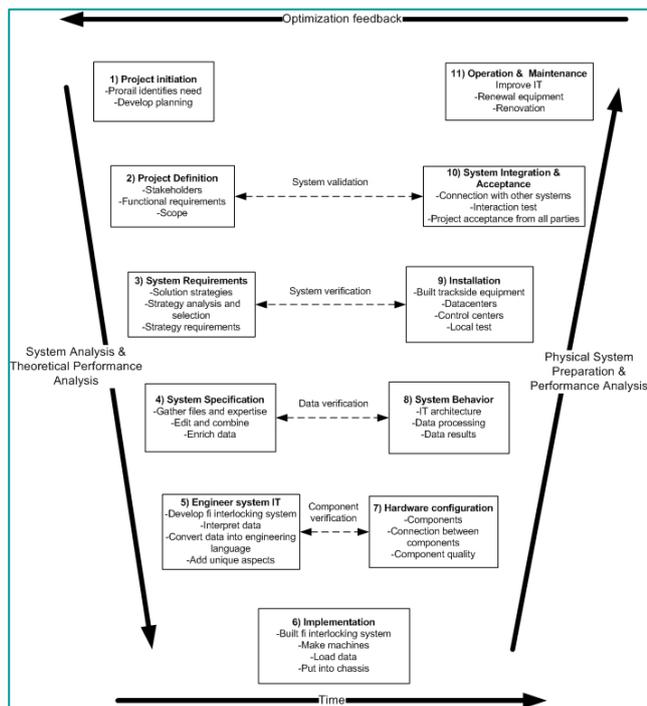


Figure 6: The general interlocking design, engineering and production chain represented as a V-shape.

Process Limitations

The current situation works but lacks the efficiency demanded of railways in this age. First of all, the process is quite cumbersome and opaque (Fries, 2003). Currently there exist various IT rail databases for railway signaling purposes. Mutual arrangements on technical aspects, validation, progress and so on is required which makes project management challenging and some processes ambiguous. Furthermore, due to high safety demands, it is of utmost importance that the infrastructure manager collects complete and correct data. Industry has (yet) few means to check this which increases the chances of errors. Second, partly as a cause of the first point of concern, the entire development process takes a lot of time. In average

cases the process takes about two years. Third, the process is expensive and makes railway infrastructure unattractive. To better explain the situation: interlocking hardware encompasses about 22% and cable cost about 14% of the total engineering design process costs (Janssen, 2013a). This implies that a maximum of 64% in cost reduction can be achieved by adapting the process chain. Fourth, an infrastructure manager desires a familiar system. Multiple countries having their own operational regulations for interlocking systems meaning that the industry continuously needs to alter their systems. Finally, many people, various databases, several validation processes and process iterations decrease the overall product quality and increases the probability of errors. Especially errors in later stages of the development process or after implementation increase costs and affect the safety case significantly.

Ideal Process Situation

Ideally, when an infrastructure manager desires a new signaling system for a certain area, an automated process almost completely designs the interlocking area. Having reliable interlocking area data in a database, the infrastructure manager could ideally just 'push a button' to generate interlocking files required by interlocking engineering companies like Siemens. Although manual adjustment will probably remain a necessity, a more standardized design and machine readable exchange process makes the process chain less cumbersome. Industry, aligned with the standardization, can namely easily convert the file into their own formats for engineering. This would mean a huge time and cost improvement, make the process very clear and reduce the number of involved people drastically. Figure 7 visualizes the difference between both situations. Recall that the probability of errors must be sufficiently low. The entire chain from data production to binary configuration must be rock solid, with minimal human intervention.

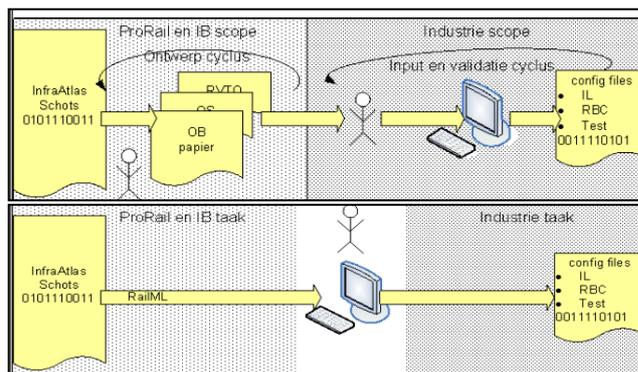


Figure 7: The top process flow shows the current practice. Non machine readable files get prepared by many employees of Prorail and engineering agencies. Siemens then needs to translate those files and before they can engineer an interlocking. The bottom process flow shows the ideal situation: Prorail selects files in RailML that Siemens can directly load into their engineering systems (Janssen, 2012).

2.4 Lean Engineering Design

In the nineties, Womack and Jones (1996) introduced the term 'lean production' for Toyota's innovative motor vehicle production process. Toyota transformed their production processes in the postwar era mainly because of two factors challenging their competitive market position. The

first was a pressure on vehicle sales prices as a result of mass production strategies at competitors in North-America and Europe. The second factor concerned the high costs and long throughput times associated with the production of various vehicle components at a low and fluctuating demand (Ohno, 1988a).

Toyota's production engineer and finally Toyota executive Taiichi Ohno lead the transformation process (Hindle, 2008). Ohno's approach aimed to increase production efficiency by reducing so-called "waste" or 'muda' throughout Toyota's production process; product's timeline from order to payment mainly indicates the performance (Ohno, 1988b). Three characteristics drive Ohno's approach. First, the elimination of eight types of 'waste'. Ohno (1988c) identified seven types of 'waste'; Womack and Jones (2003a) add one final type:

1. Product defects
2. Overproduction of unrequired products
3. Inventories of idle products
4. Redundant processes
5. Redundant movement of people
6. Redundant transport of products
7. Waiting time of products and people
8. Design elements not requested by end user

Second, Ohno's approach considers the entire supply chain rather than just the production within one firm. Tier 1/2/n suppliers and buyers need to be incorporated in order to effectively reduce 'waste'. As an illustration, when optimizing production at one firm, products could eventually still wait or go through numerous loops between firm and supplier. Third, in the case of make-to-stock production, shelve time must be as short as possible while always meeting demand. This is called Just-In-Time (JIT) management.

Womack and Jones (2003a) point out that Ohno's approach differentiates itself from a traditional organization of the production process because it does not focus on batch and queue organization strategies. Furthermore, the lean process optimization vision focuses on removing non-value added activities while traditional approaches focus on cost reduction of processes by means of benchmarking. Since Ohno's approach is not Toyota dependent as well, Womack and Jones concluded that a new supply chain process structure was developed which they called lean production. Lean, because of the tendency to return to the core of unnecessarily expanded supply chains.

Womack and Jones (2003a) define lean processing from a value leverage perspective supported by the relation between own chain, supply chain and demand chain (W.W.A. Beelaerts van Blokland, Santema, & Curran, 2010):

- a focus on **value added activities** i.e. aspects that a customer is willing to pay for;
- an **elimination of 'waste'** processes by identification of the value streams;
- a **flowing**, i.e. minimum inventories and waiting times, zero 'waste' process;
- a customer **pulled** design and delivery;
- an institutionalized mindset to continuously **perfecting** the chain by searching for 'waste'.

The Lean Enterprise Institute (2009), a non-profit think tank founded by Womack to spread the lean approach, states that lean production is not limited to manufacturing and applies to every kind of process. This is especially useful in the light of this research project since we consider a design process. For the sake of this research, 'lean production' will therefore be called 'lean engineering'.

The lean approach can be put into perspective with the agile supply chain process structure (Ludema, 2012). An agile supply chain aims to support radical short-term changes in supply and demand. Consider a computer manufacturer as an illustration. When a consumer desires a computer with a processor just available on the market, an agile supply chain provides the opportunity to directly modify and deliver such a computer. This in contrast to a lean supply chain that focuses on a stable consumer demand in terms of product functionality. In addition, Mason-Jones, Naylor, and Towill (2000) found that a lean strategy gains a competitive advantage based on mainly low costs: an agile strategy gains an advantage on the basis of a high service level. Furthermore, a lean structure differentiates itself from agile by commodity type products, long product life cycles, low profit margins and a high share of fixed costs in PPE. In short: a lean supply chain focuses on a predictable market with low variety in products while an agile supply chain does the reverse.

This section raises the following knowledge gaps:

3. How can an (interlocking) engineering design process become lean?

Companies barely applied the lean philosophy in engineering design environments. Therefore, a new methodology needs to assess how and if a lean transformation applies in engineering design processes.

4. To which extent can a lean approach standardize Siemens' engineering design processes?

An important cornerstone of a lean approach is standardization of products. Siemens has interlocking customers across the world which, at first sight, limits the possibilities for far-stretching standardization.

2.5 RailML

Industry and infrastructure managers believe that a shared data platform will make the engineering process less comprehensive (IVU Traffic Technologies AG, 2010; Janssen, 2012; Koelewijn, de Rijk, & Dragt, 2013; Schut & Dragt, 2013). They aim to align their systems and protocols to shorten the engineering design process time and costs. The idea: a standardized data exchange tool that infrastructure managers use to convert their rail data to a widely adopted format in the industry of interlocking engineering. The parties probably got their idea from Enterprise Application Integration (EAI) or Electronic Data Interchange (EDI) tools (Reuter & Rohde, 2007; Stadler, 2007) used in various industries but rail. In short, EAI provides a database and planning system for the entire chain; EDI enables standardized exchange of data across parties in a chain. The EDI tool ODETTE for example, ensures data exchange of engineering data in the automotive industry (ODETTE International Ltd., 2013).

The data and IT tools that the rail industry desires to connect concern those involved with design, e.g. interlocking area maps as OBE and OS, engineering, e.g. the interlocking IT architecture in EVP, rail network databases, e.g. InfraAtlas, and applications, e.g. BITS for validation purposes (Figure 8).

In the INtegrated European Signaling System (INESS) project, Linder and Grimm (2012) studied various data exchange tools and prove that RailML has most potential. The authors show that RailML currently features most functionality and a panel of fourteen experts score it best based on a wide range of criteria.

RailML is an open source initiative with the support of various rail industrial parties like Siemens, engineering agencies, universities, software developers, railway operators and infrastructure managers (RailML.org, 2013b). The RailML consortium choose XML as the data exchange language thanks to its transparent and clear structure. Furthermore, XML proved itself as a very decent data exchange application in various IT applications (Fries, 2003; Hürlimann & Krauss, 2003).

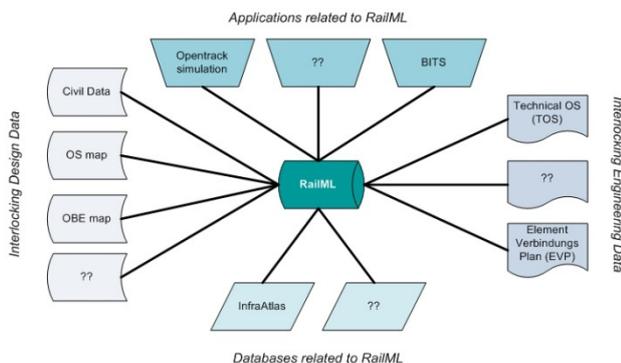


Figure 8: RailML aims to become the standardized interface that enable a smooth exchange of data between databases like InfraAtlas, applications like Vlsim (PTV AG, 2011) and design engineering data related to for instance interlocking.

The RailML structure currently includes infrastructure, timetable and rolling stock (RailML.org, 2013a). A vital part that still misses is interlocking. The interlocking development process has been started fall 2012 and in spring 2013 an agreement has been reached on the object relations that an interlocking must comply to (Appendix A). Yet, the RailML consortium has not agreed on a formalization (Quaglietta, 2013).

This section raises the following knowledge gaps:

5. How can interlocking be formalized within the RailML modeling language?

The current version of RailML, version 2.2, does not include all elements necessary for interlocking engineering design.

6. How to connect or convert IT tools, a.o. InfraAtlas, to RailML?

Parties in the chain use various data sources to design interlocking areas. RailML thus needs to communicate with those sources to retrieve the necessary data as well. A general formalization needs to be found that a.o. excludes conflicting data, provides all necessary data, has a simple structure etc.

7. How to keep the interlocking formalization of RailML standardized?

Although this research focuses on the Netherlands, the RailML nature is multi-country. This implies that the developed formalization should also be applicable for other railway organizations.

8. Which data elements can RailML not exchange in order to produce interlocking equipment?

RailML might very well prove inadequate to provide certain types of data. Those inadequacies need to be mapped.

9. How should RailML (ideally) be designed such that it supports a lean engineering design of rail interlocking systems?

RailML potentially enables the reduction of 'waste' during interlocking design. The question rises which contribution RailML could deliver to a lean design, how to develop its system architecture and which parties to involve.

2.6 The Interlocking Engineering Design System in a Nutshell

The previous sections clarified the current status of the main elements that interact as visualized in Figure 2. Appendix B contains a system analysis to identify actors, performance indicators, main improvement strategies and external factors of importance. Both provide a complete picture of the system as depicted in Figure 9.

An actor analysis identified that Prorail and Siemens desire RailML for somewhat conflicting goals. Siemens aims to become more competitive by lowering process costs while delivering improved interlocking quality. Prorail aims to reduce signaling costs while excluding surmountable train conflicts. RailML will probably reduce the probability of design errors and might even allow for a more efficient determination of signaling strategies by modeling. This covers the quality and surmountable train conflict goals. Prorail's interlocking price aim and Siemens' process cost goal might prove a trade-off: when RailML simplifies the process and allows for more competition, the process cost gains might be compensated by competitive pricing.

Another important group of actors are currently not actively involved, but have the means to block strategies from implementation. Loxia, manager of the Dutch rail network database InfraAtlas, the dedicated ministry and the design supplying engineering agencies have those means. Siemens and Prorail should not raise too much resistance and involve them in the entire process.

Four performance indicators lead to wide acceptance and success: interlocking design costs, design throughput time, amount of ambiguous activities and amount of process interdependencies. Siemens and Prorail clearly share the performance indicator of interlocking design costs. Furthermore, both Siemens and Prorail desire to reduce the amount of ambiguity in activities and interdependencies between processes. Siemens finds those goals important to improve the product quality versus cost ratio. Prorail mainly pursues those goals to improve operational safety which is a derivative of data reliability. In addition, data reliability is a prerequisite for an optimal track layout as well; maybe leading to more track availability, sustainable use of resources and improved maintenance. The

fourth performance indicator represents the duration of interlocking development as the throughput time. As a result, Siemens mainly benefits by becoming more competitive; Prorail aims for swift corrective track actions. At first hand, Siemens can take four general measures to improve the performance indicators: cutting process cost, lowering customer prices, broaden the product portfolio and/or limiting the product portfolio. Simply cutting costs however goes hand in hand with a decrease in quality. The effect of a price cut depends on the market reaction: more turnover might improve the process although a price war puts quality under pressure. A wider portfolio of interlocking products increases product complexity and worsens product quality goals. In contrast, limiting the product portfolio means that either an infrastructure managers specific wishes cannot be met and/or result in fewer competition. These four general measures result in undesirable trade-offs. A structural process change like RailML, will probably positively affect the performance indicators on the basis of the qualitative assessment in this section. The level of competition remains a factor of concern. Some competitors currently support the development of RailML; others work on alternatives (Rio, Rasoamana, &

Dumont, 2012). RailML should therefore develop a structure that all competitors can easily adopt.

This section raises the following knowledge gaps:

10. How will RailML effect the performance indicators? Parties need to invest in changing around the current engineering design process. The question focuses on what the precise (cost) benefits of RailML are.

11. What role do competitors play in the development and use of RailML?

This study develops an interlocking formalization in cooperation with Siemens and Prorail. Prorail however desires a RailML that can easily be used by competitors too. This poses demand of the IT architecture.

12. How lean is RailML?

Can RailML meet the expectations of a lean engineering design chain for interlocking systems, or are more / other tools and policies necessary?

13. What are the potential limitations to implementation of RailML?

The conditions that apply for RailML to exert its potential.

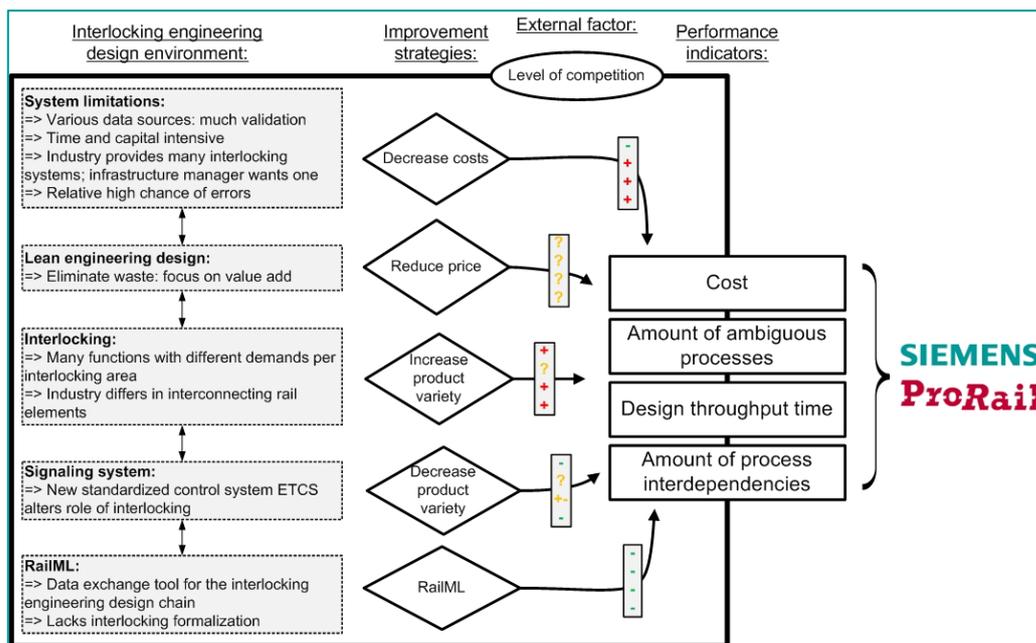


Figure 9: The system diagram presents the general improvement strategies both Prorail and Siemens can take to make the interlocking chain more efficacious. The boxes with four signs indicate their estimated effects on the performance indicators. A + means that the indicator increases, - means that it will stay about the same, a - means that it will decrease and a ? mark means that the effect could hardly be estimated.

2.7 Research Questions

This section shows how the knowledge gaps mentioned in previous sections lead to the research structure of this thesis (Table 1).

Sub Research Question 1:

- Which methodology leads to a lean transformation of the engineering design process for rail interlocking?

Gap #	Knowledge gap description
3	How can an (interlocking) engineering design process become lean?

An exploration of existing sources about lean value chains, lean transformation examples, general transformation strategies and so on, provides the fundament for a lean transformation methodology of the interlocking engineering design process. The sources that this thesis uses mostly include scientific journals, websites, reports, interviews, expert opinions and company statements

Sub Research Question 2:

- How would a future lean interlocking engineering design change the status quo?

Gap #	Knowledge gap description
4	To which extent can a lean approach standardize Siemens' engineering design processes?
9	How should RailML (ideally) be designed such that it supports a lean engineering design of rail interlocking systems?

The methodology developed at the first research question is applied here. The methodology at least needs to map the current value chain, identify and mathematically prove bottlenecks and result in a lean process structure for benchmark purposes. Conceptual mapping techniques and analytical / statistical programs will be used..

Sub Research Question 3:

- How to formalize interlocking into RailML for the purposes desired by Prorail?

Gap #	Knowledge gap description
1	Which sequence of processes does an interlocking system generally execute to ensure the seven functions of interlocking?
2	What effects will the future Dutch train control system ETCS have on the RailML interlocking process approach?
5	How can interlocking be formalized within the RailML modeling language?
7	How to keep the interlocking formalization of RailML standardized?

This research question focuses on XML modeling to derive a new interlocking schema in the RailML language.

Sub Research Question 4:

- How does RailML change the interlocking engineering design chain when implemented as the rail data exchange tool for railway signaling systems in the Netherlands?

Gap #	Knowledge gap description
6	How to connect or convert IT tools, a.o. InfraAtlas, to RailML?
8	Which data elements can RailML not exchange in order to produce interlocking equipment?
11	What role do competitors play in the development and use of RailML?

This research question aims to identify how RailML changes the current state by conceptual process modeling. RailML's ability to include interlocking data from research question 2 reveals which of the current processes RailML can eliminate or improve.

Sub Research Question 5:

- What benefits does the application of the RailML formalization have during the engineering design of an interlocking area over the traditional and lean process structure?

Gap #	Knowledge gap description
10	How will RailML effect the performance indicators?
12	How lean is RailML?
13	What are the potential limitations to the implementation of RailML

This research question finds out which performance improvements RailML will deliver compared to the current state by means of simulation. Furthermore, simulation also benchmarks RailML on its lean effect.

Table 1: Overview of the research structure.

Topic	#	Sub Research Question	Method
Lean Engineering	1	Which methodology leads to a lean transformation of rail interlocking systems' engineering design process?	Literature, experts interviews
	2	How would a future lean interlocking engineering design change the status quo?	Expert interviews, value stream mapping, data / statistical analysis
RailML / System Engineering	3	How should interlocking be implemented into RailML for the purposes desired by Prorail?	XML modeling, literature
	4	How does RailML change the interlocking engineering design chain when implemented as the rail data exchange tool for railway signaling systems in the Netherlands?	Literature, process analyses, XML modeling
Performance Analysis	5	What benefits does the application of the RailML formalization have during the engineering design of an interlocking area over the traditional and lean process structure?	Simulation, data analysis

Figure 10 shows which chapter deals with what research question. Finally, a conclusion provides the answer to the main research question. The figure structure corresponds to a system analysis approach in Figure 2 and Figure 9. Possible transformation directions (boxes with a question mark) to achieve a lean state need to adapt the current engineering (boxes outlined in blue) design (boxes outlined in red) process. RailML might provide a significant role in this lean transformation. The effectiveness of RailML largely depends on its compatibility with various signaling systems. The performance indicators score the process improvement of a (conceptual) lean approach and a process with RailML; a judgment on the degree in which RailML can make the engineering design chain lean.

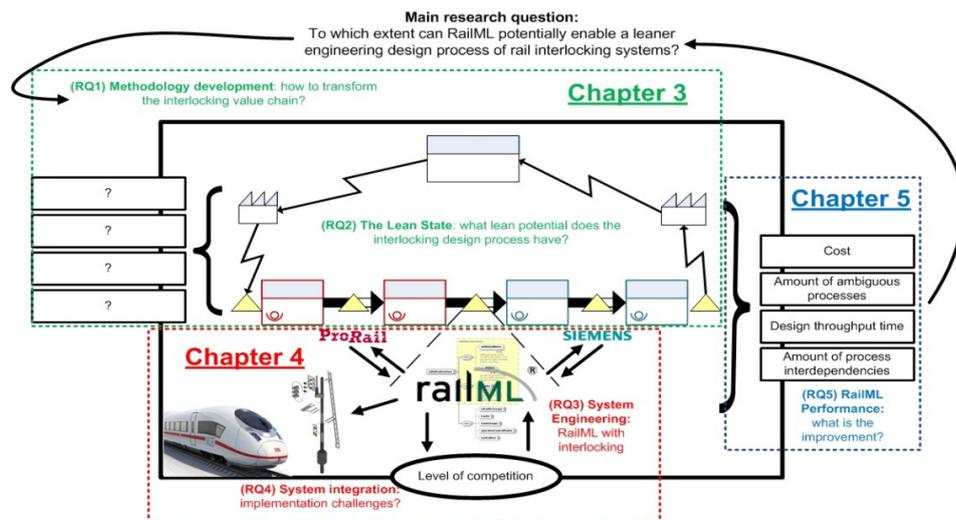


Figure 10: The thesis' research structure visualization. Each colored boundary indicates a chapter that deals with answering one or two research questions.

The next chapters address the research questions in three streams:

Chapter 3: A lean benchmark for engineering design processes (RQ 1 and 2)

A literature study into lean engineering results in a lean transformation methodology that makes evaluation of the status quo possible. Accordingly, the methodology is used to mitigate 'waste' and derive a lean state of the engineering design process for benchmark purposes.

Chapter 4: RailML development for the purposes of interlocking engineering design (RQ 3 and 4)

A case study develops a RailML v2.2 for an actual interlocking area. On the basis of that and an (RailML) interlocking literature study, a RailML interlocking formalization is proposed.

Chapter 5: RailML's improvement of the interlocking engineering design process (RQ 6)

Process analysis aims to investigate how RailML changes the interlocking engineering design chain. The benefits and disadvantages of RailML with the interlocking formalization for the chain are investigated by quantifying the performance indicators on the basis of a simulation study. Furthermore, the effects of RailML are offset against those of the lean benchmark to reveal how lean RailML is, in which performance field additional measures are required for a lean state and whether lean works for engineering design chains in general.

Chapter 6: Conclusion & Recommendations

Provides the answer to the main research question.

2.8 Academic and Business Contribution of this Thesis

The academic contribution is threefold:

1. this study researches the field of RailML, investigate its potential and provide contributions to its current architecture. This study especially investigates the field of interlocking in relation with RailML and strives

2. to make a working RailML model with interlocking tested on a interlocking area. Interlocking has not yet been formalized in RailML;
2. this study derives conclusions on the realizability and the effectiveness of RailML as standardized approach for the European railway industry. This study identifies the challenges involved to convert the Dutch railway network in the RailML model and the possibly needed extensions. Those results form a benchmark for or can be benchmarked against RailML transitions in other countries;
3. this study develops an approach to make (rail) engineering design processes lean. It also developed a performance measurement technique within that respect. The combination and performance of both form an important fundament for further development of lean design processes.

The business contribution arises at both Prorail and Siemens:

Siemens: The result of this new interlocking development strategy could lead to a quicker engineering design process at higher reliability and with fewer compatibility issues. When this leads to a competitive advantage, Siemens could gain market share by winning more tenders. It depends on company strategy whether this will also lead to higher profits: e.g. mutations, maintenance contracts, profit margins etc.

Prorail: The infrastructure manager is very interested in the RailML strategy as it for one leads to a much simpler, quicker and reliable process to gather the required data for rail projections. Furthermore, the standardized approach increases competition with consequent lower prices for railway signaling systems at higher quality. In addition, RailML will allow Prorail to introduce the radically different control system ETCS in a quick and well controlled manner.

3. A Lean Benchmark for Engineering Design Processes

This chapter defines the possible lean state of the interlocking engineering design chain. This lean state functions as a benchmark to assess the lean performance of RailML. For that purpose, this chapter first develops a lean transformation methodology for the engineering design of rail interlocking systems. Section 3.2 starts applying the methodology by stating the requirements of interlocking systems in the Netherlands that a lean state should meet. Section 3.3 conceptualizes the process structure of the current state and identifies the ‘waste’ that prevents lean engineering design. Section 3.4 then defines how to mitigate that ‘waste’ and achieve a lean engineering design process structure.

implement transformation strategies in an engineering design environment. Additional research compares various modeling and performance measurement techniques in order to derive a methodology suited to make engineering design process of interlocking systems lean.

3.1.1 Lean Transformation Directions Based on Best Practices

The relative short history of lean thinking and the conservative nature of the rail industry make it that best practices in the field of rail lean engineering design cannot be found. Therefore, comparable cases are sought in the field of transportation and lean engineering design processes. The main goal is to provide insight that a lean engineering approach might also benefit the rail sector. Best practices of lean transformation in transportation can widely be found in aerospace and automotive sectors. In addition, one shipping example is mentioned. Appendix C provides an overview of the types of ‘waste’ discussed in each paper.

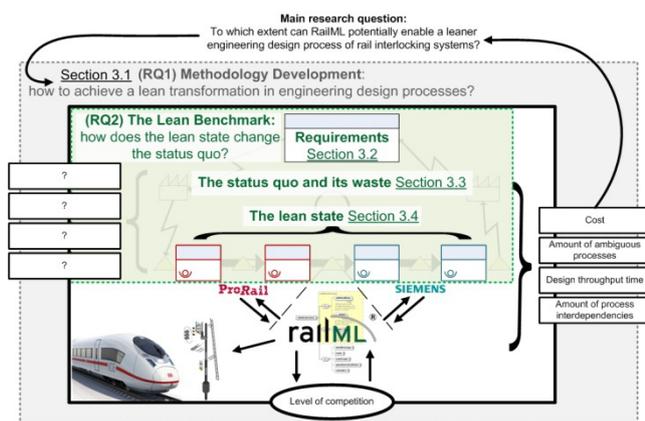


Figure 11: The outline of chapter 3 which provides answers to the first two research questions.

3.1 The Lean Transformation Methodology for Engineering Design Processes

An exploration of lean transformation projects in the past, provides directions to transform the current interlocking process as a start. In addition, those lean best practices form the fundamentals of a framework on how to

Beelaerts on Aircraft Production

W.W.A. Beelaerts van Blokland et al. (2010) investigated Eurocopter and Boeing as examples of aircraft manufacturers that implemented lean methodologies into their production system. Three major global trends made such aircraft manufacturers look into lean solutions: increasing levels of competition at manufacturers and operators, a business consolidation trend in this sector and maturity of main aircraft components and technology. In order to become leaner, the production of supply chain partners got more interwoven which made collaboration and knowledge sharing the key success factors. The aircraft manufacturer becomes the so-called lean integrator, leveraging value among partners and customers of the supply chain. The lean approach provided Eurocopter and Boeing with the means to improve the management of their complex aircraft design and production. Furthermore, the approach stimulates innovations.

Womack on Porsche

Womack and Jones (2003b) studied the transformation of a traditional German engineering agency into a lean production firm at Porsche. Porsche developed a various range of high end cars since WW2. The main characteristics of such a German engineering agency were superior engineering quality and ditto skilled staff, management’s vision that the actual product is most important for beating the competition and a stable long term financial and supplier relationship focus. This model however, almost made Porsche go bankrupt in the nineties due to the introduction of cheaper and higher quality Japanese cars produced in a lean way. Porsche’s main problem was a too strong focus on the product quality which resulted in huge

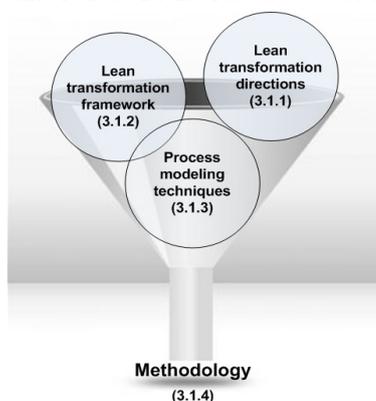


Figure 12: The outline of the literature analysis in this chapter.

inventories, many redundant quality checks, bad supplier management, chaotic work process, late deliveries and so on. A new lean approach focused on a flowing process in which suppliers deliver when Porsche needs it, engine parts are provided exactly at the right time with the right amount at the engineering booths, these construction processes are standardized and require fewer control and a clear market is targeted. In five years time, Porsche doubled its productivity rates, cut part defects by 90%, cut first unit delivery time by 55%, reduced supply lead time from weeks to three days and cut NVAW size by 90% in the amount of elements.

Kolich on Shipyards

Kolic, Fafandjel, and Zamarin (2012) investigated the production line of steel plates for ships. The main reason for their research was that European producers ought the production of a certain range of ships unprofitable, while their Asian counterparts did not. The main difference was a dramatic difference in the productivity delivered per manhour. Kolic et al. proposed that European shipyards need to get leaner by most foremost one-steel-piece flow across the entire production line. This approach resulted in fewer plate turns, necessary welds and quality checks of the welded steel plates. Simulation and case studies pointed out that the performance of the new production set-up would reduce total manhours by 30%.

Hallam on Aircraft Production Supply Chain

Hallam (2010) investigates the effects of a leaner supply chain for the production of defense aircraft to reduce the chance of growth in program cost and planning risks while stimulating innovation. The author tests the hypothesis that cumulative delays throughout production are the main cause for additional costs and a tight planning. In terms of the first aircraft delivery in a production series, lead time can, in combination with a design authority, be reduced 40% to 70% with serial or parallel design changes during production respectively. The design authority especially proves useful to mitigate the productivity loss as a result of work stops when changing design. Besides a reduction in lead time of the first aircraft, the overall production capacity in the chain increases.

Marzouk on Lean in Design Processes

Marzouk, Bakry, and El-Said (2012) researched how the design process of various consultancy firms in the field of construction engineering can become leaner. They noticed that the design process is currently a highly undervalued part of the entire supply chain and, as a result, could perform much better. Most important bottlenecks appeared poor communication, inadequate knowledge, redundant processes, insufficient planning, insufficient information management and a suboptimal allocation of resources. The authors developed a lean design process proposal that reduced uncertainty, reduced waiting times and increased the ease of information transfers. Their model proved that the total activity utilization increased by 39% and average activity utilization by 40%. Furthermore, the amount of clear process bottlenecks had been reduced from three to two and were positioned further downstream.

In the absence of lean engineering design in the rail industry, Appendix C illustrates that lean best practices in either transportation or design process management partly touch rail engineering design cases. Figure 13 illustrates that rail engineering design processes may benefit from (some of) those transformation directions and approaches as well.

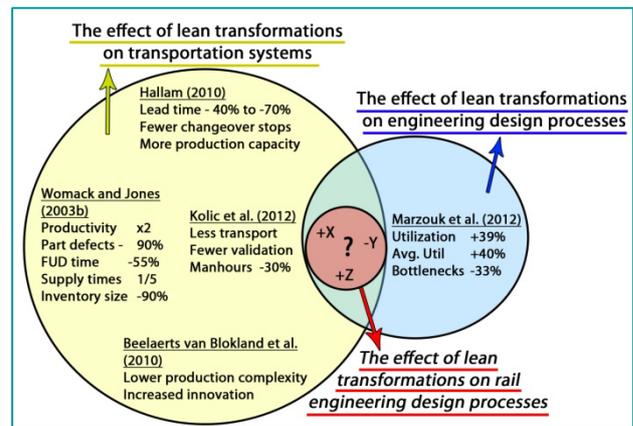


Figure 13: A visualization of the main best practice findings. The yellow circle illustrates lean literature in the field of transportation; the blue circle illustrates lean literature in the field of engineering design processes. The red circle shows the field of engineering design in rail transportation.

The potential transformation directions distinguish themselves in their ability to sustain their lean value chain approach (Table 2). The impact of a transformation direction on each aspect of the lean definition as defined in section 2.4, defines its level of lean sustainment. The remainder of this subsection elaborates on the main differences in qualitative lean performance, Appendix C provides a more detailed overview.

Value decomposition strongly focuses on prioritizing processes based on the type of value they deliver to a customer. Value decomposition however touches mostly internal rather than external processes of the chain; weakening the pull strength. Prioritizing 'waste' reduction when transforming the chain improves all lean aspects which benefit from a lower degree of the different 'waste' types as defined in section 2.4. Again, an internal focus weakens pull although improved flow lowers delivery times and makes pull score indifferent. An open source network cooperation initiative enables a lean initiative that both internal as external parties join. A clear market focus is an external optimization approach that does ensure improved internal processes. The productivity and risk balance score well on meeting customers' delivery expectations although it does not specifically result in goods that customers desire more.

A standardized engineering approach tries to bring back product variety to a core business process aligned with the main customer and company goals. Although not delivering to specific customer wishes, a much simpler and predictable work process is achieved. A concurrent engineering design approach improves product processes locally, where product knowledge exist. This approach is very effective in generating products that fulfill customer needs at the cost of process interruptions thanks to sudden improvements. A consolidated data approach ensures com-

plete data and reduced waiting times but diminishes the accessibility to optimize product design. Modular production decomposes the production process into a clear amount of steps which simplifies management, control and optimization of the processes. One potential disadvantage might however arise with too many modules; resulting in non value add. A zero inventory engineering approach focuses on continuous flow of goods, although it does not directly focus on product improvement itself. Indirectly, zero inventories could lead to fewer design errors and inconsistencies by an increased focus of the employee. Furthermore, when employees only start a new project after finishing another, data version management could improve as well.

The process management improvement by means of a lean integrator relies on the support of a lean "guru" to implement lean process strategies while ensuring that no new bottlenecks occur. A design authority does the same though focusing on choices regarding product aspects. Sharing revenues among partners aims to prevent cherry picking and stimulates chain wide initiatives to reduce 'waste' and improve competitiveness. A drawback is that a

focus on revenues is a strong internal performance measure, thereby surpassing customers' desires. A somewhat different approach shares information to achieve a quick adoption of innovations.

A critical note concerns data reliability (Palacios, 2013). Palacios argues that a concise and standardized data flow with clear control is good, but two issues will always remain: data interpretation and version management. Palacios proposes visualization to solve data interpretation issues. Version management is more complicated and Palacios stresses to always expect data version conflicts; especially when multiple parties work on the same project. In the end, stale data is deadly.

A threshold score of 20 points would leave five high potential transformation directions as shown in Table 2, that increase the chance of a lean transformation that lasts. A beneficial addition is that the five transformation directions supplement each others' weaknesses: e.g. 'waste' prioritization scores relatively low on pull while sharing information scores high.

Table 2: Qualitative impact assessment of the potential strategies, found in the literature of section 3.1, to transform the current state of the engineering design process. The scale ranges from 1 (weak sustainment of lean strategy) to 5 (strong sustainment of lean strategy).

Directions to transform a traditional rail engineering design process into a lean process		Value added activities	Elimination Flow of 'waste'	Pull	Perfection	Total score
Transformation focus	Value decomposition (Marzouk et al., 2012)	5	4	3	4	18
	'waste' priorities (Marzouk et al., 2012)	4	5	4	4	20
	Open source network cooperation initiative e.g. RailML (W.W.A. Beelaerts van Blokland et al., 2010; Marzouk et al., 2012)	5	4	4	5	22
	Clear market focus (Womack & Jones, 2003b)	4	3	5	3	18
	Productivity and planning risk balance (Hallam, 2010)	2	5	3	5	17
Engineering approach	Standardized interlocking components, procedures and deliverables (Kolic et al., 2012; Womack & Jones, 2003b)	5	4	5	4	21
	Concurrent design decisions (Hallam, 2010)	4	3	2	5	16
	Consolidation of data processing (Kolic et al., 2012)	3	4	5	2	17
	Modular process (Kolic et al., 2012)	3	5	4	5	21
	Zero information inventories (Marzouk et al., 2012)	3	4	5	4	19
Process management	Lean integrator (W.W.A. Beelaerts van Blokland et al., 2010)	4	5	4	3	19
	Design authority (Hallam, 2010)	5	4	4	3	19
	Sharing revenues among partners (W.W.A. Beelaerts van Blokland et al., 2010)	4	4	2	4	18
	Sharing information among partners (Marzouk et al., 2012)	4	4	4	4	20

3.1.2 The Transformation Framework

This subsection presents a comparative analysis of lean transformation framework literature, reviewed in Appendix C. A selection of five studies forms the basis for the developed and applied transformation framework. The five studies enrich the framework because they complement each other with slightly varying lean transformation approaches. Table 3 presents how those five studies result in one general aggregated framework for engineering design processes.

The methodology starts with a definition of the system requirements to describe the system boundaries that an eventual process chain must complement to. Engineering a rail interlocking system, this step should typically result

in route related dependencies. The second step, conceptualization, aims to catch the current process structure and their input, output, support and required information. Furthermore, the conceptualization attempts to produce a process map to attain a structural overview.

The third step encompasses the identification of 'waste' that weakens the lean performance of the process chain. This implies that the degree of 'waste' types (section 2.2) will be determined.

The fourth step of the framework develops a lean benchmark to mitigate the 'waste' discovered in the previous step.

The fifth step engineers the lean tool in question, like RailML, and the effects on the process structure will be conceptualized.

The sixth step comprises modeling to determine the relative performance of the current and engineered system process structure (i.e. RailML in this case) compared to the lean benchmark.

The seventh step analyzes the results of the previous step by means of weighted comparison on the performance indicators.

The eighth step needs to design RailML such that it mitigates the undesired effects compared to the lean benchmark and status quo. This step equals implementation of the RailML strategy according to the desires of the parties involved.

Table 3: The generation of a transformation analysis approach on the basis of five methodologies from literature.

Lean-pull system implementation (Lu, Yang, & Wang, 2011)	Concept Development Process (Jeong & Phillips, 2011)	Lean manufacturing strategy (Dotoli, Fanti, Iacobellis, & Rotunno, 2012)	Dynamic simulation models (Agyapong-Kodua, Ajaefobi, & Weston, 2009)	Lean engineering design processes (Marzouk et al., 2012)	Step	Aggregated framework
	Analyze customer needs				1	Requirements Define process boundaries
	Define customer specification					
Mapping of value added activities		Develop as-is activity diagram		Develop activity flowchart	2	Conceptualization Describe the current state of the engineering design system
		Develop as-is state map		Classify variety of project types		
			Identify 'waste'		3	'waste' analyses Identify lean suppressing 'waste' factors
			Prioritize process bottlenecks			
Describe optimal lean-pull system	Generate concepts				4 / 5	Lean State A lean benchmark that mitigates 'waste' factors. System Engineering Design RailML
	Concept selection	Develop to-be state map				
		Develop to-be activity diagram				
				Built simulation model	6	Modeling Quantitative study
Simulation	Concept test	Simulation study; if insufficient, return to to-be state map	Simulation	Change simulation parameters to derive lean solution		
Multi-Criteria Analysis			Result evaluation	Result comparison	7	Evaluation Selection of the future state strategy
Design desired future state of the system	Specify final design				8	Detailed Design Overcome the gaps to make RailML lean

3.1.3 Modeling Techniques for Engineering Design Process

The conceptualization, 'waste' analyses, lean state, modeling and evaluation steps in the framework of Table 3 require techniques to map the current and future states of the system. This subsection compares commonly used conceptualization and quantitative techniques used in lean literature.

Table 4 therefore scores various conceptualization techniques on their ability to discover each of the eight 'waste' types of section 2.4. This subsection first qualitatively elaborates on the fit of conceptual techniques for a lean transformation of engineering design processes, Appendix C provides a more detailed overview.

A value stream defines all required actions to design and produce a product. Value Stream Mapping (VSM) maps the value chain and is used for lean transformations; especially in an early design state (W. W. A. Beelaerts van Blokland, 2013). VSM allows for a clear visualization of design activities' characteristics and interdependencies, the in-between inventory or transport data and the overall planning requirements and static throughput data (Rother & Shook, 2009). Furthermore, VSM provides the possibility to depict lean process concepts. Therefore, VSM is especially strong in identifying the (non) value-added time and resource occupation of design activities. The technique however lacks a decent possibility to show the real necessity of activities to achieve the final product. In addition, Lu et al. (2011) remarks the difficulty to compare future value chain structures on effectiveness.

Jeong and Phillips (2011) addressed the disadvantages of VSM by combining the technique with discrete event simulation. This allows for the possibility to validate the VSM, determine the effects of a parameter or structural change and perform statistic, routing and queue analysis.

Dotoli et al. (2012) apply the Unified Modeling Language (UML) to describe the relation between the main process activities and their assets or actors. Therefore, a UML comparison between current and possible future state engineering designs gives a clear representation of the activities that add value to the final product. UML contains few metrics though which makes further application limited.

Computer Integrated Manufacturing Open System Architecture (CIMOSA) connects a top-level index diagram, interaction diagram, structure diagram and activity diagram (Agyapong-Kodua et al., 2009). CIMOSA aims to decompose and visualize activities as a network of dependent processes. The technique therefore scores well on identifying the activity structures that do not significantly add value. A big disadvantage of CIMOSA is that it is mostly a qualitative tool.

A Supply Chain map shows the activity and inventory relation to produce or design a product and which supplier takes responsibility for the activity (W.W.A. Beelaerts van Blokland et al., 2010; Stadler, 2007). The supply chain map thus clearly represents the (transport) flows between suppliers. A lack of further information and data makes it hard to estimate other lean 'waste' besides those related to transport from supplier to supplier.

A value chain map visualizes a network of suppliers, often compared to a focal company, and shows product's value adding relations between suppliers (W.W.A. Beelaerts van Blokland et al., 2010). The technique clarifies some of the consequences regarding complexity as design time and non value add. Although, just as with Supply Chain mapping, the technique lacks information and data for further analyses.

The Supply Chain Operations Reference (SCOR) model developed by the Supply Chain Council is a reference

model to describe supply chains in a general way (Supply Chain Council, 2010). The SCOR describes supply chains by means of a comprehensive list of categorized metrics. The SCOR model defines five main metric groups: supply chain reliability, responsiveness, agility, costs and asset management. The SCOR model is not an optimization tool; merely describes the current status of a supply chain by means of time, utilization, delivery, supply and cost performance metrics. The model therefore does not show direct conclusions on the redundancy of certain processes. The IDEF0 process technique aims to standardize the gap between conceptualization and calculation models by means of a hierarchical process description (Mahfouz, Shea, & Arisha, 2011). IDEF0 decomposes the main process of the chain into a 'tree' of sub processes until no further refinement can be made; each process explained by its input, output, information/data support and resources. The technique can therefore provide a lead in determining an effective process structure. IDEF0 however misses the data to emphasize the 'waste' aspects that follow from characteristics as time, costs, counts and so on.

Value Stream Mapping in combination with Discrete Event Simulation (DES) performs best in the comparison analysis and is relatively straightforward to implement. This approach therefore forms the main tool for finding 'waste'. The VSM with DES approach scores however average on measuring product defects, redundant processes and non value added processes. A design process does not really face product defects, that is more related to production chains. UML and IDEF0 will fill the conceptualization gap of redundant processes and non value added processes. The combination of techniques is chosen over CIMOSA because they require substantially less information and are easier to interpret. In addition, IDEF0 provides easy translation to a simulation model. Furthermore, the author of this thesis has experience with UML and IDEF0 modeling.

Table 4: Qualitative assessment of conceptual techniques to support a lean transformation. A 1 means that a technique does not or barely contributes to identify the specific type of 'waste', a 5 means that the technique can identify the type of 'waste' very well.

	Product defects	Over-production	Idle product inventories	Redundant processes	Redundant movement	Redundant transport	Waiting time	Non value add	Score
Value Stream Mapping (Agyapong-Kodua et al., 2009; Jeong & Phillips, 2011; Lu et al., 2011; Rother & Shook, 2009)	1	3	4	2	2	3	4	3	22
Value Stream Mapping integrated with Discrete Simulation (Jeong & Phillips, 2011)	3	4	5	2	4	4	5	3	29
UML activity diagrams (Dotoli et al., 2012)	1	1	2	3	1	2	1	5	16
CIMOSA (Agyapong-Kodua et al., 2009)	1	1	2	4	2	2	2	4	18
SC map (W.W.A. Beelaerts van Blokland et al., 2010)	1	1	1	1	1	3	2	1	11
Value Chain map (W.W.A. Beelaerts van Blokland et al., 2010)	1	1	1	2	1	2	2	2	12
SCOR (Surie & Wagner, 2008)	4	3	3	2	1	2	4	1	20
IDEF0 (Mahfouz et al., 2011)	1	1	3	4	2	2	1	2	16

Besides conceptual modeling, the modeling and evaluation steps in the framework of Table 3 require a quantitative tool for quantitative conclusions on process structure's performance.

Hoogendoorn, Bliemer, and van Nes (2007) evaluate various analytical tools for traffic flow analysis based on a set of seven characteristics. Five of those characteristics also apply in non traffic flow related situations: the level of detail, the computation type, the type of time dimension, the existence of equilibriums and the calculation method. In addition, Siefer (2008) distinguishes a sixth characteristic: processing technique. The level of detail scales a process' smallest entities into micro, meso or macro. The computation type is either deterministic or stochastic. A time dimension can play a role in the calculations or not. A model that represents equilibriums always results in an optimal set of variables. The calculation method can either be analytical or simulation. Last, processing can go synchronous/time-driven or asynchronous/event-driven. These six characteristics classify the various analytical modeling solutions found in literature (Appendix C) and summarized in Table 6.

Mahfouz et al. (2011) study the use of simulation with relation to a lean transformation. They discovered that a business would find simulation especially useful to determine the performance of a future value chain when elements of queuing theory, statistical distributions and prioritization rules play a role. Interlocking engineering involves statistical distributions to for example determine chances of rework, errors and project size. Queuing theory arises at various design processes where intermediary data needs to wait for a resource to be processed. Even when queuing theory nor statistical distributions would not seem evident, a simulation package would still be a very straightforward way to compute the performance of process interdependencies. Furthermore, a simulation package often allows for animation which makes communication transparent. ARENA has most potential as a simulation package

for interlocking engineering design processes because it does not merely focus on production processes like Extendsim and Simul8. Furthermore, ARENA allows for nearly unrestricted performance measurement which Flexsim and Powersim do not. In addition, ARENA offers, in contrast to TOMAS, an accessible GUI and modeling language the author is familiar with.

In contrast to simulation, two quantitative programs have potential for engineering design processes. An Excel model computes various descriptive performance indicators and SPSS identifies relations among process variables to find 'waste'. The manual and OTS approach are too comprehensive for the research time available.

Quantitative modeling techniques need to provide the desired results: the performance metrics. The performance metrics need to align with the performance indicators of Figure 9. Literature summarized in Appendix C, provides insight in the metrics that apply for lean engineering design transformations; presented in Table 5. Note that the performance indicators also cover the relevant 'waste' types as formulated by the lean philosophy in section 2.4. The next provides a main explanation to the performance metrics that do not explain itself; Appendix C, subsequent sections and chapters provide more information and formulas.

The quality loss function measures the deviation of a process' performance towards a specified outcome, e.g. the status quo. This comes close to the standard deviation with the addition of a cost factor to express the importance of a quality deviation.

The 3C value leveraging model provides insight in the balance of the value drivers Continuation, Conception and Configuration that represent the value-time curve of a process. The value-time curve indicates the size of a process' investment, the time it takes to earn it back and the market share / earnings this will provide in the long term.

Table 5: Relevant performance metrics identified in lean transformation literature that is consistent with the lean 'waste' types and the main performance indicators of this thesis.

Cost	Amount of ambiguous processes	Design throughput time	Amount of process interdependencies
In euros per project	In # of ambiguous processes	In weeks per project and equals lead time (L/T) (Dotoli et al., 2012; Rother & Shook, 2009)	In # of interdependencies
Quality loss function (Lu et al., 2011)	Non Value Added Work (NVAW) (Cuatrecasas-Arbo, Fortuny-Santos, & Vintro-Sanchez, 2011; Rother & Shook, 2009)	Value creating time (VCT) (Rother & Shook, 2009)	The amount of separate process steps (Rother & Shook, 2009)
3C value leveraging model (W. Beelaerts van Blokland, van der Meer, & Rakers, 2009)	Resource utilization (Dotoli et al., 2012; Marzouk et al., 2012)	Takt time (Rother & Shook, 2009)	System Coupling Level Index (SCLI) to measure complexity as function of interfaces (Jeong & Phillips, 2011)
Productivity or labor value add (W. Beelaerts van Blokland et al., 2009; Cuatrecasas-Arbo et al., 2011)	Error rate (Behrouzi, Kuan Yew, & Behrouzi, 2011)	Cycle time (C/T) (Mahfouz et al., 2011; Rother & Shook, 2009)	The amount of design rules
Transaction costs (de Jong, Beelaerts van Blokland, Curran, & de Jong, 2013)	Tier rejection rate (Behrouzi et al., 2011)	Total waiting time (Cuatrecasas-Arbo et al., 2011)	
		Average waiting time (Cuatrecasas-Arbo et al., 2011)	

Transaction cost theory allows for an indication of the lean cost effect by investigating the effect of different process structures on turnover, profit margin and/or inventory. A lean transformation shows an improved relative and absolute performance.

Inventory on its turn, indicates the physical storage of goods. This definition does not fully cover for data storage since it lacks the acquaintance of raw materials, assembled goods or similar. On the basis of interviews at Siemens, the term non value added work (NVAW) best matches inventory in an engineering design process. NVAW includes both non value added work-in-progress (NVA WIP) as work queues. Processes that contain NVAW will encounter three cost consequences: (1) more overhead related fixed costs due to a longer estimation of the project duration, (2) more uncertainty in determining that project duration and consequent higher probability of fines due to

late deliveries and (3) interest in the case that the project owner (i.e. Prorail in this case) pays for intermediate data results.

The tier rejection rate implies the amount of times that a project undergoes the same process. This could for instance result from errors.

The takt time stands for the time that a project may take in order to match the demand rate.

The cycle time represents the total time to execute a process, i.e. the waiting time, plus the non-value-added process time, plus the value creating time.

The amount of process interdependencies reflects the amount of processes that a particular process may affect.

The System Coupling Level Index normalizes the amount of interdependencies for fair comparisons between multiple process structures like lean and RailML.

Table 6: The benefits and disadvantages of various quantitative modeling programs/techniques used in lean transformation literature.

Modeling program	Analytical model characteristics				
	Level of detail	Computation rules	Time dimension	Presence of equilibriums	Other
Flexsim (Jeong & Phillips, 2011)	All	Stochastic	Yes	No	-Limited input: throughput, process times, queues and capacity.
ARENA discrete event simulation (Dotoli et al., 2012)	All	Stochastic	Yes	No	-Strong UML relation -Large modular processes -Animation
Powersim system dynamics (Agyapong-Kodua et al., 2009)	Meso / Macro	Deterministic	Yes	Yes	-Cause-effect identification -Non-parametric
Simul8 discrete event simulation (Agyapong-Kodua et al., 2009)	All	Stochastic	Yes	No	-Production focus -Strong VSM relation -Value stream imitation
Extendsim discrete event simulation, especially with VSMSx plug-in (Gregg, Van Andel, & Saylor, 2011; Marzouk et al., 2012; Shararah, El-Kilany, & El-Sayed, 2011)	All	Stochastic	Yes	Yes	-Production focus -Open source -Clear database structure -Situation specific plug-in development required -Limited performance measurement -Only face validation
Operations-Time Charts graphical evaluation tool (Cuatrecasas-Arbo et al., 2011)	Micro / Meso	Deterministic	No	Yes	-Production focus -Enables identification of 'waste' -(Visual) Representation of performance evolution -Complex compared to project management software -Animation
TOMAS discrete process simulation	Meso	Deterministic	Yes	Yes	-Not event but state triggered -Closely aligned with common business applications -Delphi based, no GUI -Object oriented -Parallel processing -Animation
SPSS statistical analysis	Micro	Deterministic	No	No	-Identification of data distributions -Identification of variable correlations -Cause-effect analyses -No what-if analyses
Excel spreadsheet	All	Deterministic	No	No	-Transparency -Many mathematical options -Complexity rises with many interdependencies
Manual	All	Deterministic	No	No	Complexity rises quickly

3.1.4 The Lean Transformation Methodology in a Nutshell

Figure 14 visualizes how the transformation directions (3.1.1), the transformation framework (3.1.2) and modeling techniques form one lean transformation methodology for engineering design processes.

The engineering design of interlocking systems follows a specific chain of activities that lead to a to-build specification. That is where the engineering design stops and production starts. Application of the lean transformation methodology results in a benchmark that mitigates all 'waste' in the engineering design chain. System engineering indicates how RailML will change process structure.

Then on the basis of modeling, a lean benchmark can quantify the lean effect of RailML.

The following sections develop the lean benchmark on the basis of interlocking requirements, 'waste' in the current state and application of the transformation directions. Chapter 4 engineers RailML for the purposes of interlocking and to indicate its effects on the current process structure. This report puts this detailed design step in the middle of the project instead of the end because a RailML interlocking formalization does not exist yet. Chapter 5 then compares the status quo, RailML and lean benchmark states by means of modeling.

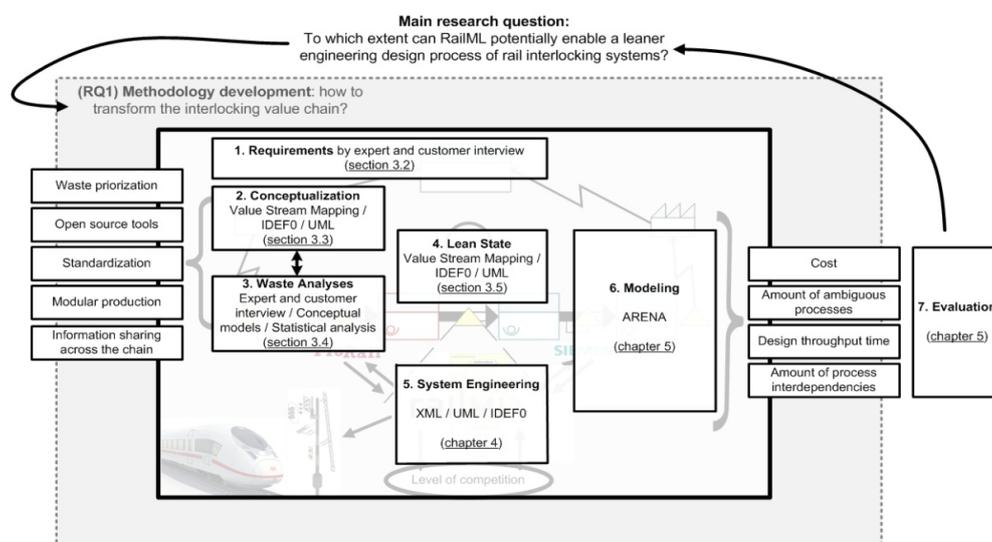


Figure 14: The lean transformation methodology merged with the research structure visualization.

3.2 Prorail's Requirements for Interlocking Systems

This section presents the requirements for interlocking systems demanded by the client: Prorail. One could however argue that the train operator is the actual client of the system. In reality, the operator does not act as a client since he lacks the necessary interlocking knowledge. The operator will however influence the process when interlocking logic significantly affects operations. This requires Prorail to make very clear trade-offs between safety and operational conditions like track capacity.

Prorail initiates an interlocking project based on three grounds (Mol, 2009; van de Luijngaarden, 2007; van den Nieuwendijk, 2010):

- a new interlocking system for all relevant areas on a corridor, e.g. renewal of an entire interlocking;
- a small (i.e. few points) or big (i.e. many points) modification to a current interlocking, e.g. the removal of a switch;
- a modification to support the implementation of other train control systems like ETCS.

A lean development of interlocking systems ideally relies on a standardized approach in which an interlocking area is composed from various standard 'building blocks'. A group of requirements may allow for such standard 'building blocks', e.g. similar switch types, when they have such an unambiguous application that an IT system can capture it in rules. To illustrate: requirements for a simple switch always prohibit flank movement of the incoming or diverging track. A simple and quick engineering design process results when interlocking systems are composed of only such 'building blocks'.

Unfortunately, interlocking systems contain unambiguous requirements that do not allow for standardized 'building blocks'. First of all, the positioning of track elements like switches, detection and signals always differs per interlocking area; even when the track layout seems the same. In addition, some exotic flank protection measures exist that depend on either the interlocking area type and track complexity. Therefore, generating interlocking IT with the "push of a button" can hardly become reality.

Figure 15 indicates the main groups of requirements for an interlocking system; the next five subsections explain what

each requirement group is about and why it can be standardized or not.

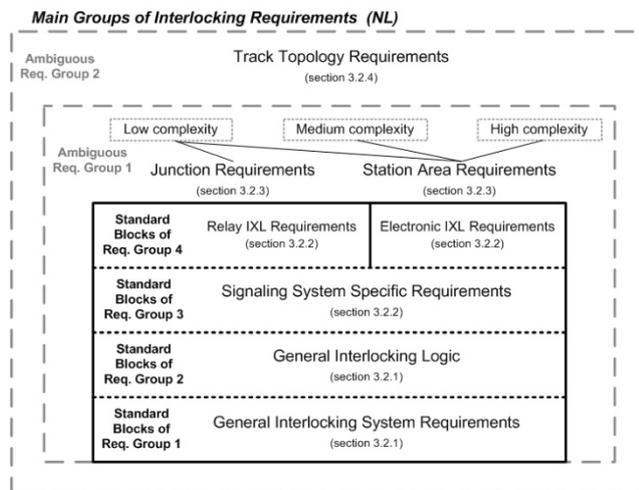


Figure 15: The main groups of requirements that lead to an interlocking system for the Dutch railway network. The fixed black blocks indicate the requirements that can be standardized and as a result be automated with an IT tool. The dotted grey boxes around the standard 'building blocks' indicate the requirements which contain varying requirements that do not allow for standardization.

3.2.1 The General Interlocking Requirements

General interlocking requirements elaborate on the IT system and the interlocking logic that applies to the entire Dutch railway network. These requirements allow for standardization since an interlocking system should always meet them. Prorail distinguishes functional and non functional IT system requirements; the functional system needs to include (Prorail, 2012c):

- communication with traffic control systems;
- control switchable field elements such as points and signals;
- set, control and release of routes containing multiple movable track elements;
- the possibility for trains to combine and/or switch drive direction;
- the enforcement of measures when a train passes its allowed movement authority;
- a mean to provide the signal system of commandos;
- a mean to provide the train protection system (ATB-EG, ATB+ & ATB-NG) with required route information;
- the activation of level crossings and warning equipment;
- the possibility for shunting movements;
- communication with neighboring interlocking systems;
- maintenance possibilities.

The relevant non-functional requirements encompass (Prorail, 2012c):

- very low chance of failure, i.e. $1E^{-9}$ errors per hour at an interlocking area with 14 signals, 28 detection sections and 12 single switches. Industry calls this the fourth

Safety Integrity Level (SIL4). The main challenge for an interlocking engineering party depends on his ability to quantify this level of failure. This limits the amount of parties that actually engineer interlocking systems;

- the to-build interlocking architecture (e.g. SIMIS W) needs to be proven in practice and may only contain modifications as the interlocking area demands;
- the system should operate for at least 15 years and the tendency is an even longer lifetime;

Maintenance, as part of the Reliability, Availability, Maintenance and Service (RAMS) definition, misses in the above list of non-functional requirements because Prorail defines this per interlocking area. The party that executes maintenance mostly depends on the complexity of the interlocking system. Generally, interlocking products from industry require the assistance of that company (like SIMIS W) and Prorail or an engineering agency may maintain a simpler interlocking (like the PLC). The trade-off for an infrastructure provider usually comprises multiple factors like cost, probability of interlocking errors, maintenance quality, availability of engineers and so on. Since the track topology requirements incorporates maintenance, further detail is outside the scope of this research.

In terms of interlocking logic, Prorail defines nine common track layout scenarios with possible train interactions that an interlocking needs to prevent by protecting the flanks, conflicting routes and overlap (Prorail, 2012a).

Table 7 defines those nine scenarios which allow for standardization. Prorail defines area specific logic for other track scenarios like a double point, ladder or double cross-over.

3.2.2 Interlocking Requirements Depending on the Hardware Type and the Signaling System

Prorail will very likely demand an electronic interlocking system when dealing with a new and ETCS interlocking project, e.g. the Delft tunnel project (Boxmeer, 2012). A new interlocking project has to comply with ETCS in the future as well thanks to European regulation (InStall, 2011). Besides, Prorail demands the electronic interlocking over traditional relays due to an operational reason. A drawback of a relays based interlocking is that once it activates a route, it cannot easily undo this. In most cases, either the train needs to pass certain points or the relay waits for a timer to elapse, before a route becomes passive again (Theeg, Maschek, et al., 2009; Theeg, Nasedkin, et al., 2009). In general, this leads to capacity reductions. ETCS on the other hand, increases track capacity mainly by providing (and enforcing) dynamic braking curves to the train driver and shorter block sections / following distances (Winter, 2009b).

In addition, relays based interlocking systems impose difficulties for ETCS level 2 or 3 migration strategies. Consider the trackside dual signaling migration strategy at Utrecht-Amsterdam with both ATB-EG and ETCS level 2; in which ETCS uses multiple block sections instead of one ATB-EG block section (Goverde, 2012b; Janssen, 2013b; Winter, 2009b).

Table 7: Prorail's standard interlocking logic. The letter pairs imply both driving directions.

Track scenario:	Scenarios to prevent:	Topology:
Simple switch	Prevent: C-B when A-B is active or, B-C when A-B is active.	
Diamond crossing	Prevent: A-B when C-D is active or, C-D when A-B is active.	
Crossover	Prevent: C-D and A-B when A-D is active or, when it does not concern a coupled switch, i.e. both switches can only be in straight or both in bend position, A-D when C-D is active or, A-D when A-B is active.	
Crossover to change course	Prevent: C-D and A-B when either A-D or A-E is active or, C-D when C-E is active or, C-E when C-D is active or, not in the case of a coupled switch, A-D or A-E when C-D is active or, A-D or A-E when A-B is active.	
Double slip crossing	Prevent: C-D and C-B when A-B or A-D is active or, A-B and A-D when C-D or C-B is active.	
Single slip crossing	Prevent: C-D and C-B when A-B or A-D is active or, A-B and A-D when C-D or C-B is active.	
Temporary middle third track	Prevent: C-F, D-E, A-B and E-B when A-F is active or, A-F, C-F, A-B and D-E when E-B is active or, A-F, C-F, C-D and E-B when D-E is active or, A-F, D-E, C-D and E-B when C-F is active.	
Side track	Prevent: A-D when B-C or A-B is active or, B-C when A-D or A-B is active.	
Two double tracks merging	Prevent: A-E when A-C is active or, A-C and D-B when A-E is active or, B-F and A-E when B-D is active, D-B when B-F is active.	

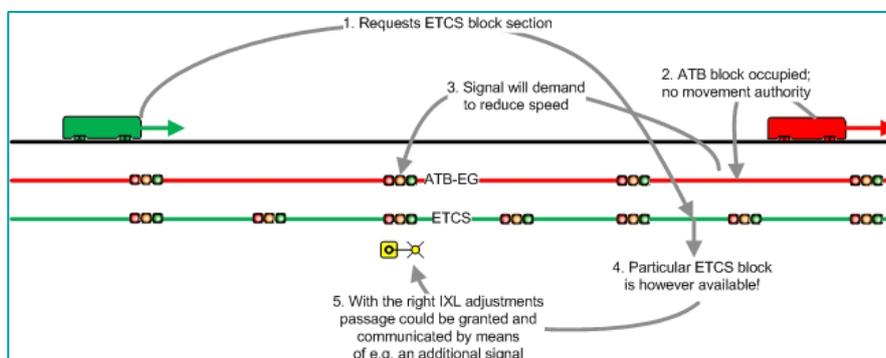


Figure 16: Shows an example of a dual signaling trackside migration strategy with two trains driving at different signaling systems. An electronic interlocking allows easier modification to support an ETCS train to interlock a free ETCS block while the corresponding ATB section is occupied.

In the case that an ATB-EG train interlocks a block section and an ETCS train follows, then the classic signaling system would refuse the ETCS train to enter that same ATB-EG block even though the sub ETCS blocks might be available (Figure 16). As a result, the track capacity at ETCS will drop. Prorail could take various measures to allow an ETCS train to enter the ATB-EG block with a preceding ATB-EG train. As for Utrecht-Amsterdam, they use additional signals to indicate whether ETCS overrules ATB-EG. All measures however lead to an extensive extension of the relay system to cope with additional detection and element dependencies. Such a relay extension is very comprehensive compared to an electronic interlocking. The previous implies two things: first, different requirements exist depending on the type of hardware, e.g. the duration of the release timer. Second, the type of signaling system imposes a different set of requirements, e.g. to include additional signals. Both groups of requirements do allow for standardization since the requirements per solution, e.g. ETCS, ATB-EG or electronic IXL, do not change per interlocking area.

3.2.3 Ambiguous Interlocking Requirements Depending on the Track Topology

Two general track topology layouts exist: the open track and station areas. An open track connects station areas. By definition, open track does not contain switches although sometimes trains may cross tracks in an area that is not a station either. This leads to a second, interlocking topology categorization: a station area or junction. The station area contains more complexity thanks to a larger number of rail elements and tracks than a junction can contain. In addition, station areas sometimes contain more 'exotic' interlocking logic than the general logic in Table 7. The flank zone safety system rule is an example. When a train drives into a side track, for instance to pick-up passengers, then this rule determines whether an interlocking may interlock a route on the same block for another train (Figure 17). The situation raises questions of safety because the passing train could cross the same block section as the other train drives at the side track. On the other hand, track capacity may reduce when trains cannot pass. Therefore, Prorail judges each of these situations independently.

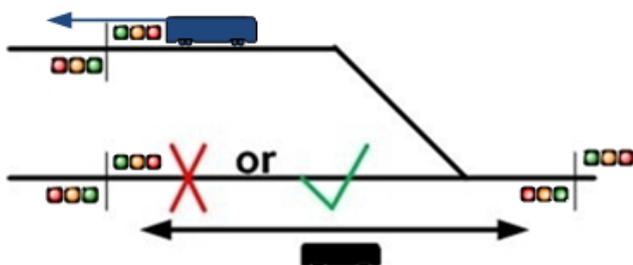


Figure 17: An example of the flank zone safety system rule. When the blue train moves onto the side track and has not left the switch's detection area, the black train on the through track is not allowed. This rule however says that for reasons of capacity, a through train may sometimes request /use the through route at certain station areas.

Interlocking systems at junctions usually stick to the general logic as shown in Table 7. Interlocking routes in junction areas does however occur at other circumstances,

such as higher speeds, which raises other interlocking requirements. An example is a crossover at high speed with three or more parallel tracks (Figure 18). Prorail needs to determine whether a train may drive at the third track's neighboring block section when a train switches from track one to track two at high speed. This interlocking question arises because the switching train may in an extreme case derail and crash into another train.

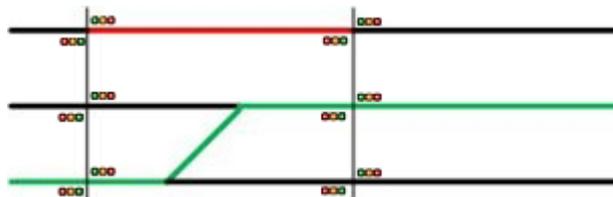


Figure 18: The "demand" switch example on open track. Sometimes the interlocking needs to prohibit a train on the third red track when a train switches to the middle track (green line).

Besides topology layout, more rail elements influence the interlocking requirements as well. The amount of rail elements at station areas comes down on various variables like the amount of tracks, track circuit sections, amount of merging directions and so on. Appendix H investigates the relation between all station interlocking areas in the Netherlands for combinations of rail elements. In general, more elements of one type, like switches, also increases the amount of elements of another type, like track circuit sections. The analysis shows a linear relation between the number of welds, track circuit sections and signals. In contrast, the analysis shows a less linear pattern for the amount of switches or tracks with either the amount of signals or track circuit sections. This implies that the amount of switches and tracks causes additional uncertainty in the type and amount of requirements. The uncertainty results from special demands of the infrastructure manager. On the one hand, this proves the statement of "exotic" interlocking logic regarding topology. On the other hand, the uncertainty could also arise as a result of different train flow strategies. To illustrate, a strategy that allows trains to follow shortly after another requires substantial more track circuit sections and signals than usual; relatively independent of the amount of tracks and switches. An interlocking system must include all those elements with characteristics like geographic location which does not allow for standardization (since the location of elements is unique).

In order to express varying project complexity in the remainder of this study, three interlocking areas are defined on the basis of common station sizes in the Netherlands. The amount of rail elements are derived by comparing station areas in InfraAtlas:

- little complexity (based on station with two platforms): about 2 tracks, 6 signals, 4 single switches, 7 detection sections;
- medium complexity (based on station with four platforms): about 3 or 4 tracks, 14 signals, 12 single switches, 28 detection sections;
- high complexity (based on station with \geq four platforms or extraordinary topology): about >4 tracks, >40 signals, >40 single switches, >100 detection sections.

3.3 The Current State of the Interlocking Engineering Design Process

This section identifies the interlocking engineering design activities considered as ‘waste’ according to the lean philosophy. Subsection 3.3.1 first maps and quantifies the current sequence of interlocking design and engineering processes. Subsection 3.3.2 presents the lean performance of Prorail’s interlocking design; subsection 3.3.3 does the same for Siemens.

3.3.1 Today’s Interlocking Engineering Design Process

The engineering design process covers project definition, system requirements, system specification and IT system engineering as shown in Figure 6. This subsection investigates those processes in more detail.

The value stream mapping conceptualization technique identifies the chain of processes with associated resources, information and key parameters in Appendix D. Documentation on the precise chain of events does not exist; especially not since multiple parties participate. Therefore, various sources provided the necessary data and information to develop the VSM. Those include interview sessions (Koelewijn et al., 2013; Storck & Dragt, 2013; Wisotzki & Muehlhause, 2013) of which Appendix F contain summaries, process manuals (Siemens, 2013b; Siemens AG, 2013), design manuals (Prorail, 2010c, 2010d, 2012b) brainstorming with various Siemens employees, intermediary design maps and reports and project proposals.

Figure 19 shows a simplified map with the current process sequence from project definition till the construction of the actual interlocking system.

As the name implies, the engineering design process encompasses a design stage and an engineering stage. The design stage focuses on formulating project requirements, formulating signaling solution boundaries and gathering or measuring data such that an interlocking can be engineered. Prorail controls and manages the design stage but engineering agencies take care of the design tasks. It is worth mentioning that the design stage does not just focus on engineering interlocking systems. The design stage could also form the basis to engineer traction systems for example. This study however focuses on interlocking systems and will therefore not consider other applications of the design stage.

The engineering stage extracts the relevant data from the design stage and engineers the interlocking on the basis of an extensive amount of guidelines from the infrastructure manager. Usually an industrial party but sometimes also an engineering agency will take care of engineering. Before manufacturing of the interlocking hardware, Prorail compares whether simulation behavior on the basis of the design stage aligns with the engineering result. When the comparison provides sufficient results, the production process starts and the engineering design process ends.

The design stage starts when Prorail and the relevant stakeholders agree for the need of a (re-)new(ed) signaling system at an interlocking area. At that point, Prorail and stakeholders define project requirements during the

“Client Requirements Specification” (CRS). The project requirements include agreements on e.g. rules and regulations, cost, time and RAMS.

The CRS contains agreements on headlines, the “Functioneel Integraal Systeemontwerp” (FIS) aims to translate these requirements into realizable strategies. An engineering agency executes the FIS, Prorail monitors the result. The FIS processes recently became mandatory to reduce the number of rejected results from the “RailVerkeer Technisch Ontwerp” (RVTO). Prorail discarded quite some RVTOs because the basis for an unambiguous rail concept, i.e. project definition and system requirements, missed from time to time. The FIS first results in a selection of strategies with consequent trade-offs. Accordingly, Prorail chooses one final strategy that the engineering agency complements with more detail like civil data and situation drawings. This results in system requirements of the to-be situation.

The statement of final design and specification of the interlocking area form the RVTO and aim to design the to-be state for the purposes of rail operation. Another, or the same engineering agency as the FIS develops the RVTO. The statement of final design first enriches the FIS with additional rail characteristics that impose infrastructure and signaling restrictions. Examples include traction, level crossing situations and braking curves. In addition, the engineering agency develops a project planning for every subsequent process in the interlocking chain. The second RVTO process specifies the future state of the interlocking area in terms of e.g. track layout, signal aspect dependencies, longitudinal maps and cross sections. The RVTO results in a traffic related overview track map (VT-OBE), signal maps (OS) and a report. The engineering agency always develops these files from scratch and does not (fully) rely on previous versions.

The same or another engineering agency elaborates on signal technological feasibility of the track layout in the “SeinWezen OverzichtsDossier” (SWOD). The SWOD for example investigates the adaptation of track circuits and detection in the context of the new signaling strategy. Sometimes the result demands a change of the RVTO specification. The SWOD results in a final “Overzicht Baan & Emplacement” (OBE) map, an overview map of return wires / physical interlocking infrastructure (OR) and state of indications (SVA) necessary to engineer an interlocking. The completion of the SWOD implies the end of the interlocking design stage. Prorail starts a tender to select which industrial party may initiate the interlocking engineering.

Prorail then delivers the files to the interlocking engineer, e.g. Siemens. Siemens needs to validate the data and digitize them in the first place. After all, OS, OBE and OR maps arrive in a machine readable form. Siemens also validates the data, most and foremost the signal aspect dependencies, to minimize the probability of mistakes. When the data is completely prepared, the topology describing IT system can be developed; at Siemens, that final system is called Element VerbindungsPlan (EVP). An EVP relies on a Technical Overview of Signalaspects (TOS) that reflects signal aspect dependencies in the interlocking area. EVP engineering then combines the TOS with the location and functionality of the interlocking relevant rail

elements in a node-branch model. In this way, the interlocking knows which elements it needs to interlock and which elements to check before interlocking a route. Siemens converses the EVP into firmware and test hardware to investigate whether both hardware and software perform as desired. The EVP conversion is therefore very similar to making a prototype; the EVP test similar to a prototype test. Prorail controls the (test) results before Siemens actually starts building the interlocking. Prorail uses a simulation tool called BITS to emulate an interlocking

environment and compare it to the behavior of industry's interlocking product. When no significant issues arise, the design engineering process stops and the production process starts.

Important note: The next of this report focuses on interlocking areas of stations because they contain all levels of complexity. Please refer section 3.2 for the possible project types this could result in.

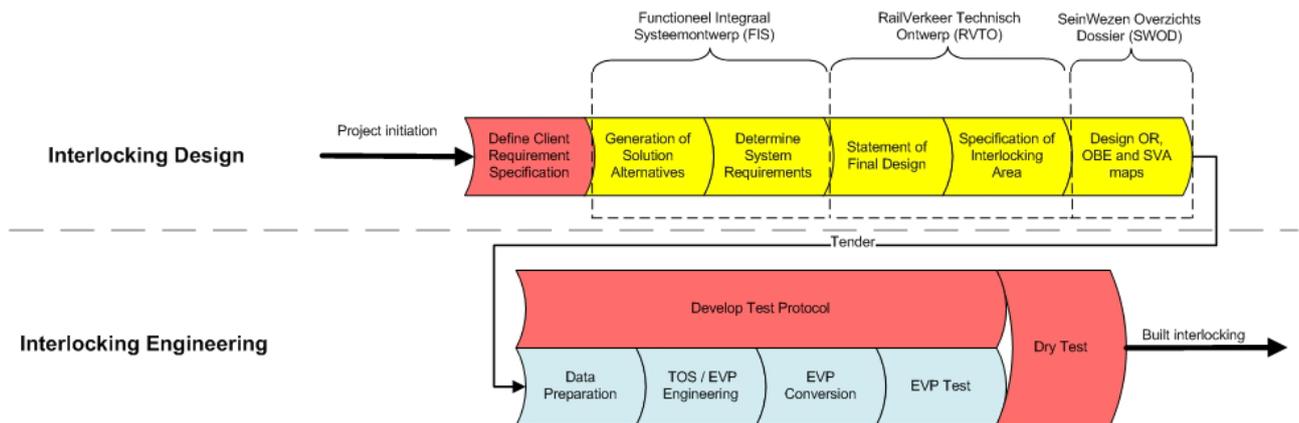


Figure 19: An aggregated overview of the engineering design chain of rail interlocking. The red process boxes and arrows correspond to an action of Prorail. The yellow process boxes are executed by one or more engineering agencies. The blue boxes belong to one interlocking engineer like Siemens.

3.3.2 The Lean Performance of the Interlocking Design Process

The interlocking design process defines client requirement specifications and develops OR/OBE/SVA maps that meet those requirements. Prorail provided mainly time related process characteristics, which puts a focus on the time performance indicators of Table 5. Appendix F provides summaries of the process related interviews at Prorail. The reliability of the figures depends mostly on the knowledge and experience of the five different people interviewed. A comparative analysis of the interviews does show a consistent picture but should be enriched with more interviews and figures for higher precision. Appendix D presents the calculations behind this analyses.

The cost performance could only be assessed by developing a productivity index. This index measures the weekly performance of an employee in terms of the product between design rules and average track elements:

Equation 1: The productivity equation applied for the design stage.

$$Productivity = \frac{(\# \text{ rail elements at complexity level} * \# \text{ of design rules})}{\text{Value added time of process}} / \# \text{ employees}$$

Figure 20 shows that a medium complexity project generally delivers best productivity values. Employees could work relaxed when the project has few challenges and they deal with difficult situations with consequent slow progress when working on high complexity projects. Figure 20 furthermore shows that the CRS, RVTO and SWOD process have lower productivity rates than the other processes.

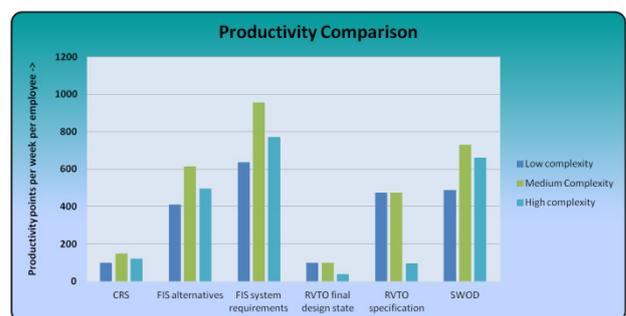


Figure 20: Productivity for each engineering design process per interlocking project's station area complexity type. The dry test and test protocol development processes do not have a productivity value due to an uncertain estimate of the amount of design rules.

The amount of ambiguous processes, NVAW sizes, tier rejection rates and error rates provide insight in the process ambiguity criterion. Experts (Koelewijn et al., 2013; Storck & Dragt, 2013) and the Value Stream Map in Appendix D reveal the most obvious ambiguous processes:

- Development of the OBE map happens at RVTO and SWOD. Although both focus on another aspect, the SWOD engineers often have to redo RVTO work.
- The RVTO development often starts from scratch while related documentation, even old RVTOs, are available. The high cost of making errors causes this situation.
- Much data in the RVTO and SWOD processes are still exchanged by non machine readable formats. Each translation process, which occur at the RVTO, SWOD, test protocol development and Siemens' collect and digitize track data processes, thus equals to double work.

- The client requirement specification process often gets adjusted while the FIS or sometimes even the RVTO process takes place. In the worst case, the process needs to restart at the FIS.
- Engineering agencies start too early with FIS, RVTO or SWOD and miss essential information and/or data. This results in rework.

Non value added work (NVAW) actually implies idle projects that wait for employees to finish their other projects. In reality, employees will work on multiple projects at the same time. Therefore, NVAW distinguishes non value added work-in-progress (NVA WIP) from queue sizes. Queue sizes do not follow from the static data delivered by Prorail but NVAW do by dividing the non value added process time over the total process time. This performance metric shows that non value added project work could be avoided by better planning. Figure 21 shows that especially during the start of the chain, projects pile up. Those piles are likely to occur due to the large chance of rework when a CRS gets changed.

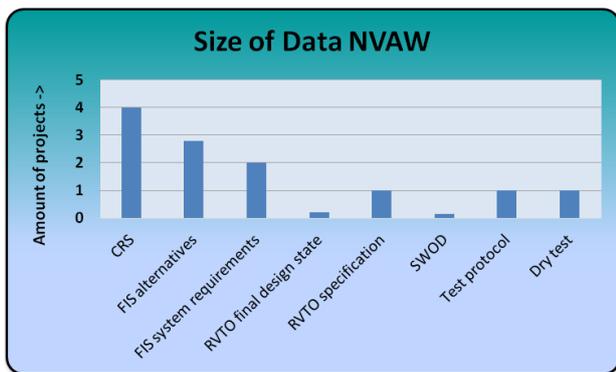


Figure 21: The amount of interlocking projects in idle state per engineering design process.

The tier rejection rate relies on estimates from experts (Koelewijn et al., 2013). Prorail checks the FIS and RVTO after completion. Experts explain that, when dealing with an medium complexity project, they never accept the result at once. The second time that engineering agencies offer their FIS and RVTO, it usually gets accepted. Consequently, a FIS and RVTO of a project at low complexity has a low chance of getting rejected. A FIS and RVTO of a project at high complexity could be rejected twice or more. A site visit intended to find the inconsistencies between rail data (VT-OBE, OBE and OS maps) and the real track situation at Santpoort Noord. Appendix G visualizes the differences by means of photos. The analysis showed a relatively high number of inconsistencies for the small interlocking area between real life and the RVTO process (VT-OBE and OS): 27. After the SWOD, which improves the VT-OBE into an OBE, no inconsistencies between real life and the OS or OBE remain. Furthermore, a recent change of infrastructure numbers also lead to 12 inconsistencies between the real life infrastructure and the OBE.

The lead times, shares of value added time per process, cycle times and waiting times, provide insight in the time performance indicator. Figure 22 hints that lead times, the time it takes to finish a project, might possibly grow expo-

nentially with increasing complexity. There is an insufficient amount of data to prove that hypothesis though it would make sense. Section 3.2 showed that more complexity increased the amount of rail elements. The possible amount of relations between the elements grows even stronger.

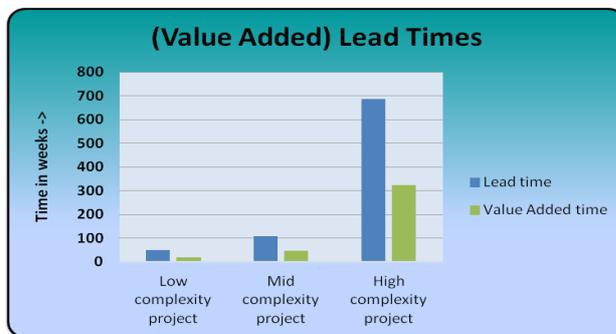


Figure 22: The total engineering design process time under command of Prorail; per type of station area complexity.

Figure 23 presents the shares of value added time versus non value added time (or waiting time) per process. The figures encompass all levels of project complexity since the experts believed non value added times and value added times change with the same proportion. RVTO's final design statement and SWOD have high value add shares as those processes include few uncertainty. The CRS and FIS however contain much uncertainty as the CRS can change over time which makes previous work non value adding.

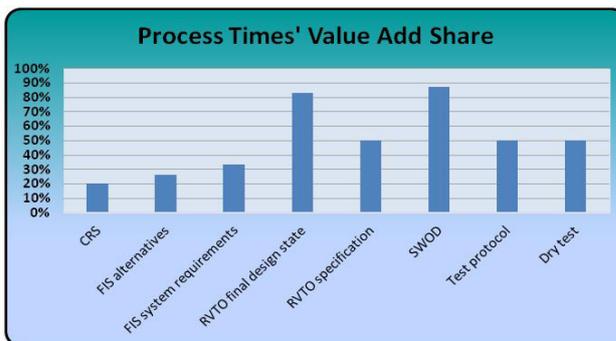


Figure 23: The share of value added time per engineering design process versus the non value added time.

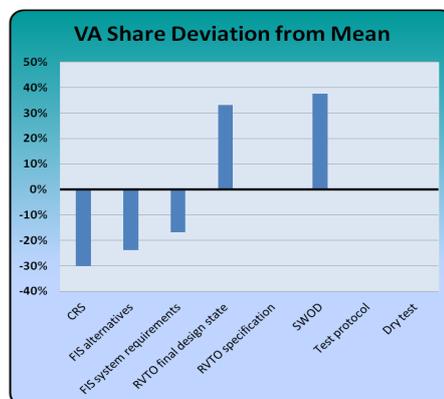


Figure 24: The difference from each value added share to the mean of all value added shares. The RVTO specification, test protocol and dry test processes barely deviate from the mean.

Figure 25 and Figure 26 make clear that a high complexity project needs substantially more cycle time to complete the RVTO process than a medium or low complexity project (a low complexity project shows the same pattern as medium). This behavior is probably caused by the increasing complexity as Figure 22 already hinted. Besides, the other peaks contain large portions of waiting time. This directly relates to the relatively large NVAW.

The fourth and last criterion comprises the amount of process interdependencies here measured by the amount of processes, the absolute amount of relations and the System Coupling Level Index (SCLI). The amount of process steps can be discovered from the value stream map in Appendix D. Appendix D also contains an N2 diagram to determine the amount of relations at those 13 process steps; Table 8 presents the conclusion.

The total number of existing interfaces in Table 8 also leads to the System Coupling Level Index (SCLI) of Jeong and Phillips (2011):

Equation 2: The System Coupling Level Index according to Jeong and Phillips (2011).

$$SCLI = \frac{TEI}{TPI}$$

Where:

$$Total\ Possible\ Interfaces\ (TPI) = \frac{m(m+1)}{2}$$

m = the number of modules or processes

TEI = the total number of existing interfaces between processes

Table 9 shows the SCLI and amount of design rules for each process. The SCLI and design rule figures show that the development of RVTO specification and SWOD contain most complexity. The SCLI value of RVTO specification and SWOD both encompass about 30% of the total amount of relations. The amount of design rules for these two processes is also much higher compared to other processes.

This subsection concludes with the main 'waste' that follows from this subsection's data analysis. The CRS and FIS formulate the project definition that forms the basis of the engineering design chain. The CRS and FIS appear as a strongly connected process with many iterations due to many actors with diffuse wishes. As a result, project management becomes hard and rework is sometimes required. Therefore:

- improve planning of CRS and FIS or make a clear split;
- reduce NVAW.

The RVTO and SWOD processes contain many sub processes with ditto interactions. Therefore:

- increase productivity of RVTO and SWOD by reducing complexity, e.g. by means of a supportive IT system;
- reduce ambiguity in RVTO and SWOD;
- merge or eliminate processes to reduce complexity; especially within RVTO and SWOD.

Usually two but sometimes three different engineering agencies participate in the interlocking design processes. Each engineering agency works according to its own standards with ditto data input, transformation and output requirements. Furthermore, engineering agencies measure a lot of data from scratch. This occurs while substantial amounts of data are usually available in the case of interlocking modification and ETCS projects but the reliability and accessibility is poor. Therefore:

- reduce number of data translations, interpretation mistakes and rework due to missing data;
- introduce an IT system or data exchange format in the chain to share data and manage file versions.

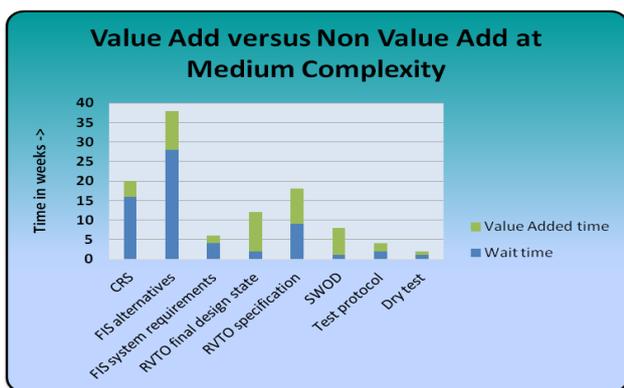


Figure 25: The absolute (non) value add times per engineering design process at medium complex station areas. The behavior of medium complexity areas is comparable to low complexity areas.

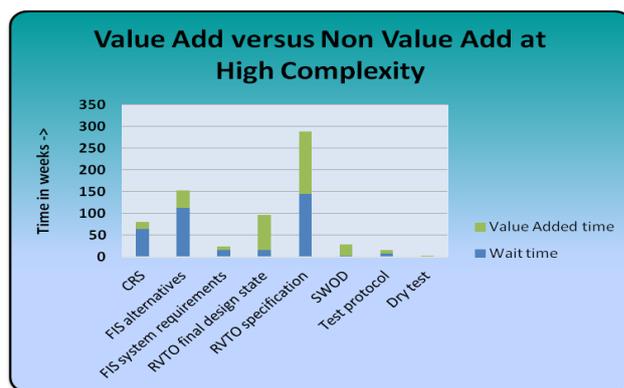


Figure 26: The absolute (non) value add times per engineering design process at medium complex station areas.

Table 8: The total amount of existing interfaces (TEI) for each engineering design process.

	CRS	FIS alternatives	FIS system reqs.	RVTO final	RVTO specification	SWOD	Test protocol
TEI	5	3	4	4	24	21	5

Table 9: The complexity of each engineering design process expressed in the SCLI and number of design rules.

	CRS	FIS alternatives	FIS system reqs.	RVTO final	RVTO specification	SWOD	Test protocol
SCLI	0.064	0.038	0.051	0.051	0.308	0.269	0.064
Design rules	21	216	130	87	370	359	

3.3.3 The Lean Performance of Siemens' Interlocking Engineering Process

Four interlocking projects assess the lean performance of Siemens. Each offer includes hour and cost data associated with the activities to engineer and produce the specific interlocking (consider Figure 19). The activities include:

- project management
- project control
- operation
- other support
- engineering
- testing
- other
- hardware

The projects represent the bids for Mistral corridors Apeldoorn-Deventer (apd-dv), Maastricht-Sittard (mt-std) and Den Dolder-Amersfoort (dld-amf). In addition, this analysis includes the recent interlocking project of the Delft tunnel (rsw-dtz), which Siemens engineers in the coming years. The limited amount of projects reduces the reliability of statistical analyses. Each process step contains 13 cases; divided per project type or corridor gives only 3 to 4 cases. Ideally, the data approximates a normal distribution for which about 30 cases or projects are required. Therefore, the results that investigate one aspect (e.g. the cost proportion per process step or corridor) provide reasonable results. The results that investigate multiple aspects (e.g. the cost proportion per process step split per corridor) should be assessed very critically. The statistical analyses follow a non-parametric approach and provide results with reasonable certainty. The data reliability can substantially increase by gathering more (contractual) data of comparable interlocking projects in the Netherlands. The next shows the most relevant figures; Appendix E presents all test results. Due to the confidential nature of the data, only relative figures are presented.

Table 10: The amount of cases for each of the nominal scales engineering process, project type and project corridor. The derivatives, like monetary allocation per corridor in

Project Characteristic	Amount of cases
Projects	4
Engineering process	13
Project type new	16
Project type small modification	12
Project type big modification	12
Project type ETCS	12
Project corridor Mistral	16
Project corridor Delft tunnel (only new project type)	4

Relative cost figures and productivity values assess the cost performance indicator. On average, the interlocking testing process consumes the largest portion of the interlocking engineering at Siemens (figures 27 and 28). Small, big and ETCS interlocking modification projects have a relatively large testing component compared to new interlocking projects (figure 29). This finding makes sense since Siemens outsourced tests in the Delft project. This also explains the variety in testing cost shares. New interlocking projects compensate low testing cost shares with relatively high cost shares for data preparation and EVP conversion. The EVP engineering contributes a relatively stable 30% of total costs for all project types and corridors. EVP engineering requires substantially more engineering costs than the other process steps. Furthermore, this result implies that each project requires about the same share of EVP engineering time; independent of project size. In addition, for all four process steps, at least half of the costs is absorbed by engineering activities, about 20% by project management activities and the remainder by a mix of hardware and testing activities.

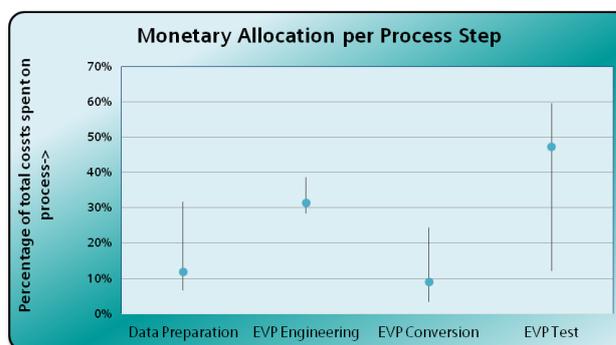


Figure 27: The distribution of total interlocking engineering cost shares for each interlocking engineering process. The dots present averages, the line ends minimum and maximum values. Each range is based on 13 cases.

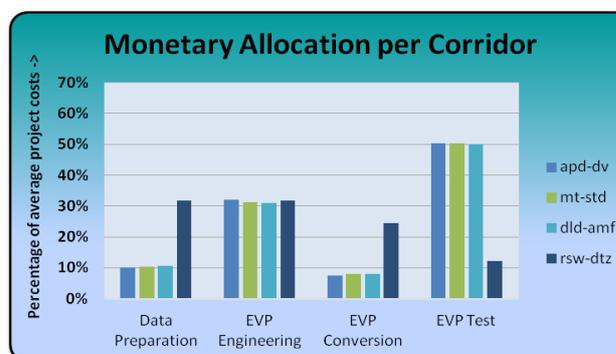


Figure 28: The distribution of total interlocking cost shares for each interlocking engineering process though categorized per corridor. Each bar is based on 4 cases with the exception of the rsw-dtz bars; those are based on 1 case per bar.

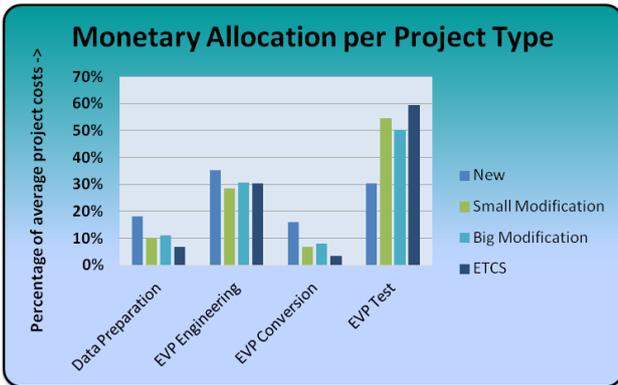


Figure 29: The distribution of total interlocking cost shares for each interlocking engineering process though categorized per project type. Each bar is based on 3 cases with the exception of the new project type that is based on 4 cases per bar.

Statistical analysis in Appendix E also points out that, when using a 5% level of confidence:

- the distribution of costs per activity (project management, project control, operation, other support, engineering, testing, other and hardware) is statistically equal between all corridors with the exception of project control costs, other costs and hardware costs. This implies that the ratio of costs between a new, small modification, big modification and ETCS project only differ between corridors, at the activities related to project control, hardware and others (Table 11);

Table 11: Illustrates by example what is meant with the statement that the distribution of costs per activity is statistically equal between all corridors with the exception of project control costs, other costs and hardware costs. The ratio for engineering costs in the green boxes is equal to each other and independent of corridor. On the other hand, the ratios between hardware cost accounts differs in the red box compared to the blue box.

#	Project Type	Corridor	Costs	
			Engineering	Hardware
1	New	A-B	10	10
2	Small modification	A-B	20	10
3	Big modification	A-B	40	20
4	ETCS	A-B	80	10
5	New	C-D	20	30
6	Small modification	C-D	40	50
7	Big modification	C-D	80	80
8	ETCS	C-D	160	100

- the distribution of costs per activity vary statistically between the four processes. The distribution of total costs per process, operation costs, engineering costs, testing costs and hardware costs differ significantly per process stage;
- the distribution of costs per activity is statistically unequal between all project types with the exception of operation costs, testing costs and hardware costs. This implies that the cost structures of each interlocking project type (new, small modification, big modification and ETCS) differ significantly from each other with the exception of hardware, testing and operation costs;

- there exists a positive correlation (0.737) between the process totals and the amount of interlocking track elements. Thus: the more elements, the higher the total interlocking cost and/or the other way around. It does however not count for each cost activity. More elements does not directly lead to a statistically positive relation with testing and operation costs.

The value generated per hour of work reflects productivity. Figure 30 shows that the data preparation process generates most value per hour, although it also has most variety. Especially new interlocking projects, of which a large amount of data needs to be collected and translated from maps into an IT system, explain the variety by some relatively low productivity rates (figure 31). New project types also most and foremost cause the average productivity rate of an EVP engineering process to be the lowest of the four processes (Figure 32). This finding counts for all corridors. Especially the new interlocking project at Delft generates some of the low productivity rates found in the four processes. Small interlocking modification projects on the Mistral corridors, closely followed by major modification projects, represent the peak productivity rates.

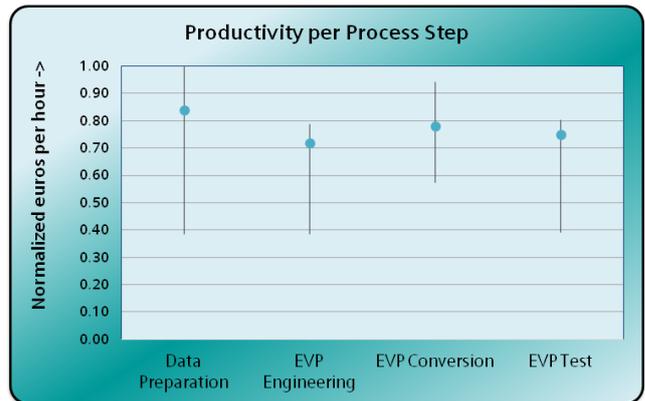


Figure 30: The distribution of normalized productivity rates, originally measured in euros per hour, on a scale of 0 to 1 where 1 stands for the highest productivity rate over all four processes. The dots present averages, the line ends minimum and maximum values. Each range is based on 13 cases.

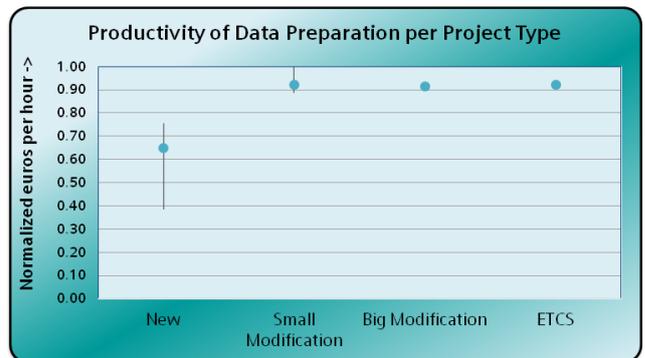


Figure 31: The distribution of normalized productivity rates, originally measured in euros per hour, for the data preparation process categorized by project type on a scale of 0 to 1 where 1 stands for the highest productivity rate in the data preparation process. The dots present averages, the line ends minimum and maximum values. Each range is based on 3 cases with the exception of new project types that base on 4 cases.

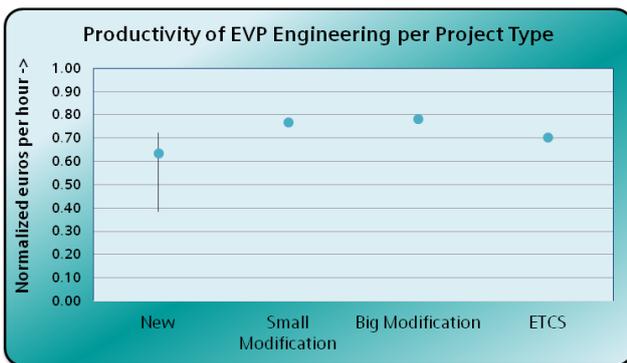


Figure 32: The distribution of normalized productivity rates, originally measured in euros per hour, for the EVP engineering process categorized by project type on a scale of 0 to 1 where 1 stands for the highest productivity rate in the EVP engineering process. The dots present averages, the line ends minimum and maximum values. Each range is based on 3 cases with the exception of new project types that base on 4 cases.

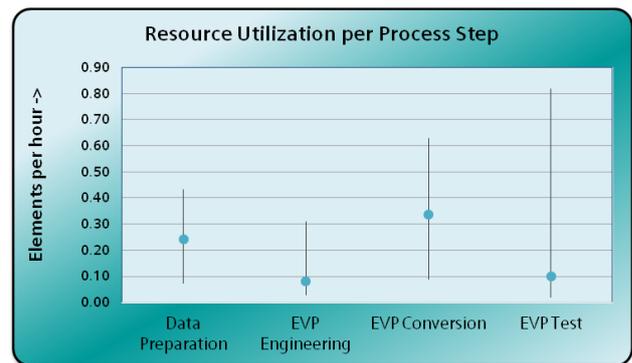


Figure 33: The distribution of resource utilization rates in rail elements per hour categorized for each of the four processes. The dots present averages, the line ends minimum and maximum values. Each range bases on 13 cases.

The number of ambiguous processes, resource utilization levels, error rates and the tier rejection rates assess the process ambiguity criterion. Expert interviews provide insight into the number of ambiguous processes (Wisotzki & Muehlhause, 2013):

- The data preparation process encompasses the understanding, development and validation of relevant and machine readable data. Those activities would not be necessary when both industry and infrastructure managers use (the same) IT systems or data exchange format.
- Siemens validates the engineering design in every process stage and finishes with a test of the interlocking prototype. Then, the infrastructure manager tests the interlocking behavior again by means of comparison with BITS' simulation results before production may start. The experts believe that this sequence of tests is required because of the various people involved, the large amount of data transfers and a substantial amount of non machine readable data transfers. The experts however assume that an IT tool or data exchange format may make the tests redundant.

The amount of track elements processed per manhour, labor being the main resource, indicates the resource utilization rate. Corresponding with productivity rates, the data preparation and EVP conversion have the highest average element processing per hour (Figure 33). The result is however skewed by the large variety. In general, the big change and ETCS projects have relatively high resource utilization rates at data preparation and EVP conversion. The resource utilizations at other processes and those of the other project types are however relatively low. The new Delft interlocking project forms an exception whereas it ranges between a high utilization of 0.3 to 0.8 elements per hour. When not considering the Delft project, the EVP projection and test utilization rates are on average about half the EVP conversion rate and a third of the data preparation process.

Figure 34 presents the ratio between productivity and utilization rate. This ratio provides insight in the importance of process steps. A high productivity rate that goes associated with a low utilization rate means that although use of resources is inefficient, the value created is significant. Thus, a high productivity versus utilization rate means an important stage in the interlocking process. Figure 34 then leads to the conclusion that the EVP engineering and EVP tests form the critical processes of interlocking engineering.

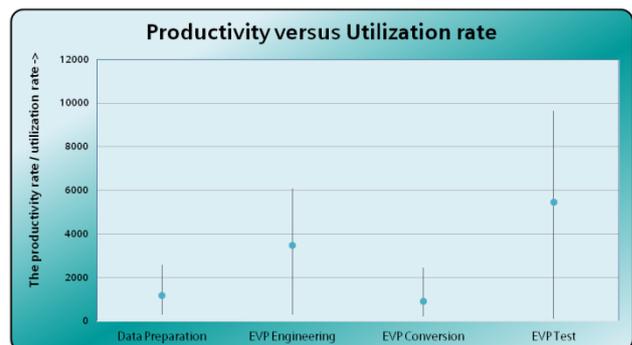


Figure 34: The productivity rates divided by the utilization rates. The dots present averages, the line ends minimum and maximum values. Each range bases on 13 cases.

Expert interviews provide an indication of the error rate and tier rejection rate (Wisotzki & Muehlhause, 2013). In terms of errors, the following situations could occur:

- Maps get translated in the wrong way. Therefore, the result sometimes misses track elements, represents track elements differently or positions track elements on different geographical locations.
- Partly based on the previous bullet, the EVP could miss or contain wrong relations and elements. When this type of error remains unnoticed, unsafe train flows may be granted.

Tier rejection could occur after internally testing the interlocking. The infrastructure manager then checks the interlocking as well and could find undesired design specifications. Various grounds exist but it is likely a result of a wrong interpretation during the data preparation process. As a result, Siemens needs to evaluate preceding processes

and in a worst case scenario restart from the data preparation process.

The project data only includes the contractual amount of hours required for each process. Therefore, cycle times indicate the design throughput time performance at Siemens. Figure 35 shows that the proportional cycle times closely meet the cost proportions in Figure 27. This raises the believe that only the processing time matters. Descriptive statistics and statistical analysis in Appendix E however show that the only non manhour related account hardware does form a significant portion of total costs, though it is a low and almost equally divided share over the four engineering processes (on average 13.8% of the total costs spent on a single engineering process as EVP conversion). Therefore, the amount of manhours forms the most important driver to achieve a cost reduction. Especially the engineering, project controlling and test accounting activities during EVP engineering and the EVP test have much time and cost reduction potential.

Statistical analysis in Appendix E also points out that, when accepting a 5% level of confidence:

- the distribution of time per activity (project management, project control, operation, other support, engineering, testing and other) is statistically equal between all corridors with the exception of project control time, other time and hardware time. This implies that the ratio of time between a new, small modification, big modification and ETCS project only differ between corridors, at the activities related to project control and other;
- the distribution of time per activity varies statistically between the four processes. The distribution of total time per process, operation time, engineering time and testing time differ significantly per process stage;
- the distribution of time per activity is statistically unequal between all project types with the exception of operation time and testing time. This implies that the cost structures of each interlocking project type (new, small modification, big modification and ETCS) differ significantly from each other with the exception of testing and operation time;
- there exists a positive correlation (0.793) between the process total time and the amount of interlocking track elements. Thus: the more elements, the longer the interlocking engineering design project duration. It does however not count for each time activity. More elements does not directly lead to a statistically positive relation with testing and operation costs. At 0,01 difference, the 'other' time account would also not lead to a significant relation.

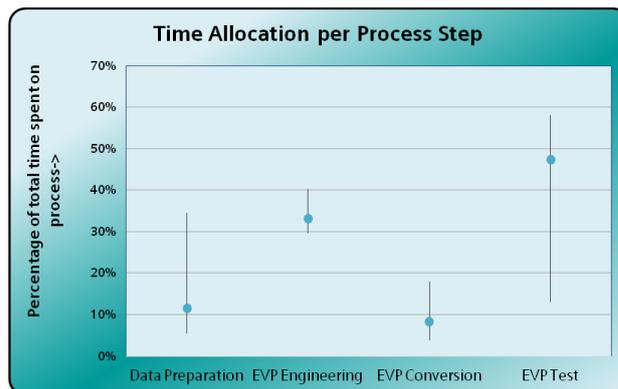


Figure 35: The distribution of total interlocking engineering time shares for each interlocking engineering process. The dots present averages, the line ends minimum and maximum values. Each range bases on 13 cases.

The amount of separate process steps, the amount of absolute process relations or interfaces and the SCLI describe the interdependency performance criterion. Figure 19 shows that the process at an industrial party as Siemens only has four clear process steps. The input, SWOD, and output, dry test, processes should however be taken into account as well due to the feedback relations that exist. Therefore, the interlocking engineering process at Siemens consists out of six processes.

Table 12 identifies that the number of relations for the dry test is actually larger than indicated in the previous subsection. As explained earlier, the dry test at Prorail could lead to conclusions that may require adjustments across the main engineering processes. Furthermore, Table 12 shows that the EVP test process contains second most relations; caused by the need to validate the EVP and interlocking test machine configuration.

Table 12: The complexity measured for the interlocking engineering process in amount of interfaces and the SCLI value.

	Data Preparation	Engineer EVP	EVP Conversion	EVP Test	Dry Test (at infrastructure manager / Prorail)
TEI	3	3	2	4	6
SCLI	0.143	0.143	0.095	0.190	0.286

This subsection concludes with the main 'waste' that follows from this subsection's data analysis. In general, the four engineering processes at Siemens benefit from:

- reducing cost variability by minimizing the importance of hardware or standardizing the hardware approach, e.g. by eliminating tests with prototype machines;
- minimizing overhead costs (process control, process management and the 'other' accounts) since they form a fixed share for all four processes independent of project type or corridor;
- reducing the amount of different (IT) protocols and non machine readable data transfers;
- an EVP engineering and testing focus since both form the backbone of the interlocking engineering.

The source of many required validation tests and rework originates in the data preparation process. Furthermore,

this data preparation process causes low productivity rates for new interlocking projects. A lean process therefore:

- merges, limits or eliminates the data preparation process;
- increases the remaining average productivity as the utilization rate is low for new and small modification projects, but high for big modification projects and ETCS projects.

The EVP conversion has a relatively low engineering time share. A lean process therefore:

- reduces the share of project management costs at the EVP conversion.

The value chain contains an ambiguous amount of interlocking system test. Furthermore, the testing time and cost account have statistically the same share for the four project types. In addition, the Delft interlocking project has substantially higher utilization rates and outsources a lot of testing at the same time. A lean process therefore:

- reduces or merges the number of testing (sub-)processes;
- reducing the amount of complexity measured in interface relations;
- makes interlocking tests (more) dependent on project size, e.g. by the number of track elements;
- increases productivity and utilization rates of the testing process.

3.4 The Lean State of the Interlocking Engineering Design Process

This section designs a new interlocking engineering design chain with the knowledge of the existing process 'waste' (section 3.3) and the successful lean transformation strategies (section 3.1). Subsection 3.4.1 first defines how the findings of previous chapters translate to a lean design process structure. Subsection 3.4.2 describes and conceptualizes the new process structure.

3.4.1 The Future State Approach: from a Traditional Interlocking Process to a Lean one

Rother and Shook (2009) define eight key questions to guideline a lean future chain design. These eight questions concern:

1. meeting the takt time;
2. a pull or push oriented process approach;
3. which processes can be merged to form one continuous flow without intermediate storage;
4. the location of a 'kanban supermarket' to mitigate the effects of upstream design deviations;
5. which process will determine the design planning;
6. how will different project types be leveled;
7. what will the control time frame be;
8. which process improvements are required to reduce 'waste'.

The current practice prevents a takt time approach to work since the customer, i.e. Prorail, does not request a rate of

interlocking projects per time frame. A takt time approach makes more sense with (continuous) production processes where each (level of) process results in the same product. Every result of an (interlocking) design process contains unique aspects. Therefore, first priority is a higher degree of standardized interlocking engineering design, e.g. by designing identical station areas. Then, a takt time approach becomes interesting.

A pull oriented design process starting downstream, makes project management less complex and reduces the amount of NVA WIP data. The CRS and FIS processes have relatively large NVA WIP; probably caused by the interactions with various actors. The unpredictable process times should not affect preceding stages, e.g. by pushing many or no projects in a time frame. Therefore, preceding processes should pull when the FIS needs to finish. As a result, a pull approach also minimizes the use of resources, i.e. mostly employees in a design process.

Merged processes also reduce the amount of NVA WIP and reduce data errors and cycle times. A separate or parallel process structure increases the amount of interfaces and thus complexity. The future chain design aims to combine those tasks that one party can execute with the support of IT systems.

The concept of a 'kanban supermarket' replaces NVA WIP between two separate processes. The 'kanban supermarket' differs from NVA WIP by stopping the upstream process when it keeps a certain amount of projects. This approach supports a pull process structure to keep a design process focused on a few projects. Different from a production process, employees could work on various projects within a time frame before completing it. NVA WIP never contains products not undergoing a transformation but equals files with incremental updates until completion. An employee processing multiple files then increases the chance on errors, delays, inconsistencies and so forth.

The future scheduling point in the chain positions itself at the end of the FIS process and the start of the RVTO. As discussed earlier, the FIS and CRS contain much uncertainty due to actor interactions. The preceding processes do not have that variability in process times. Therefore, the transition between FIS and RVTO is key to manage and reduce average process times.

Currently, the interlocking engineering design process already levels various projects. Leveled design meaning a mix of project complexities instead of batches of equal project complexity. A more systematic leveling approach would forcefully give some projects an upfront delay.

The control time frame reflects the data evaluation moments in time to check whether corrective action needs to improve the design process. This aspect of Rother and Shook is meant for operational purposes and will therefore not be included.

Besides the guideline of Rother and Shook, the literature analysis provided five high potential transformation directions (section 3.1): 'waste' prioritization, an open source Enterprise Application Integration (EAI) or Electronic Data Interchange (EDI) tool (Reuter & Rohde, 2007; Stadler, 2007), process standardization, modular design and information sharing. The next discusses how these trans-

formation directions take shape in the interlocking value chain.

Table 13 categorizes the process improvements of section 3.3 according to the three types of Womack and Jones (2003a): clearly non value add (first priority), non value add but impossible to avoid without IT (second priority) and value added activities (third priority). A lean value chain design should at least comply with all first priority and most second priority process improvements. The lean chain should enable third priority improvements though not necessarily focus on it.

Open source EAI or EDI tools should provide the IT support to enable the second priority process improvements in Table 13. RailML could contribute a significant share for a new EDI system to centrally store railway network related data that is required in all processes of Figure 8. The already open source character of RailML simplifies the incorporation of interoperable demands for the various railway systems in Europe. An EAI tool would most and foremost deliver benefits during the first processes of the interlocking engineering design. Furthermore, an EAI may improve the alignment between the infrastructure manager and the party that executes outsourced processes. Discussion of specific EAI tools is however outside of this research.

Standardization in engineering design processes mostly comes down to the reduction of translation / conversion processes: e.g. Siemens that cannot directly use track data presented on OBE maps. In other words: all processes and parties should speak the same design language. This demands that a standardized protocol or IT system contains the essential data to perform the activities at different parties' processes in the chain. In other words: output of process #1 should match the required input for process #2. An IT tool like RailML could play a standardizing role as well as it can formalize rail infrastructure elements in a general way; such that every rail infrastructure manager in the world could completely describe their system. Besides standardized communication, the actual work can be standardized as well. Standardized work means that design and engineering of various projects can largely rely on the same 'building blocks'. This makes the work focus on the unique requirements. Section 3.2 describes to which extent one could standardize work.

A modular design of a lean chain implies a clear distinction between activities within processes. A modular approach aims to achieve flexibility and enable the development of various interlocking projects. The chain could for instance distinguish separate workflows for each type of project complexity with an own pool of employees each. As a result, the demand variety of interlocking projects would not provide large planning fluctuations. Furthermore, all projects should contain as much variable costs as possible; at least distinguished by the three station interlocking complexity levels in section 3.2. Large fixed cost accounts, like testing nowadays, should be avoided to increase the attractiveness of an interlocking project.

A lean state needs to stimulate the sharing of design information and data. Every party and employee in the chain needs to be aware of design decisions, the format of process protocols and IT systems and the version of data files. For these purposes, processes require clear design decisions and visualizations to prevent black box results for

subsequent parties in the chain. In addition, the upstream processes need the requirements of the downstream result to effectively determine the project limits and mitigate rework. An IT tool in which each party can translate requirements back in upstream direction ensures this. Furthermore, 'failsafe' design methods should encourage innovative ideas and prevent bureaucracy. An IT tool could measure the effective transformation between two processes and assess whether this matches the goals of the project and the interlocking requirements.

Table 13: The 'waste' factors or improvement areas of section 3.3 categorized by the three 'waste' definitions of Womack and Jones.

Priority	'waste' definition	Process Improvement
1	Clearly non value added activity	<ul style="list-style-type: none"> • Eliminate ambiguity in RVTO and SWOD • Eliminate NVA WIP • Reduce the number data translations, interpretation mistakes and rework due to missing data • Eliminate Siemens' data preparation and EVP conversion processes • Make interlocking tests (more) dependent on project size
2	Non value added activity but impossible to avoid without the support of IT systems	<ul style="list-style-type: none"> • Improve planning of CRS and FIS • Merge or eliminate processes to reduce complexity; especially within RVTO, SWOD and test processes • Minimize overhead costs of Siemens' activities since they form a fixed cost share independent of project type or corridor • Reduce the amount of different (IT) protocols and non machine readable data transfers
3	Currently a value added activity	<ul style="list-style-type: none"> • Increase productivity of RVTO and SWOD by reducing complexity • Reduce cost variability of Siemens' activities by minimizing the importance of hardware or standardizing the hardware approach • Increase the average productivity of data preparation at Siemens • Reduce the share of project management costs at the EVP conversion • Increase productivity and utilization rates of the testing processes

3.4.2 The Lean Interlocking Engineering Design Process

Figure 36 presents an engineering design process structure after a lean transformation. Multiple new process structures could result. An important assumption in the design of this lean process structure was to mitigate as many of the 'waste' found in section 3.3. In reality, the parties in the chain could find some of those lean concepts too controversial. The next explains the (close to) maximum but realistic potential of a lean transformation.

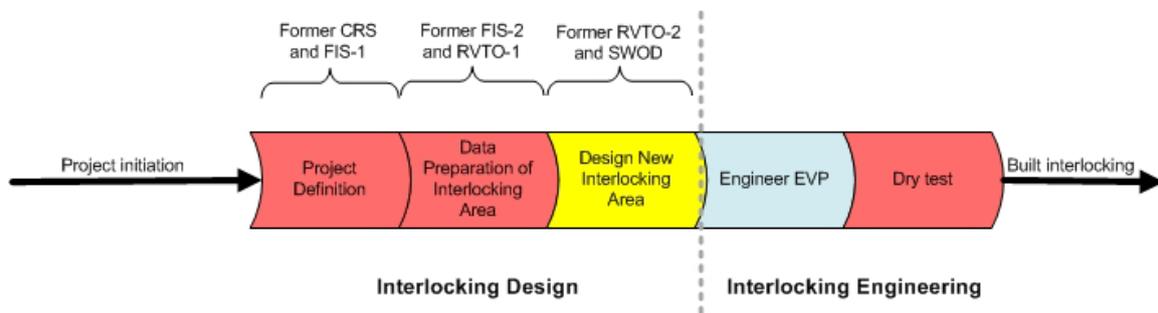


Figure 36: The aggregated ideal lean future engineering design chain of interlocking systems. The red blocks indicate processes executed by Prorail. The yellow blocks indicate processes executed by an engineering agencies. The blue box indicates a process executed by an industrial party like Siemens.

The design of the new chain starts at the end, the dry test process, to make sure each process step still meets the output requirements of section 3.2. The three different testing processes, a test at Siemens, development of a test protocol and a dry test, now combine into one test process. Currently, the total testing process is quite unproductive and complex due to the different processing formats used. Prorail has the disposal of topology drawings, requirements in text and an own interlocking testing tool. Siemens has the disposal of a prototype and an EVP. Aligning those by one standardized format potentially simplifies the testing processes to one comparison test of EVP and Prorail's requirements. In the long run, Siemens can eliminate the EVP conversion as well since it focuses on testing whether the conversion from various data files into an EVP aligns with the hardware architecture. The use of a proven standard data exchange methodology, throughout upstream process steps, does not require a physical interlocking test anymore since it will be proven to which specifications an EVP should adhere in order to comply with the hardware. An IT program can support and visualize the dry test to limit the chance of errors and provide a possibility for face validation. A face validation remains important to prevent a black box situation and keep system mistakes to a minimum (Palacios, 2013). Therefore, completely automating this process does not seem favorable. Prorail should ideally execute the testing process because they take final responsibility for the interlocking.

The EVP engineering tools need alignment with the output of Prorail's interlocking design and the testing input. Ideally, this process goes completely by an IT system. Realistically, some manual adjustments remain required. Section 3.2 clarified that certain interlocking situations stay unique; some cannot even be captured in mathematical expressions. Especially in the case of completely new interlocking areas, varying requirements remain. Furthermore, the EVP engineering itself contains some manual decisions that need the experience of a modeler for safety reasons. Therefore, EVP engineering remains a separate process at Siemens that another party cannot complete without related knowledge.

An engineering agency determines the structure of the new or updated signaling system, a.o. for the purposes of interlocking, in the interlocking area design process. Currently, this process starts with analyzing the implications

of the either new or changed track layout for the signaling system. The second process would typically position the signaling related elements like signals, detectors and speed limits/trajectories. Then, on the basis of requirements from the infrastructure manager, the engineering agency defines the unique interlocking logic that the modelers in the EVP engineering process need to manually program. The design process results in a machine readable formalization of the complete rail infrastructure and its interdependencies with exception of those unique elements and logic that cannot be standardized. The result needs to comply and/or update the open source IT system to ensure a complete and correct database that all parties in the chain are aware of. Prorail validates the data by means of a comparison test; enabled by the IT system. In contrast, new interlocking projects have no reference points for comparison. Therefore new interlocking projects' validation can e.g. occur by means of simulation.

One could argue whether an engineering agency still needs to accomplish this process instead of the infrastructure manager. First, the lean 'waste' analysis in section 3.3 identified that multiple engineering agencies would weaken data reliability but there is no indication that they should be eliminated from the process. Second, the signaling system requires to meet high safety standards. Prorail's validity acts as a second opinion to ensure those standards. Third, engineering agencies' exclusive knowledge requires the need to involve them in the eventual transformation process.

This design topology process covers most of the former RVTO and SWOD processes. The big advantage in terms of process elimination and merged processes, comes from the standardized and modular process approach. IT programs can then enable quick transformation of civil / track data into signaling system specifications.

The engineering agency needs data related to the interlocking area and the requirements imposed by the parties that initiated the interlocking project; the infrastructure manager gathers this in the data preparation process. Such data could for instance include the current and future topology of the interlocking area, civil data regarding longitudinal profiles and topographical maps. The infrastructure manager can retrieve that data from the main open source IT system. The infrastructure manager can then choose to adjust this data by combining or aligning it for the purposes of the project: e.g. add a new track or com-

bine interlocking areas. The result gets send to the engineering agencies in a general accepted format.

The data preparation process covers the current RVTO final design statement and FIS system requirements process. The new structure reduces complexity and increases reliability by the introduction of a single data source and IT tools to generate the track topology of the project's interlocking area. A straightforward process structure remains instead of the separate development of several maps and reports. Furthermore, the development of such topological track maps from scratch was considered an ambiguous process as civil track data is available.

The "Client Requirement Specification" (CRS) and "Functioneel Integraal Systeemontwerp" (FIS) largely combine into one process that defines the signaling project. An incremental process structure seems the most logical solution to prevent the need for various readjustments in the FIS due to changes in the CRS. A decision making process based on incremental phases of progress with a strong focus on problem definition could lead to a more stable process with fewer or no need to readdress results. A rough outline first decides on the goals, second the requirements, third the available means, fourth the alternatives, fifth a small selection of alternatives, sixth one alternative and seventh a more detailed description of the alternative. Comparable decision making processes occur when planning large infrastructure projects (Teisman, 2000).

This lean interlocking engineering process structure deals with all type 1 'waste' factors of Table 13. The type 3 'waste' factor of substantial hardware costs is not addressed. This may even be worsened by the use of new IT systems. Furthermore, the type 3 'waste' factors regarding productivity and utilization rates will not directly improve as well. The simpler structure and increased support of IT systems will probably improve these two factors.

4. RailML Development for the Purposes of Interlocking Engineering Design

Chapter 3 proposed a lean value chain structure of the interlocking engineering design chain. Experts (Janssen, 2012; Koelewijn et al., 2013; Storck & Dragt, 2013; Wisotzki & Muehlhause, 2013) believe that RailML could be the catalyst to achieve that lean value chain. In order to assess that hypothesis, system engineering needs to state the limits of RailML's data exchange possibilities for the interlocking engineering design chain. Therefore, this chapter investigates RailML's capability to capture a real railway network (section 4.1), to exchange data required to engineer interlocking systems (section 4.2) and to cope with a different (future) signaling system: ETCS (section 4.3). Prorail proposed to model the Dutch interlocking area of Santpoort Noord in the latest version of RailML (version 2.2).

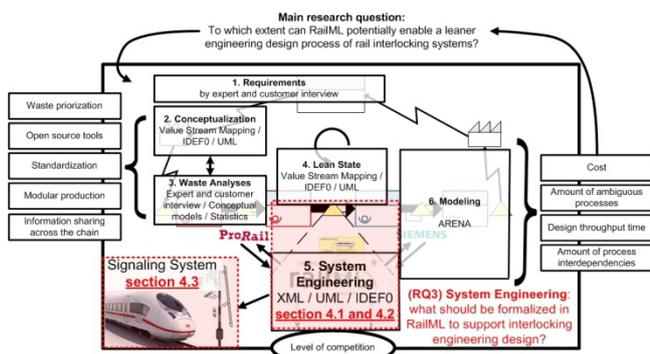


Figure 37: The red boxes visualize the research topics covered in this chapter.

4.1 The Development of RailML v2.2 for Application in the Dutch Railway Network

This section develops a RailML model to gain insight in the interlocking engineering design processes that RailML v2.2 could support and/or restructure in a lean way. In addition, Prorail requested Siemens to describe the development process of RailML for interlocking purposes. The request especially demanded a definition of the required rail data, the RailML file structure and the RailML formalization. The next subsection first introduces the interlocking area of Santpoort Noord. Subsection 4.1.2 studies the current formalization of RailML 2.2 by the RailML organization. Subsection 4.1.3 uses that formalization to define a general RailML design protocol for Prorail. Subsection 4.1.4 describes the main concepts.

4.1.1 The Santpoort Noord Interlocking Area

Santpoort Noord (or according to Prorail acronyms: Sptn) is a railway station located at the north western edge of the village Santpoort. The village is located between Driehuis and Velsbroek, about 20km west of Amsterdam and belongs to the municipality of Velsen in the province of North-Holland. Trains on the Haarlem-Uitgeest corridor provide service to this railway station.

The history of this railway station lead to a rather interesting track topology, though useful for this study. Until 1999, the station served as a node from which trains from Haarlem could not only drive towards Uitgeest, but also in the direction of IJmuiden (Figure 38). Back in the days, two train sets drove in combination from Haarlem to Santpoort Noord (Bramet, 2013). The train sets split there; one drove to IJmuiden and the other in the direction of Uitgeest and v.v.. In order to provide (freight) trains the possibility to overtake when passenger trains made a relatively long stop, Santpoort Noord got three tracks instead of the two on the connecting open track. In the years until 2013, the railway infrastructure to IJmuiden has been removed though the three track station topology remained. Figure 39 shows the general track topology; Appendix G contains the detailed topology in the form of OBE.



Figure 38: The track topology of Sptn in 1965 (SporenplanOnline, 2010)

Figure 39 shows that the interlocking area is representative for an average, medium complex (section 3.3) interlocking area as it contains various track conditions/elements:

- 6 switches (of which 4 are coupled to another);
- 10 signals;
- 2 crossovers;
- 26 detection welds; 21 detection sections (consider Appendix G);
- 2 speed regimes;
- 18 routes (relevant for interlocking in section 4.2).

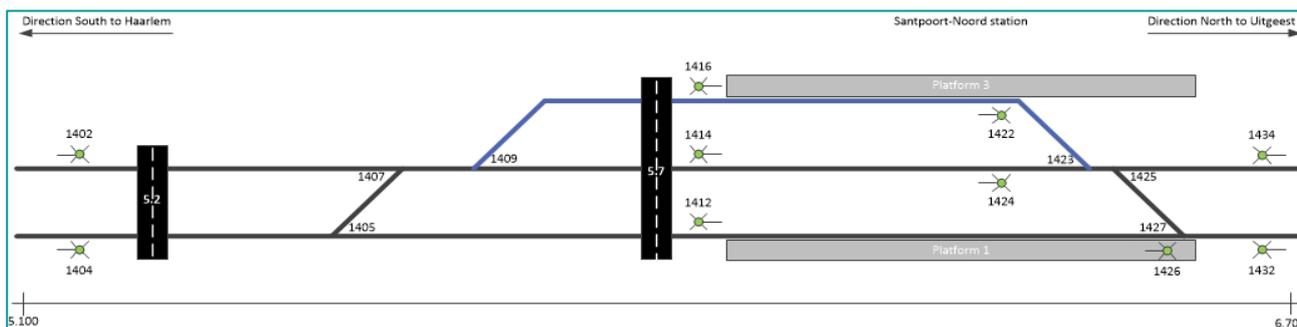


Figure 39: The current track topology of Sptn including the most notable rail elements.

4.1.2 RailML v2.2 Structure Overview

The RailML structure currently includes three rail schemas: infrastructure, rolling stock and timetable (Figure 40) (Quaglietta, 2013; RailML.org, 2013a). Infrastructure encompasses a complete description of physical and virtual rail elements that impose boundaries for the rolling stock and timetable schemas (Fries, 2003). The infrastructure schema includes those rail elements to cope with the track bound limitation of railway operations. This goes beyond detectors, switch positions and signals: also data on for example gradients, speed policies, curve radii, coordinates and so forth form an essential part of the schema. The infrastructure schema describes track positions according to nodes and branches (Fries, 2003; Schut & Dragt, 2013). This results in a schematic overview of tracks with discrete curve and gradient data.

The rolling stock schema represents the characteristics of vehicles that move on the tracks (Fries, 2003; Lehmann & Albrecht, 2008; Quaglietta, 2013). There are two essential types of rolling stock: the vehicles that propel themselves and those which cannot propel themselves. The schema includes straightforward static data for both vehicles on vehicle length, mass, gauge, type, country acceptance certificates and so on. The vehicles that can propel themselves distinguish themselves by acceleration, deceleration and safety system characteristics.

The timetable schema shows a realizable pattern of train movement over a section of track. Realizable means that the trains will not physically interact. The timetable therefore reflects the sequence of trains that use the infrastructure as described in their associated schemas.

Besides the infrastructure, rolling stock and timetables schemas, a fourth schema takes care of infrastructure visualizations (Lehmann & Albrecht, 2008). This schema positions rail elements relative to each other on a computer to create a schematic overview for modeling purposes.

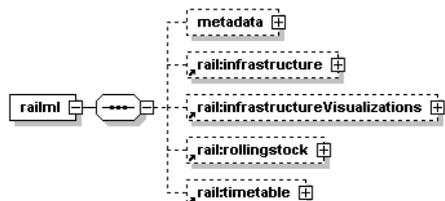


Figure 40: The main RailML structure including the three essential element structures infrastructure, rolling stock and timetable (RailML.org, 2013a).

Fries (2003) stresses the importance of a standardized RailML that every railway infrastructure manager could potentially use. This implies that each schema may not contain country specific attributes and that country specific RailML files are compatible with each other. Five challenges arise. First, the technical infrastructure equipment, e.g. detection and signal types, behaves different per mode of train operation. Second, various track allocations of block sections exist. Third, each railway system uses a different signaling system; diverse signal aspects and related information as speed limits. Fourth, there exist different ways of train control in the context of safety systems; one main differentiation exists between discrete and continuous control systems. Fifth, the RailML file should contain as little data as possible to limit the probability of errors. This for instance excludes traction, wires, construction elements of the super- and substructure and secondary infrastructure.

Hürlimann and Krauss (2003) and Nash, Hürlimann, Schuette, and Krauss (2004) explain that XML has high potential to achieve a standardized RailML format because XML can define both system characteristics as programming structure. Both aspects are essential to form a new meta language that RailML needs to become. The independent meta language enables the exchange of data between various applications and databases due to a standardized interface: Electronic Data Interchange (EDI). The authors show that XML has proven itself as a meta language in various cases. The authors also point out that import and export of XML exchange files, requires functions. The knowledge to develop those functions is however widely available as XML forms the basis of the HTML language of the world wide web.

An XML file follows an hierarchical structure in which rail elements start and end by means of a tag (Fries, 2003; Hengartner, 2003). Each root element may contain a variety of sub elements. Each sub element may contain sub elements on its own such that a tree structure for example:

```

<TrackElements> = start of root element
  <speedChanges/> = a possible sub element
  <tunnels/> = a possible sub element
  <levelCrossings/> = a possible sub element
</TrackElements> = end of root element
    
```

XML distinguishes simple elements and complex elements. Complex elements contain root sub elements; simple elements contain only contain one attribute with a consistent data type. An attribute corresponds to a specific variable characteristic, for example:

```
<speedChange id="1">           = a complex element with
                                attribute id
  <geoCoord coord="100"/>    = simple element with
                                attribute coord; no ele-
                                ments possible
</speedChange>
```

The verification of a RailML file, i.e. is the XML file also a RailML file, occurs by checking whether it matches a fixed XML schema: the XML Schema Definition (XSD) (Thompson, Mendelsohn, Beech, & Maloney, 2012). The XSD describes the structure of an XML file by the unique objects, objects' characteristics like elements and attributes and relations between objects. The XSD describes the structure in XML for easy comparison. The XSD does not demand any specific values for attributes.

4.1.3 Santpoort Noord in RailML 2.2

This subsection discusses the RailML 2.2 development of Santpoort Noord for the purpose of engineering an interlocking. This subsection focuses on the main modeling concepts and elements included in the RailML file. Appendix I contains that final RailML file; Bosschaart & Janssen (2013) also developed an extensive step-by-step modeling guide that provides the basis for this subsection.

The engineering of interlocking systems in the Netherlands focuses on the infrastructure root element of RailML. The type of rolling stock and the train sequence in the timetable does not influence interlocking logic. The infrastructure visualizations play a role for simulation of the interlocking formalization. Infrastructure visualization however only takes care of the element representation on a computer screen and therefore does not affect the interlocking logic.

The infrastructure root element contains six sub elements (Figure 41). 'Infrastructure Attribute Groups' describes some general characteristics of the railway network in question and may be filed for the purposes of recognition. The sub elements do however not contribute to interlocking functionality.

The 'Tracks' sub element describes the position of track and the related elements on that track like signals. Furthermore, the tracks sub element also describes how multiple sections of track connect. Since the interlocking requires the position and status of movable track elements, this sub element forms a vital part for the interlocking formalization in RailML.

'Trackgroups' aims to categorize tracks per corridor. This infrastructure sub element might be useful to identify the relations between multiple interlocking systems.

'Operation Control Points' group the infrastructure elements for operational reasons of traffic control. 'Controllers' define rail operation facilities positioned along the track. Both sub elements influence actual train operation

but do not influence interlocking logic; do not require a specification for this study.

The 'Speed Profiles' sub element encompasses the parameters that determine the train's speed. The interlocking system communicates speed levels via signal aspects though it does not take actual trains' behavior into account. The interlocking assumes and indirectly enforces speed restrictions. Therefore, 'Speed Profiles' play no role in the interlocking formalization.

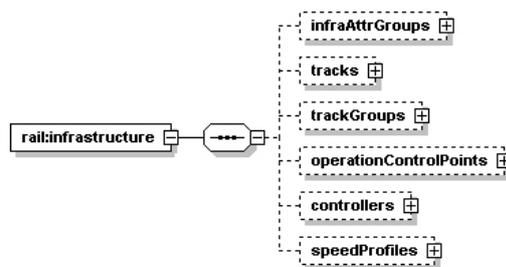


Figure 41: The six complex sub elements of the infrastructure root element shown in Figure 40.

The track sub element mainly elaborates on the geographic positioning of tracks ('trackTopology'), the non signaling related rail elements like gradients, switches and tunnels ('trackElements') and the signaling related rail elements like signals and detectors ('ocsElements'). The interlocking logic involves each of those categories because it needs to find and interlock movable track elements and communicate the result of this process to the signaling system.

'TrackTopology' modeling desires a clear definition of a track. In general, three approaches apply: a node-branch structure, a mainline-sidetrack structure and an element-to-element structure. The node-branch structure defines a track from node to node; where nodes are switches or buffer stops. The mainline-sidetrack structure defines a track from corridor start to corridor end within the boundaries of the interlocking area. The element-to-element structure defines a track from a specific rail element like a signal to the next.

The element-to-element structure may be convenient when it for example directly represents interlocking routes. The approach would however lead to overlap and consequent redundant track data. Redundant data needs to be prevented as it increases workload and the chance of inconsistencies.

The other two approaches do not lead to redundant data. Figure 42 visualizes the differences between the two. The node-branch structure's advantages over the mainline-sidetrack structure include accordance with InfraAtlas, unambiguous representation of tracks (i.e. when multiple corridors come together) and the mathematical decomposition. The mainline-sidetrack structure favors from a limited amount of track elements and ditto connections, an easier way to develop the timetable root element¹ and the higher adoption rates in RailML development.

¹ Timetable development mostly takes into account the nodes where trains can switch lines rather than all switches.

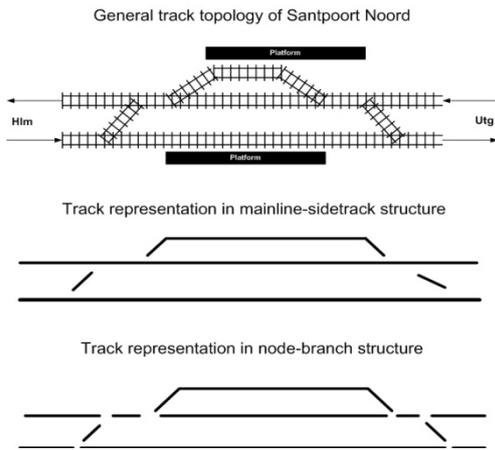


Figure 42: The differences between the mainline-sidetrack and node-branch representation of tracks in the case of Sptn. One can clearly note the difference in amount of track elements required. Each line represents a track in RailML.

This study uses the node-branch structure to represent tracks mainly due to the smooth alignment with InfraAtlas. This characteristic increases the ease in which RailML can work as a standardized data exchange tool since InfraAtlas is an essential database for the engineering design of interlocking systems. Furthermore, both track structures can be connected or combined. This ensures compatibility between RailML files of different regions/countries and the timetable input into RailML if necessary.

A 'trackBegin' refers to the side of the switch, the node, where the track starts. A switch has a front, right and left side. Accordingly, the 'trackEnd' refers to the side of the switch where the track ends. InfraAtlas uses the same approach. Further track descriptions therefore form combinations or references to those definitions to ensure consistency with InfraAtlas. A 'track' id combines the trackEnd

and trackBegin from low to high number with an area code, e.g. *Sptn_Switch1R_(anotherAreaCode)_Switch2F* instead of *Sptn_Switch2F_(anotherAreaCode)_Switch1R*. This decision benefits from a clear description and alignment with InfraAtlas. The disadvantage is however that, due to fluctuating numbering of elements in InfraAtlas, connecting pieces of track are described in contradicting direction. The modeler may find this unfortunate though IT tools and applications will not experience issues.

RailML connects tracks by using the 'trackBegin' and/or 'trackEnd' sub element 'connection' which has an id and a reference to another track's connection id. A 'track' defines a 'switch' connection when a track ends/starts at the front side of a switch. This allows for more than one connection. Figure 43 provides an example of this 'trackTopology' approach by means of connecting a fictive track between fictive switches 9 and 10 (highlighted in green) to a track between switch 10 and 11 (highlighted in red). The track starts at the right side of switch 9, "Example_9R", and ends at the front side of switch 10, "Example_10F". Then, a standard straight track connection is made to the red track between switch 10 and 11. The identification for this connection uses the track end id again plus the switch side it wants to connect to: the right side R giving id="Example_10FR" and ref="Example_10RF" (or in other words, the track end 10FR wants to connect with 10RF). The red track description does the same but the other way around resulting in a unique connection; thus providing id="Example_10RF" and ref="Example_10FR". The trackEnd 'Example_10F' has another connection however, diverging left from the original course. For that purpose, a switch sub element has been filled. The combination of two connection ids must be unique in order to describe a connection. Otherwise, the IT tools cannot distinguish between two different route possibilities.

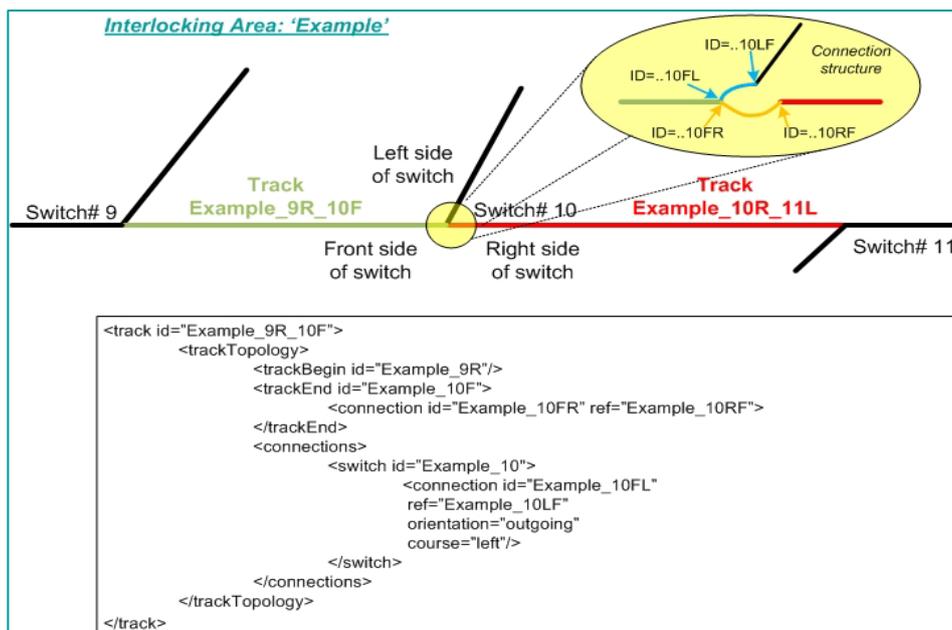


Figure 43: The 'trackTopology' sub element illustrated by means of connecting the green track between switch 9 and 10 to the red track between switch 10 and 11. The enlarged yellow oval visualizes the formalized connection structure of RailML v2.2 in more detail. The blue connection in the yellow oval uses a switch connection element; the dark yellow connection concerns a standard track connection at track end or begin.

The second sub element 'trackElements' of the 'track' element describes characteristics like position and variable changes of non signaling elements. Five 'trackElements' in RailML v2.2 may need a description for the purposes of interlocking systems: 'levelCrossings', 'tunnels', 'bridges', 'speedChanges' and 'gradientChanges'. The interlocking needs to know the existence of operational 'LevelCrossings'. The interlocking may only grant a route authority when a train activated the level crossing. The activation can occur in two ways: directly when a train requests a route or when passing a dedicated detector or track circuit section (Theeg, Maschek, et al., 2009; Theeg, Svalov, & Schoene, 2009). The Dutch network usually applies the latter method to limit the closing time of level crossings. Therefore, the interlocking may only change the start signal aspect when the train has passed such an activation point. RailML however does not include elements that indicate those activation points. 'Tunnels' and 'bridges', i.e. movable bridges, work the other way around. An interlocking may always grant a route unless the tunnel blocks the entrance, e.g. by a water surge barrier, or the bridge opens. Therefore, one may debate whether to include 'bridges' or 'tunnels' for the purpose of interlocking since they impose a rather static track obstruction. The fact that traffic control may never open bridges or close tunnels when an interlocking granted a route across those elements, speaks in favor of including both for the purpose of engineering interlocking systems. Interlocking systems only require a description of 'speedChanges', i.e. in the infrastructure root element, and 'gradientChanges' when using ETCS. ETCS requests routes on the basis of dynamic speed and braking curves.

The identification of the track element types, with the exception of level crossings, follows InfraAtlas. InfraAtlas defines track elements by a combination of a switch number, switch side and an element sequence number. InfraAtlas does not use a mathematical logic to define which of two switches on a track defines an element. Level crossings form an exception as they do not get numbered in InfraAtlas. The mileage is proposed for that purpose. An additional area code makes the track element descriptions unique for the Netherlands. The next provides an example of the 'gradientChange', 'speedChange' and 'levelCrossing' elements in the Sptn interlocking area:

```
<track id="Hlm_137aL_Sptn_1407R">
  <trackElements>
    <gradientChanges>
      <gradientChange id="Sptn_1407R_1200"
        pos="X" name="1407R1200"
        absPos="5.135" dir="up" slope="-0.190"
      />
    </gradientChanges>
    <speedchanges>
      <speedchange id="Sptn_1407R_200"
        pos="x" absPos="5.416" vmax="100"
        dir="down" />
    </speedchanges>
    <levelCrossings>
      <levelCrossing
        id="Hlm_137aL_Sptn_1407R_5.290"
        pos="x" absPos="5.290" />
    </levelCrossings>
  </trackElements>
</track>
```

The third sub element of the 'track' element concerns specifically with signaling elements of which 'signals', 'trainDetectionElements' and 'balises' might involve the interlocking system. Signals obviously play an important part for mainly two reasons: one, they usually determine the start and end of an interlocking route, two, interlocking systems often communicate and manage signal aspects. These arguments however depend on the train control system, e.g. ETCS level 3 in theory does not need static signals at all. There remains one other reason to always define some (virtual) signals though: to distinguish open track and station interlocking areas. An interlocking system needs to know when to communicate with a neighboring interlocking system on the basis of an approach time, a distance or an amount of block sections. Current practice in the Netherlands distinguishes both interlocking areas by the borders of the OBE maps (Prorail, 2010c). RailML does not distinguish between OBE maps anymore and a virtual signal provides the opportunity to do so. The Sptn RailML represents signals by their number, as does InfraAtlas. Virtual signals are represented like most track elements which coheres to the InfraAtlas approach of defining map borders. The next shows an example of the chosen signal representation in XML:

```
<track id="Hlm_137aL_Sptn_1407R">
  <ocsElements>
    <signals>
      <signal id="Sptn_1402" pos="X"
        absPos="5.135" name="1402" dir="up" />
      <signal id="Sptn_1407R_1600" pos="x"
        absPos="5.100" dir="down" virtual="true"
        description="Open track to station area
        transition"/>
    </signals>
  </ocsElements>
</track>
```

'TrainDetectionElements' describe the locations of the devices that detect trains. The interlocking needs to read the results from the detection devices in order to check whether a requested route contains rolling stock. Those devices detect trains on the basis of track circuit sections or by counting train axes. Sptn and the Dutch Railway network in general, use track circuit sections with the exception of some corridors. InfraAtlas identifies track circuit borders that define the edges of a track circuit section, like most track elements. In addition, InfraAtlas defines the circuit sections mostly by the switch number (i.e. when it is adjacent to a unique switch) plus a sequential letter:

```
<track id="Hlm_137aL_Sptn_1407R">
  <ocsElements>
    <trainDetectionElements>
      <trackCircuitBorder
        id="Sptn_1407R_300"
        pos="X"
        name="1407T_1414CT"
        absPos="5.416"/>
    </trainDetectionElements>
  </ocsElements>
</track>
```

'Balises' potentially provide a useful addition to 'TrainDetectionElements' in the context of interlocking systems. Comparable to 'speedProfiles' and 'gradientChanges', balises deliver necessary (train loca-

tion) data for the ETCS train control system to request routes from the interlocking system.

The second ‘infrastructure’ root element of interest for interlocking engineering design ‘trackGroups’, categorizes tracks into a line set. An interlocking system or RailML interlocking formalization may use this to distinguish separate interlocking areas and/or for track (element) relative positioning. In the case of Stpn, all tracks defined form a part of the Haarlem – Uitgeest corridor.

In short, the next XML schema shows the possible RailML v2.2 structure relevant for interlocking systems in the Dutch Railway Network:

```
<railml>
  <infrastructure>
    </infraAttributes>
    <tracks>
      <trackTopology>
        <trackBegin>
          </connection>
        </trackBegin>
        <trackEnd>
          </connection>
        </trackEnd>
        <connections>
          </switch>
        <connections>
      </trackTopology>
      <trackElements>
        </gradientchanges>
        </speedchanges>
        </levelCrossing>
        </bridges>
        </tunnels>
      </trackElements>
      <ocsElements>
        </signals>
        </trainDetectionElements>
        </balises>
      </ocsElements>
    </tracks>
    <trackGroups>
      <line>
        </trackRef>
      </line>
    </trackGroups>
  </infrastructure>
</railml>
```

4.2 The RailML Interlocking Formalization

This section develops a formalization that provides, together with the RailML v2.2 description in the previous chapter, the opportunity to exchange data to engineer an interlocking system. Subsection 4.2.1 summarizes the literature that already elaborated on a possible way to formalize interlocking in RailML. Subsection 4.2.2 describes the process that led to the interlocking formalization. Subsection 4.2.3 concludes this section with the final XML interlocking formalization.

4.2.1 Current Status of the Interlocking Subschema in RailML

Lehmann and Albrecht (2008) proposed a RailML interlocking formalization as subschema of the infrastructure schema. They aim to capture the relations between those rail elements necessary to enable safe routes for trains.

The study was initiated to measure the performance of various interlocking systems by means of simulation.

Lehmann and Albrecht approach the interlocking formalization from the viewpoint of the hardware. Therefore, they define the “mainSignalBox” as the attribute of interlocking which has, besides standard tags, a “type” (e.g. relay, mechanical and electronic) an “operationControlCenter”, an “ownControlRange” and a “subSignalBox”. The “operationControlCenter” allows for the possibility to link an interlocking to a dedicated control center. The “ownControlRange” provides the opportunity to define specific sections within an interlocking area that may operate as an interlocking area on its own. This function becomes obsolete for an electronic interlocking system. The “subSignalBox” contains mainly three sub elements relevant for capturing interlocking functionality: “routes”, “trackSections” and “overlaps”. Routes include a list of possible route options. TrackSections represent trackfree detection points. Overlaps include data on the location of danger points and overrunning distances of signals. Lehmann and Albrecht only designed the “route” schema in more detail since the other two did not seem very relevant for simulation purposes.

A “route” contains the track characteristics for a certain section and the interdependencies with other rail elements. The authors, focusing on the German railway system, define a route as long as the block section; this counts for the Dutch railway system as well (in Dutch: “enkelvoudige rijwegen”) (van der Meij, van der Werff, Janssen, Bartholomeus, & Dragt, 2013). A block section could however contain multiple route possibilities. A route typically refers to a rail element defining its start and its end. A route’s attributes include maximum speed, distance, interlocking setup time and interlocking release time. Special sub elements include references to rail elements, flank protection elements, overlap and relevant track sections. The next summarizes Lehmann and Albrecht’s interlocking schema structure:

```
<RailML>
  <infrastructure>
    <interlocking>
      <subSignalBox id="x" name="y" type="z">
        <routes id="x" vmax="y">
          <routeStart/>
          <routeDestination/>
          <elements/>
          <flankProtectionElements/>
          <overlaps/>
        </routes>
        <trackSections/>
        <overlaps/>
      </subSignalBox>
      <ownControlRange/>
    </interlocking>
  </infrastructure>
  <timetable/>
  <rollingStock/>
</RailML>
```

Lehmann and Albrecht explain that the German railway system tags every rail element in the same interlocking area with the same code. This rule allows easy identification of interlocking areas and open track areas. Currently, rail elements only have a connection with a track. The authors propose to study a connection between those rail

elements and the interlocking by means of an attribute with reference tag.

Fries (2003) proposes the use of an interlocking database with relations to RailML instead of an interlocking formalization in RailML. Fries proposes a database like MS ACCESS because of probable redundant data issues. In order to minimize the chance and risk of inconsistencies and the use of old data files, Fries believes that an IT tool needs to manage and track changes. RailML cannot do that. Schut and Palacios, both working on (rail) IT tools, acknowledge this issue (Palacios, 2013; Schut & Dragt, 2013).

As a consequence of Fries' approach, a conversion tool needs to transform the RailML infrastructure data into the interlocking database. The conversion tool can find the elements along a defined track and determine the relations between them. Fries mentions five challenges with this approach. First, the process could contain errors and the database has no means to find those errors itself. Second, RailML positions itself at meta level of detail to connect with a wide variety of databases, applications and output files. As a result, it may occur that RailML will sometimes not contain data at a micro level of detail necessary for a certain application as interlocking. Third, the conversion time may become substantial for large interlocking areas as the conversion tool needs to find every route with associated rail elements. Fourth, a database has more difficulties to capture some track descriptions compared to RailML. Fries provides some examples: the limited amount of variable conditions and the distinction between tracks with double track or more. Fifth, the database format does not make real connections between elements like RailML does with tracks, switches and so on.

For a few years, a group of representatives of Siemens, Alstom, Signon, Thales, Infrabel, OBB, SBB and the ON-time project group work on the official formalization of an interlocking subschema in RailML (RailML.org, 2013b). At their last meeting in February 2013, they agreed on a UML object relation structure that might form the basis of the interlocking subschema (Appendix A). The RailML group thus follows the approach of Lehman and Albrecht to include interlocking in RailML instead of using the separate database approach of Fries. In contrast to Lehman and Albrecht however, the RailML group formalizes the interlocking structure from the viewpoint of requested train routes instead of the interlocking hardware.

The RailML group's approach has three main disadvantages for application in interlocking engineering design:

- Redundancy: especially the inclusion of start-target aspect combinations makes the formalization redundant because target aspects equal start aspects at the subsequent signal
- Aspect relations: signal aspect dependencies not only depend on aspects but also on speed profiles. As such, the formalization misses certain dependencies and it is difficult to capture speed changes at static speed signs
- Over complete / Opaque: from the viewpoint of interlocking engineering design it is questionable whether all route elements, signal aspect groups and signal types need a definition

The RailML interlocking group also defined how the UML translates to a (temporary) XML structure; Appendix J contains an XML example of Sptn:

```
<interlocking>
  <interfaces>
    </interface>
  </interfaces>
  <routeGroups>
    </routeGroup>
  </routeGroups>
  <routes>
    <route>
      <start>
        </signalRef>
        </trackTerminationRef>
      </start>
      <target>
        </signalRef>
        </trackTerminationRef>
      </target>
      <segments>
        <segment>
          <element>
            </signalRef>
            </trackSection>
            </switchRef>
            </crossingRef>
            </derailerRef>
            </trainDetectorRef>
            </levelCrossingRef>
          </element>
          <flankProtection>
            </signalRef>
            </trackSection>
            </switchRef>
            </crossingRef>
            </derailerRef>
            </trainDetectorRef>
            </levelCrossingRef>
          </flankProtection>
        </segment>
      </segments>
    </route>
  </routes>
  <signals>
    </signal>
  </signals>
  <signalTypes>
    </signalType>
  </signalTypes>
  <signalAspects>
    </signalAspect>
  </signalAspects>
  <signalAspectGroups>
    </signalAspectGroup>
  </signalAspectGroups>
  <signalAspectDependencies>
    <signalAspectDependency strTypeRef="" trgSignalTypeRef="">
      <dependency strSignalAspect="" trgSignalAspect="">
        </routeDescriptionAfterTargetSignal>
      </dependency>
    </signalAspectDependency>
  </signalAspectDependencies>
</interlocking>
```

4.2.2 The Development of the RailML Interlocking Formalization for Sptn

A four step approach leads to a rational, top-down way of formulating an interlocking XML structure for RailML:

1. determine the route interlocking process for a 2-block 3-aspect, continuous signaling with braking supervision system as the Dutch one;
2. apply this to an actual interlocking area (Sptn);

3. discover and formalize which data/information the interlocking schema should encompass;
4. reduce the formalization to the core necessities and align with RailML group's definitions.

1) Determine the route interlocking process for a 2-block 3-aspect, continuous signaling with braking supervision system as the Dutch one

Theeg, Maschek, et al. (2009) explain that interlocking systems can either interlock a route in a reversible or an irreversible way. Both methods require the correct position of movable track elements before the start signal may show a proceed aspect. Reversible route locking however implies that a signaller may directly change the position of movable track elements when the start signal aspect returns to stop. Theoretically, a train could still drive the route while changing the elements. Irreversible route locking prevents that scenario and demands that only a train can release a route when crossing a certain point. Alternatively, special release options exist like a timer. Irreversible route locking comes in two ways: before or after signal clearance. SIMIS W in the Netherlands applies the irreversible route interlocking before signal opening method. In that case, Theeg, Maschek, et al. (2009) distinguish nine steps in the interlocking route process (Figure 44).

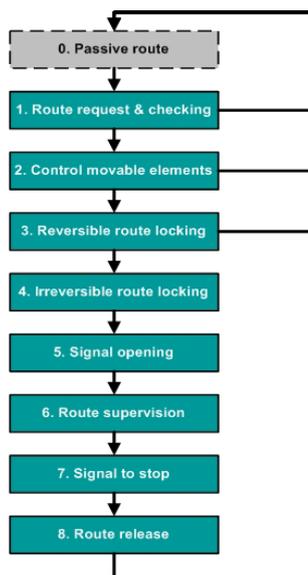


Figure 44: The process steps to interlock a route according to the method that irreversibly locks a route before signal clearance (Theeg, Maschek, et al., 2009).

The interlocking system starts interlocking a route when a train demands a route. In the Netherlands, multiple routes can be part of one block section; a block section equals a combination of two subsequent signals connected by rail in a direct way for operation. Ideally, the interlocking system interlocks a route two block sections in advance to ensure an optimal speed profile and track capacity in a 2-block 3-aspect signaling system like ATB (Wiggenraad, 2012). Figure 45 illustrates that a route request at two blocks distance allows a train to pass at the maximum allowed speed, assuming an available route in which a train can always come to a complete stop. In reality, a train

can and sometimes must, e.g. at a terminal track / station, request a route only one block in advance. As a result, the signaling system could limit the train's speed to 40km/h for the next block section.

Upon receiving a request to interlock a route, an interlocking system will first check whether another train already claimed (a part of) the route. Furthermore, the interlocking system will read the measurements of the detection system to exclude the possibility of rolling stock on the requested route. The interlocking will then search the route to control the movable track elements. Before or after the interlocking reversibly changed the movable track elements on route, it also needs to ensure flank protection. In the case of danger from the flanks, the interlocking needs to either alter additional movable track elements or demand signal aspects at the flanks. Table 7 provides some examples.

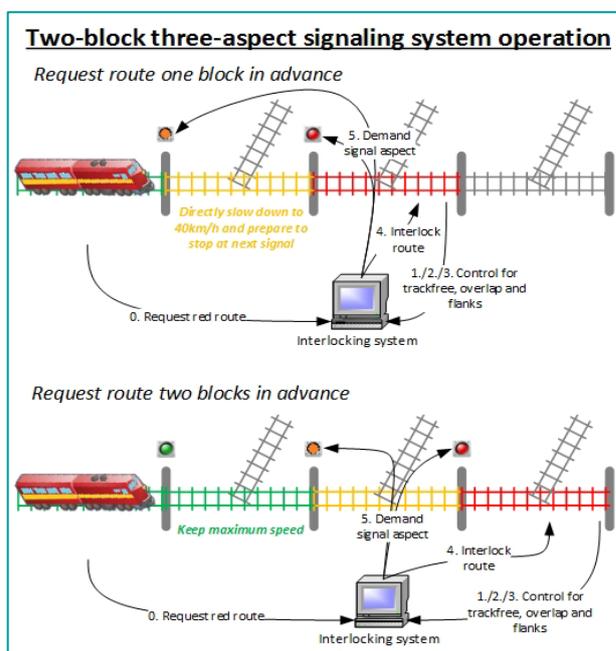


Figure 45: Illustration that compares train requests one block in advance with train requests two blocks in advance for the Dutch railway network with a 2-block 3-aspect signaling system.

In the case of a clear track, safe flanks and correct position of movable track elements, the interlocking will irreversibly lock the movable track elements. In addition, the interlocking needs to communicate this status to the signaling system. The communication occurs by means of signal aspects. Literature sometimes distinguishes the interlocking from signaling (Trinckauf, 2009). This distinction is probably a remainder of the past in which signallers changed movable track elements by hand (the interlocking) and could then change the signal arm by hand: two different processes (Theeg, Nasedkin, et al., 2009). Nowadays in reality, track equipment only contains an interlocking system to both control movable track elements as signals (Janssen & Quaglietta, 2013). Theeg, Maschek, et al. (2009) confirm the same in their description of interlocking route's life cycle. This makes sense as a signal in a station area must know the status of track elements and route requests of trains in order to depict the right signal

aspect. The interlocking formalization therefore needs to include signal aspect dependencies as the to-build system requires those relations in reality. One could however still speak of a separate signal system in the sense of the aspect types and interrelations allowed per e.g. country.

When the interlocking may show an aspect better than red, the result is an active route on which a train may traverse. As section 4.1 explained, a level crossing gets activated later than the route request to minimize waiting times. Figure 45 explained that the interlocking ideally communicates the best status of the start signal two blocks in advance. Activating the level crossing two blocks in advance results in long closing times; therefore this happens via activation detectors somewhat more than one block from the level crossing. As a result, level crossing activation happens after signal clearance. Besides activation of level crossings, the route supervision step also keeps track of train dynamics at the end of the route. In case flank issues might arise, the interlocking could still change the end signal aspect to red in the hope this prevents a conflict.

Eventually, when a train successfully leaves the requested route, the route becomes passive and another train may request the route again. Then the process repeats itself. The interlocking does not need to change the start signal aspect as the Dutch signals do this automatically on the basis of neighboring signals and switches.

Figure 46 illustrates that the interlocking system controls a large part of the signaling system in Figure 3 by controlling signals, controlling movable track elements, providing input to the train protection system, providing traffic control with track occupancy and reservation statuses and by communicating with neighboring interlocking systems. A neighboring interlocking system can either be a neighboring station area or the open track.

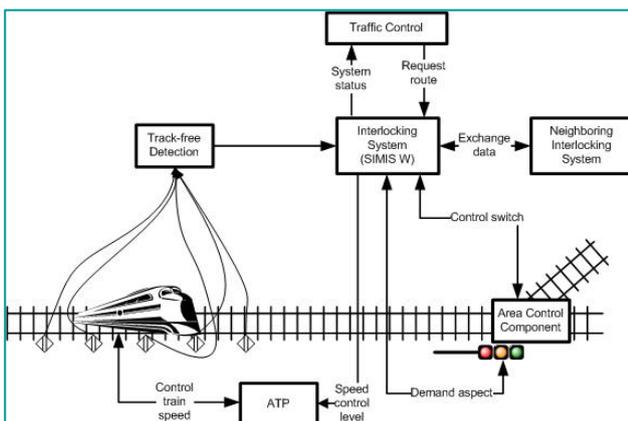


Figure 46: The signaling system structure with ATB-EG in the Netherlands. Each block represents a (sub-)system; each line represents the direction and type of data communication. The diamonds represent detection points in the form of track circuit sections or axle detectors. An area control component stands for a (group of) rail element(s).

2) Apply (1) to an actual interlocking area (Sptn)

A route request at the Sptn interlocking area shows the implication of the interlocking procedure in the previous paragraph. Assume the hypothetical situation that a train requests a route from signal 1404 to signal 1422 since it wants to pick up passengers at platform 3 (please refer

Figure 39 and Appendix G for the track topology and rail elements). Then, a train would request the route two signals earlier at signal 501 in the Haarlem area.

First, the interlocking system needs to check whether this route already conflicts with another active route. Possible conflicts could be a route from signal 1434 to signal 1416. In addition, the interlocking system needs to ensure that no rolling stock is present on this route by reading the status of the track circuit sections. No further elements could constrain the route in the absence of water surge barriers, movable bridges and derailleurs.

Second, the interlocking needs to put three movable track elements, all switches, in the correct position. This means that switch 1405, 1407 and 1409 should all go in bended / deviated position.

Third, the flanks of the train need protection from other trains (Figure 47). Flank collisions could potentially occur from train movements starting at signal 1402 towards signal 1424/1422, train movements train movements starting at signal 1412 heading towards signal 132 and train movements starting at signal 1414 heading towards signal 132/506. Separate flank protection measures are not required in this case. A route from signal 1404 to 1422 directly prevents the possible flank movements by fixing the switches and putting the consequent start signals to stop (e.g. for a route from signal 1402 to 1424). Train movement from signal 1412 or 1414 in contradicting direction, cannot be interlocked due to incompatible switch positions. Furthermore, a route from signal 1432 or 1434 to signal 1416 causes overlap with the interlocked route 1404 to 1422. The consequence is that these routes cannot be interlocked either; protecting all possible flank movements of route 1404 to 1422.

Fourth comes irreversibly locking the route. The interlocking should as fifth communicate a signal aspect to the starting signal 1404. Each area has their own set of signal aspect dependencies. The relations mostly depend on local speed restrictions that result in a different signal aspect. Following the example, the OS map (Appendix G) indicates two possible signal aspects: yellow and green flashing. The interlocking needs to communicate the yellow aspect when the target signal has a red or yellow flashing signal. The interlocking needs to communicate the green flashing aspect when the target signal has a yellow, yellow 4, green or flashing green aspect.

Sixth, the level crossings need to be activated by the train; this occurs over a block section in advance.

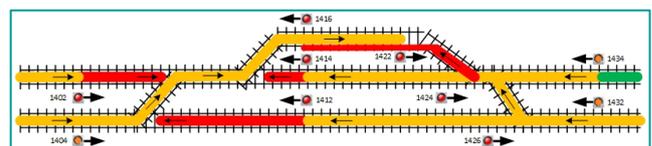


Figure 47: This figure shows that separate flank protection measures are not necessary when interlocking route 1404 to 1422. The lines' colors correspond to signal aspects.

The Sptn area does not contain a less common task for the interlocking system: route decision making. Figure 48 shows that some interlocking areas allow multiple routes between two signals. Therefore, the interlocking should know or determine which route to interlock. Currently, this occurs by means of a route hierarchy, i.e. experts deter-

mine that one route is preferred over alternative routes (Theeg, Maschek, et al., 2009). The interlocking system first needs to check the above process for the preferred route. When that route cannot be interlocked, e.g. because of overlap, the interlocking needs to check for any other alternatives. The interlocking refuses the route request when it cannot find an alternative.

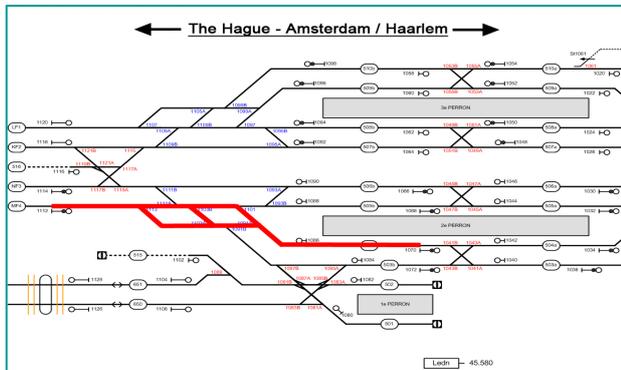


Figure 48: The figure shows the interlocking area of Leiden. The red line indicates that a route from signal 1112 to signal 1070 could follow three different routes (SporenplanOnline, 2010).

3) Discover and formalize which data the interlocking schema should encompass

On the basis of the previous and section 4.1, the following data/information require a formalization as they cannot and should not be extracted from the RailML v2.2 infrastructure schema for the purposes of interlocking engineering:

- RailML does not indicate the detector or track circuit border that announces a route request.
- RailML does not indicate the status of movable track elements, e.g. a switch's position and the closing of water surge barriers.
- RailML does not indicate the status of level crossings.
- RailML does not provide the required status of track-free detection.
- RailML does not indicate tracks' relations to other tracks and corresponding rail elements for the purposes of flank protection.
- RailML does not provide a mean to choose between various route options.
- RailML does not contain signal aspect dependencies for each possible route.

The data gaps fit into the sequence of process steps for route interlocking in Figure 44. A route based formalization approach as proposed by the RailML group thus seems sound. A twist to the route based approach is given from the train's perspective to request a route.

The Sptn interlocking example illustrates that a route starts at a dedicated signal and ends at a dedicated signal. One begin signal could have multiple route activation request signals as Figure 49 shows.

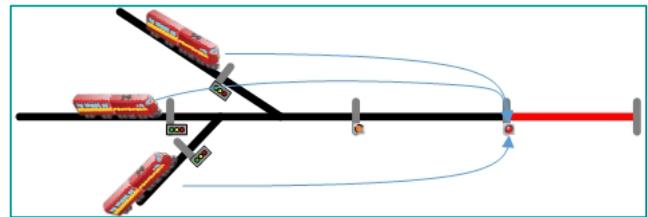


Figure 49: A track topology in which the red route can be activated from three different points.

The interlocking needs to control route specific rail elements, i.e. belonging to a signal begin and signal end pair. Furthermore, the interlocking needs to control route specific movable track elements on the flanks, if any. In addition, each pair of begin and end signal may have multiple route alternatives (Figure 48) and consequently one preferred route.

Besides inclusion of route elements, the interlocking formalization also requires signal aspect dependencies in order to determine the aspect of the begin signal. A begin aspect depends on the type of begin signal and the target signal. The begin signal type determines the total amount of possible aspects. A target signal enforces a speed profile thanks to e.g. topology. Furthermore, the target signal's actual aspect may degrade the speed profile and consequent begin aspect as well.

Figure 50 shows the result of this route based object relations in the form of an UML. Appendix J provides the XML structure for a Sptn interlocking area route from signal 1404 to 1422.

Another way to develop the interlocking formalization is from the perspective of the signals. Assuming that the interlocking system is smart enough (read: a geographic approach based on node-branch models) to find the relevant rail elements on a route from the infrastructure description, it only needs to know the relations between the aspects of two subsequent signals. In the Netherlands, these relations always depend on the interlocking area and cannot be derived from rules or algebra (van der Meij et al., 2013). What an interlocking can determine based on rules, is the effect of one signal aspect, e.g. yellow demands to stop before the next signal. Then, on the basis of a signal's aspect code, approach speed profile and train's target, all signal aspect dependencies follow as shown in Figure 51, Figure 52, Figure 53 and Figure 54. The result corresponds to the Dutch OS maps.

Figure 51 shows the signal aspects with corresponding targets and approach speed profiles of signal 1. Furthermore, all red aspects do not have a signal aspect dependency.

Figure 52 then shows that the exit speed profiles follow on the basis of the entrance exit speed profiles and corresponding signal aspects.

Figure 53 accordingly links the exit speed profiles of signal 1 to signal aspects of signal 3. Then, the exit speed profiles of signal 3 can be determined as well.

Figure 54 does the exact same for the relation between signal 1 and signal 2. This procedure leads to a complete representation of an OS map.

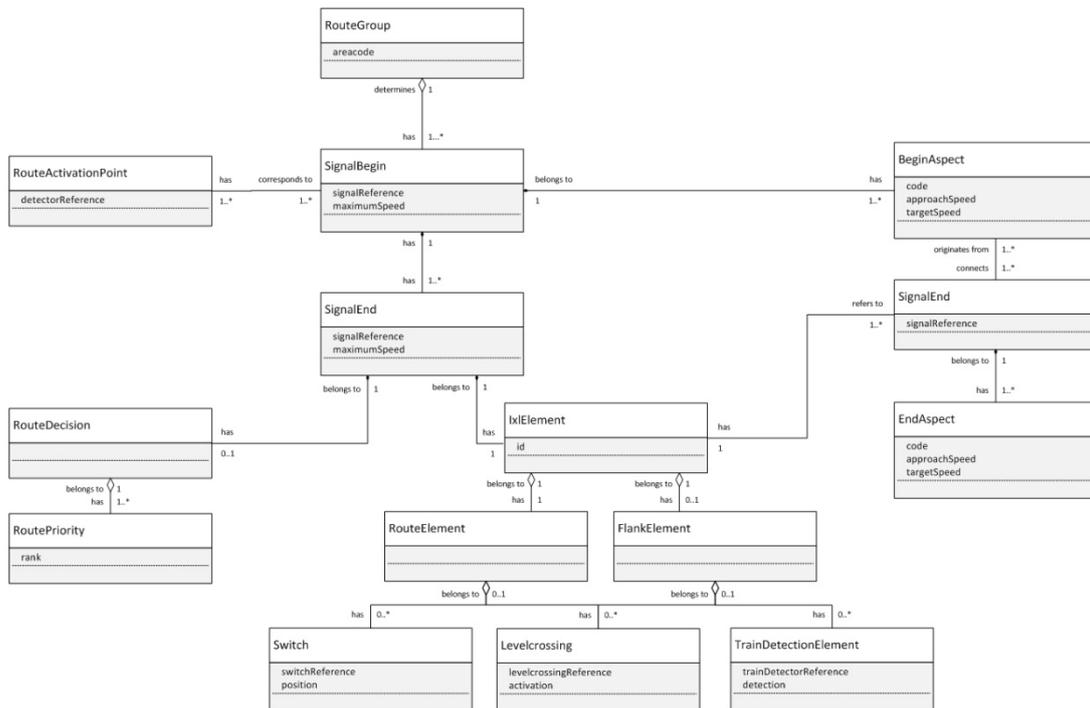


Figure 50: RailML interlocking formalization's object relations on the basis of a route approach from the perspective of a train. The classes correspond to a Sptn route from signal 1404 to 1422.

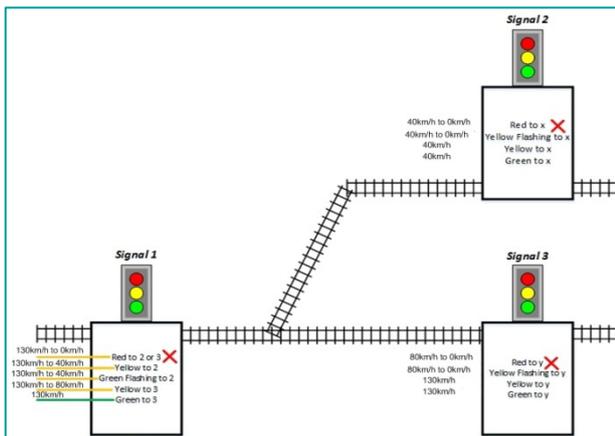


Figure 51: Part 1 of a fictive track topology to illustrate the formalizing approach for the signal aspect dependencies.

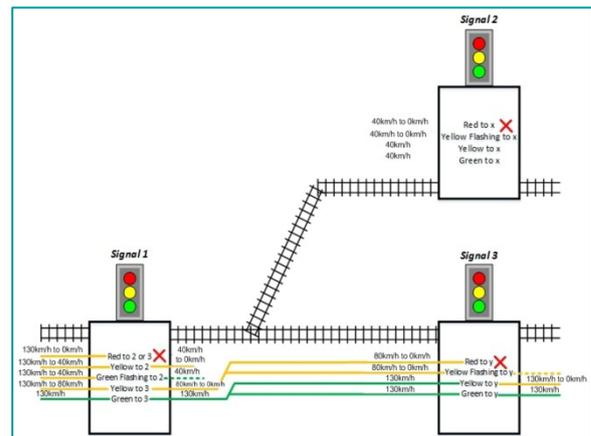


Figure 53: Part 3 of a fictive track topology to illustrate the formalizing approach for the signal aspect dependencies.

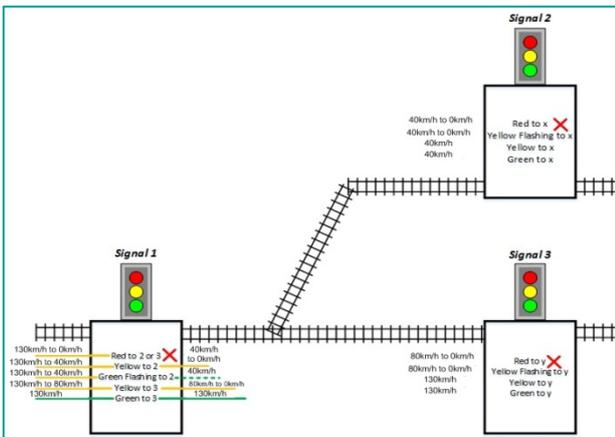


Figure 52: Part 2 of a fictive track topology to illustrate the formalizing approach for the signal aspect dependencies.

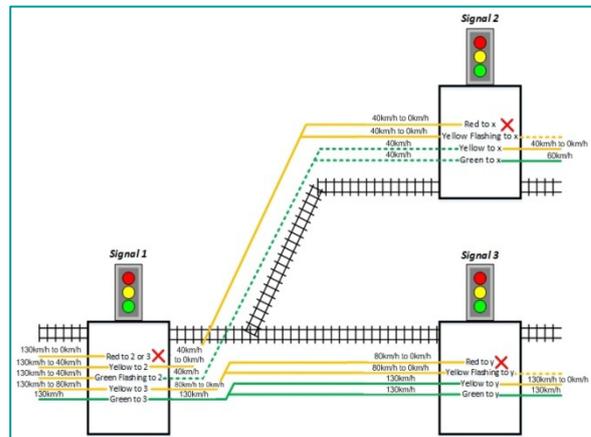


Figure 54: Part 4 of a fictive track topology to illustrate the formalizing approach for the signal aspect dependencies.

The signal dependency approach thus only describes signal aspects per signal in the interlocking area to capture the interlocking process. Each aspect does however at least require the aspect code, speed profile and target

signal for a complete picture. Figure 45 applies this method as shown in the last four pictures to the OS of Sptn. Appendix J contains an XML version of this approach for the route from signal 1404 to 1422 at Sptn.

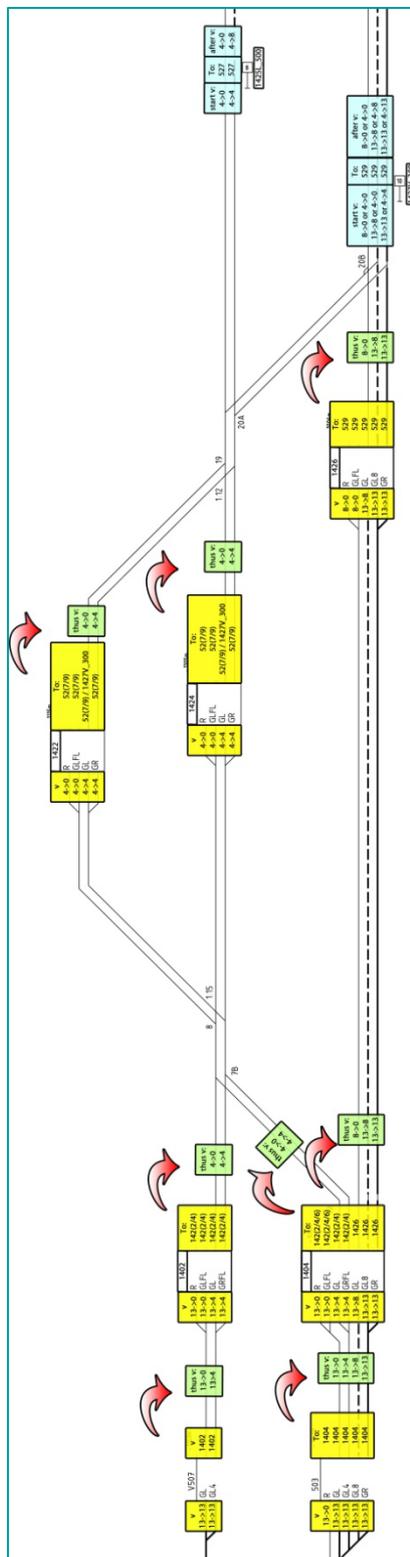


Figure 55: The most recent OS map of the Sptn interlocking area in the direction from Hlm to Utg. The yellow boxes show, including with the signal aspects, the necessary entries for the interlocking formalization of signal aspect dependencies. The green boxes show how logic enables the unique identification of the speed profile lines in the OS map. The blue boxes show the differences in speed profile enforced by static speed signs along the track instead of signals.

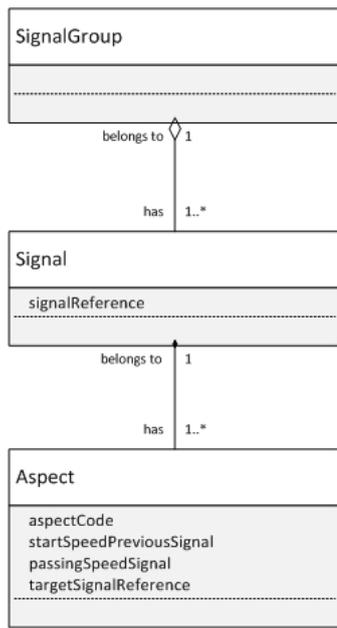


Figure 56: The UML for a RailML interlocking formalization on the basis of just signal aspect dependencies.

4) Reduce the formalization to the core necessities and align with RailML group’s definitions

The route based approach formalizes the interlocking system in a comprehensive way when compared to the signal approach. Furthermore, the route based approach includes redundant data. The signal approach namely showed that the speed profile, target and possible aspects per signal are sufficient to depict all signal aspect dependencies. In contrast, the route based approach also includes static maximum speeds and speed profiles with corresponding signal aspects of the route. In addition, the reference for each signal aspect dependency to the rail elements does not add additional information either. That is to say, the route formalization already groups the elements per signal end and thus block section. Also, Figure 46 states that traffic control demands routes from the interlocking which makes a routeActivationRequest element superfluous.

The signal aspect approach formalizes the signal aspect dependencies in a very concise way. The conciseness also limits transparency. The targetSignalReference as sub element instead of an attribute would directly distinguish the amount of different aspect codes per unique approach speed profile. This categorization is more convenient for face validation as it represents the possible input signal aspects; just as an OS map depicts them. When these do not correspond with the output of the preceding signal(s), a modeler can quickly take corrective action.

Besides some lack in structure transparency, the signal based formalization misses a description of rail elements and possible route priorities. The interlocking formalization requires the inclusion of at least flank protection elements because section 3.2 found that some flank requirements do not allow for standardization. In addition, probably not every engineering and simulation tools can find rail elements along the route on the basis of a node-

branch model. As a result, either the RailML interlocking schema needs to formalize all movable track elements per block or all tools require to change. Besides rail elements, experts and not logic determine route priorities. Therefore, route priorities demand a formalization as well.

Figure 55 also indicates that static speed signs need a formalization since they affect the ATB code. One could formalize those as virtual signals, although a static speed sign does not behave according signal. Signal’s aspect codes unambiguously determine the output speed profile, but a static speed sign’s speed level does not unambiguously determine the output speed profile (e.g. sign ‘1427V_300’ in Figure 55). Furthermore, two subsequent signals indicate a block section while a static speed sign does not. Therefore, signals should not follow the same object structure as static speed signs.

In order to mitigate the disadvantages of both approaches, the signal based formalization, including a slight restructuring with the targetSignalReference as sub element and the formalization of static speed signs, forms the basis of the new aspect dependency schema within the interlocking formalization. TargetSignalReference changes to targetReference since it could either connect a signal to a signal or a speed sign or the other way around. The speedChangeSignGroup covers the speed signs that affect the ATB codes; when the code does not change, RailML does not need to include it a speed dependency. The structure of the speedChangeSignGroup follows that of the signalGroup with the exception that both the start and after speed profile compared to the sign need a definition. The ixElement class from the route based approach complements the aspect dependency schema to include the route and flank elements and route priorities. Figure 58 shows the UML of the final structure. The elements use the names of the RailML group as much as possible.

4.2.3 The Interlocking XML Structure

The next shows the interlocking schema that aligns with the UML in Figure 58; Figure 57 provides the proposed new RailML root schema. Appendix J presents an interlocking formalization for the Santpoort Noord interlocking area in the direction of Uitgeest. The formalization is based on the older RVTO OS map in Appendix G and not the OS of Figure 55 because Prorail delivered the RVTO OS map to formalize.

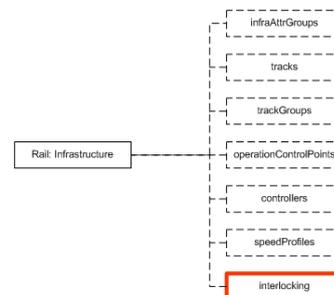


Figure 57: The root RailML elements with the interlocking schema.

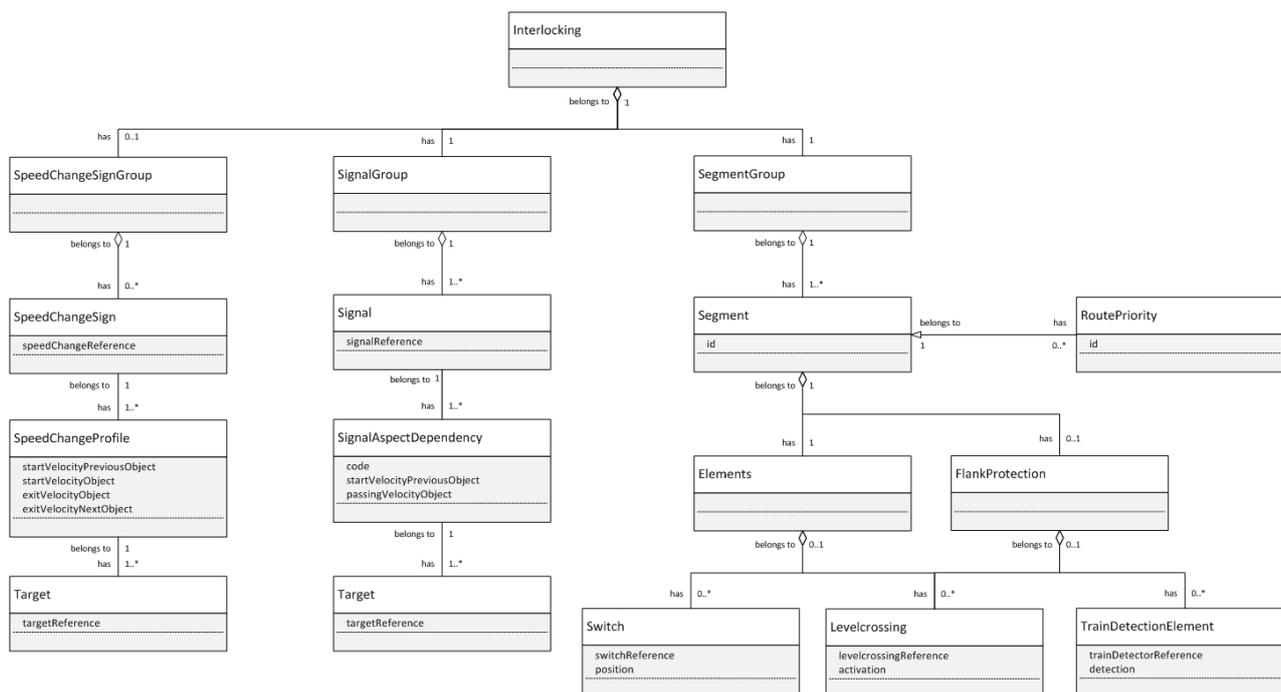


Figure 58: The final UML for the interlocking formalization in RailML. Notice that the UML corresponds to a mix of signal aspect classes of Figure 56 and mostly ixl element sub classes of Figure 50.

The final interlocking schema on the basis of Figure 58:

```

<interlocking>
  <signals>
    <signal>
      <signalAspectDependencies>
      </targetRef>
      </signalAspectDependencies>
    </signal>
  </signals>
  <speedChangeSigns>
    <speedChangeSign>
      <speedChangeProfile>
      </targetRef>
      </signalChangeProfile>
    </speedChangeSign>
  </speedChangeSigns>
  <segments>
    <segment>
      <elements>
        <signalRef/>
        <trackSection/>
        <switchRef/>
        <crossingRef/>
        <derailerRef/>
        <trainDetectorRef/>
        <levelCrossingRef/>
      </elements>
      <flankElements>
        <signalRef/>
        <trackSection/>
        <switchRef/>
        <crossingRef/>
        <derailerRef/>
        <trainDetectorRef/>
        <levelCrossingRef/>
      </flankElements>
      </routePriorities>
    </segment>
  </segments>
</interlocking>
    
```

The following discusses four remarkable elements. First, a signalAspectDependency links an aspect to an incoming speed profile with a start speed at the previous signal / speed sign and a target speed at the signal. Such a combination may lead to one or more target signals. In the case of a green aspect at signal 1404 (Figure 55), there is only one possible target:

```

<signalAspectDependency code="GR" sVpo="130" pVo="130">
  <targetRef refid="Sptn_1424"/>
</signalAspectDependency>
    
```

The codes of course depend on the specific signaling system in place. sVpo and pVo are abbreviations for “start velocity previous object” and “passing velocity object” respectively as stated in the UML of Figure 58.

Second, the RailML file only needs to incorporate a change of speed code at a speed sign when the change does not logically follow from the sign. For instance, a sign stating 130km/h logically implies that the ATB code for a train with a green aspect increases to 130km/h if the previous speed limit was lower. It also makes sense that a yellow aspect results in at least the dependency to stop at the next signal. The OS files in a.o. Figure 55 however clarify that a speed sign might also ‘randomly’ change the code when a signal provided a yellow or a green aspect combined with a speed indication. Experts determine these changes which do not follow from logic. These changes require a definition or else the dependencies will not match. Together with the input speed profiles defined at the subsequent speed sign or signal, an IT tool can reconstruct the full signal aspect dependencies as shown on an OS map. The next XML code shows how this formalization looks in the interlocking formalization:

```
<speedChangeSign refid="Sptn_1427V_300">
  <speedChangeProfile sVpo="40" pVo="0" eVo="4" eVno="80">
    <targetRef="Bv_529"/>
  </speedChangeProfile>
</speedChangeSign>
```

Where eVo stands for “exit velocity object” and eVno for “exit velocity next object”. Unfortunately, now a signal and the speed sign define identical exit and start speed profiles respectively since the speed signs’ exit speed profile does not follow from signal aspects. This characteristic therefore enters some redundancy in the formalization.

A third element to discuss comprises the description of (rail-)elements and / or flankProtection at the segments element. The definition segments follows the RailML group and is actually similar to the former routes element. Besides a reference of elements, the interlocking needs to know the required position of those elements. For a route from signal 1404 to 1426, the xml code becomes:

```
<elements>
  <signalRef/>
  <trackSection/>
  <switchRef>
    <switch refID="Sptn_1405" position="normal"/>
  </switchRef>
  <crossingRef>
    <levelcrossing refID="Hlm_137bV_Sptn_1405V_5.290"
      beam="down"/>
    <levelcrossing refID="Sptn_1405R_1427L_5.76"
      beam="down"/>
  </crossingRef>
  <derailerRef/>
  <trainDetectorRef>
    <trackCircuitBorder refid="Sptn_1405V_100" detection="false"/>
    <trackCircuitBorder refid="Sptn_1405V_300" detection="false"/>
    <trackCircuitBorder refid="Sptn_1405V_600" detection="false"/>
    <trackCircuitBorder refid="Sptn_1405R_200" detection="false"/>
    <trackCircuitBorder refid="Sptn_1405R_400" detection="false"/>
    <trackCircuitBorder refid="Sptn_1405R_600" detection="false"/>
  </trainDetectorRef>
  <levelCrossingRef/>
</elements>
```

The element positions in above case are indicated by the position of the switch (normal or bend/deviated), the beam position of the level crossing (up or down) and the detection of trains (true or false). The formalization could also indicate the switch position by the relevant connection (consider yellow circle in Figure 43).

The fourth element to discuss is the route priority. RailML can show a route priority by the sequence of switches a train needs to take. For example in the case of the same route from signal 1404 to signal 1426:

```
<routePriorities>
  <routePriority id="Sptn_1405R"/>
</routePriorities>
```

Figure 39 also shows that one switch (1405) will be crossed and it will be crossed over the right side (R) of the switch. In the case of an additional fictive switch, switch 88, which is crossed in the left direction, the XML would become:

```
<routePriorities>
  <routePriority id="Sptn_1405R_88L"/>
</routePriorities>
```

4.3 The Effects of the New Train Control System ETCS on the Interlocking Formalization

This section investigates the effects on the interlocking formalization with the replacement of ATB by ETCS. Subsection 4.3.1 provides a general introduction to ERTMS and its control system ETCS. Subsection 4.3.2 investigates how the interlocking structure alters. Subsection 4.3.3 indicates the changes to the interlocking formalization and subsection 4.3.4 elaborates on the changes to the engineering design chain.

4.3.1 ERTMS and ETCS

ERTMS contains three components: the European Train Control System (ETCS), GSM-R and the European Traffic Management Layer (ETML) (Goverde, 2012b; Winter, 2009a). ETCS includes both train protection as well as speed signaling. GSM-R communicates movement data to trains such as speed limits, signal position and track status. ETML aims to standardize traffic control databases and programs.

Especially ERTMS should standardize the protocols and procedures necessary to realize an interface, where on board train equipment communicates with trackside equipment without problems or adjustments necessary. Ideally, ETCS will replace EU railways’ current train control systems like AWS (United Kingdom), ATB (The Netherlands) and TVM (France). The benefits of such a standardized ETCS system arise from interoperable railway traffic. This means lower operational costs by achieving time, labor and rolling stock savings from easier border traffic (i.e. rolling stock requires fewer protection systems). In addition, revenues might increase due to a more competitive cross border connection compared to other modes of transport. Last but definitely not least, wide adoption and experiences with one standardized system will in time improve the safety case.

ETCS generally deals with three operational levels:

- Level 1:
 - similar to most ATP systems with movement authority through Eurobalise and train integrity and position by track circuit/track-free (Figure 59);
 - Level 1 with infill: same as regular but movement authority can now be received any time. Another ‘infill’ approach uses additional balises, although a train will most of the time still communicate discretely (Figure 60);

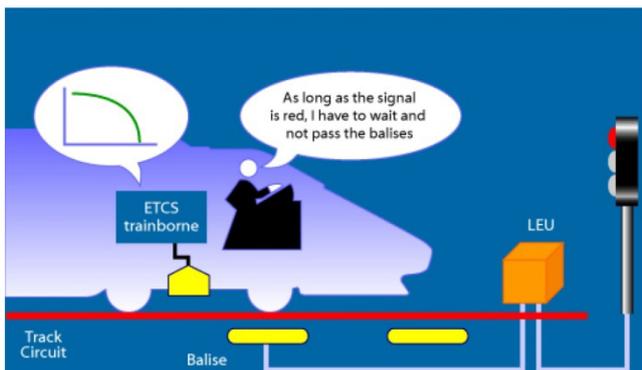


Figure 59: Regular process structure of ETCS level 1 (Goverde, 2012a).

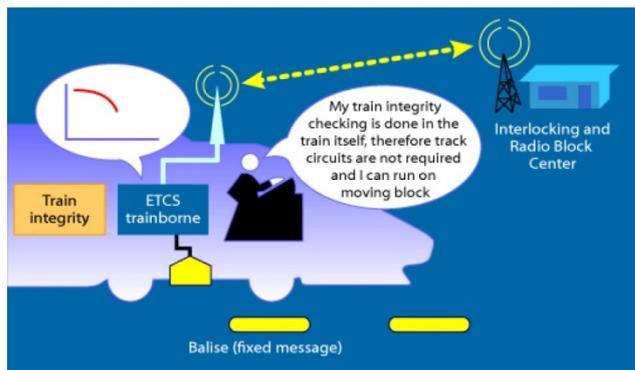


Figure 62: The ETCS level 3 process structure (Goverde, 2012a).

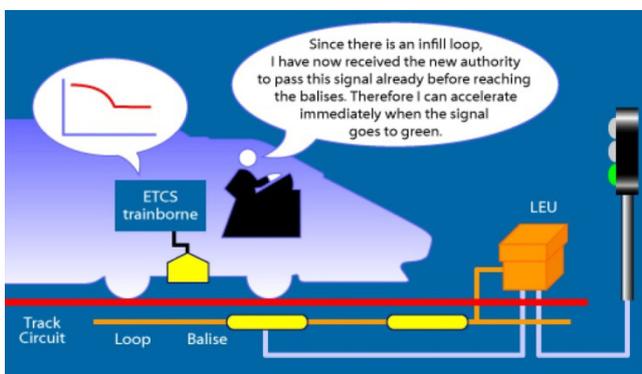


Figure 60: The infill process structure of ETCS level 1 (Goverde, 2012a).

- Level 2 (Figure 61): no lateral signals but cab based signaling. Movement authority goes by GSM-R, position calibration via Eurobalise and train integrity with track circuit / axle counters;

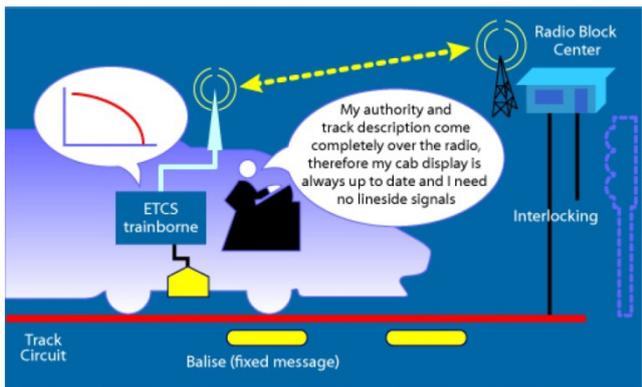


Figure 61: The ETCS level 2 process structure (Goverde, 2012a).

- Level 3 (Figure 62): equal to level 2 but train integrity is determined onboard and send via GSM-R. Therefore, this level allows for moving blocks instead of fixed blocks. No railway service operates on level 3 yet.

4.3.2 The ETCS Level 3 Interlocking Structure Compared to ATB-EG

Figure 63 compares the signaling system structure of ATB-EG and ETCS level 3. Prorail (currently) prefers to implement ETCS level 3 over the other possible ETCS levels (please refer chapter 2 for an explanation of the different ETCS levels). This study focuses on the Netherlands with consequent Dutch infrastructure manager Prorail and will therefore only consider the differences with ETCS level 3.

Figure 63 shows that the position of the interlocking changes fundamentally with the introduction of ETCS level 3. Whereas the interlocking takes the central position at ATB-EG, the RBC will take that role at ETCS level 3 (Theeg & Vlasenko, 2009b). The interlocking would currently read the status of the detection system, communicate with traffic control and neighboring interlocking systems, provide the ATP, ATB-EG, with speed restrictions and control signals and movable track elements. Although in reality ETCS level 3 interweaves the RBC and SIMIS W in reality, in theory SIMIS W only takes care of controlling movable track elements.

Besides a different positioning of the interlocking, the process structure alters fundamentally too (Theeg & Vlasenko, 2009a). Train detection with ETCS level 3 does not occur by reading trackside detectors anymore. In contrast, a train can determine its position by itself, mainly by using reference positions in balises, and send this to the RBC over the air by GSM-R. At the same time, GSM-R allows trains to receive movement authority by air instead of a communication via trackside signals. When a train violates the movement authority (in a negative way) by e.g. speeding, ETCS level 3 can apply the (emergency) brakes over the air as well. The main advantage of this approach is continuous train localization which makes block sections superfluous and increases track capacity. Communication with neighboring interlocking systems goes via RBC-to-RBC data transfer instead of mutual interlocking system communication. This results in one centralized communication flow between areas instead of one for every mutual (sub-)system. The relation of ETCS level 3 with traffic control becomes somewhat more opaque as the direction of the data flow is not always clear, e.g. a route request could directly go to an interlocking, via the RBC or directly to both.

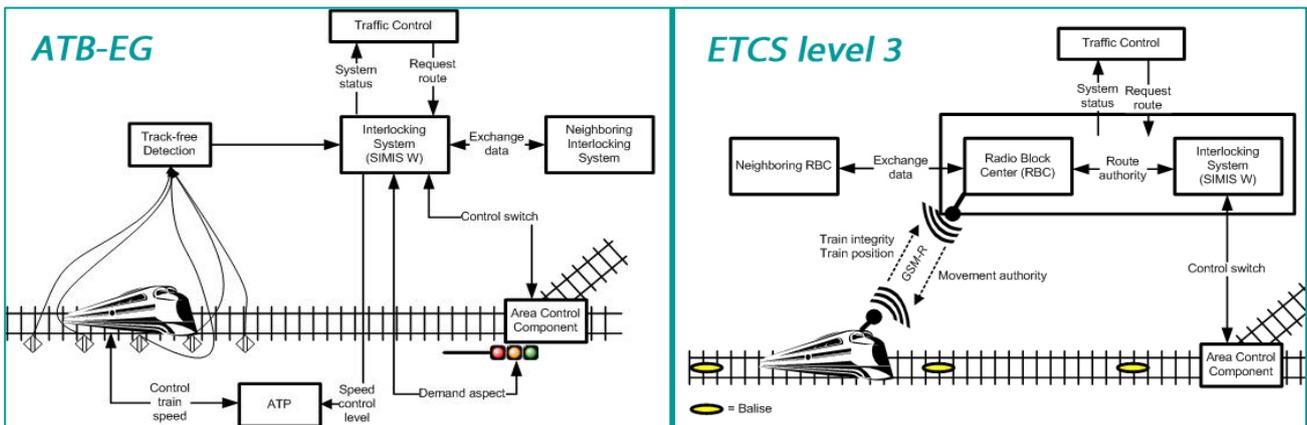


Figure 63: The left side of the picture shows the connections between the most trivial signaling systems for the purposes of SIMIS-W interlocking with ATB-EG (equals Figure 46). The right side of the picture does the same but for ETCS level 3. Each block represents a (sub-)system; each line represents the direction and type of data communication. The diamonds in the ATB-EG picture represent detection points in the form of track circuit sections or axle detectors. An area control component stands for a (group of) rail element(s).

4.3.3 The ETCS Level 3 Interlocking Formalization

The UML diagram in Figure 58 shows that the interlocking formalization comprises two main elements: signals and their aspect dependencies and segments and their associated rail elements. With respect to the signals and their signal aspect dependencies, one needs to question whether the relation between an aspect and a start-target signal still holds when implementing ETCS level 3. A fundamental difference between ETCS level 3 and ATB-EG comprises the loss of trackside signals and the introduction of onboard signaling. ETCS level 3 provides dynamic braking speed profiles to the driver which he or she needs to submit (Figure 64). The definition of signaling therefore changes as ETCS, in theory, does not provide a signal aspect with according (target) speed level for the coming block section anymore. In fact, ETCS level 3 could make signal aspects and block sections completely obsolete. For the ease of driving however, the train's onboard speed indicator might show a simple aspect system (green, orange and red) with associated speed level for a certain section of track. The latter dampens the fluctuating speed level effect that might arise in certain track topologies. For instance when a train starts on platform 3 at Spn and requests a route to Utg via switch 1427, then the speed indicator could indicate a maximum of and show a speed curve between 60km/h, then 40km/h from the curve to switch 1423, very briefly 100km/h between switch 1423 and 1425, followed by 40km/h between switches 1425 and 1427 and finally 100km/h again. One understands that this confuses the driver while a fixed speed level of 40km/h for this section of track would not significantly reduce travel time or capacity. The train's onboard interface should therefore translate the ETCS data into a driver friendly representation. The source ETCS data however still contains all relevant parameters to determine detailed braking profiles.

Figure 64 clearly shows that ETCS level 3 does not make use of signals anymore and allows for a more optimal speed curve (compare the two areas below the red line: ETCS' is considerably larger). One might find it remarkable that the ATB curve indicates 120km/h instead of the allowed 100km/h. ATB however does not have a code that enforces 100km/h; the 100km/h comes from the infra-

structure and signal speed indicators. The speed levels in ETCS match those in ATB although one could also desire to change them since ETCS does not limit itself to the available ATB speed codes.

As a result of the above, the interlocking formalization does not require the 'signal' sub element of signals anymore with ETCS level 3 (subsection 4.2.3).

Onboard train integrity and positioning provide accurate real-time train speed and location to the RBC. The RBC accordingly calculates when the interlocking must interlock the movable track elements in order to prevent a suboptimal speed profile. Then, the RBC can also determine the dynamic route activation point. ETCS makes the second sub element of 'signal', signal aspect dependencies, superfluous too due to the absence of trackside signals and because speed does not depend on static (signal) aspect combinations anymore.

The inclusion of the 'segment' element depends strongly on how the RBC commands the interlocking. The RBC could either request a route or a movable track element when in range. Currently, ETCS level 2 at the HSL-south applies the first approach by using virtual signals and changes the interlocking formalization to:

```

<interlocking>
  <segments>
    <segment>
      <elements>
        <signalRef/>
        <trackSection/>
        <switchRef/>
        <crossingRef/>
        <derailerRef/>
        <trainDetectorRef/>
        <levelCrossingRef/>
      </elements>
      <flankElements>
        <signalRef/>
        <trackSection/>
        <switchRef/>
        <crossingRef/>
        <derailerRef/>
        <trainDetectorRef/>
        <levelCrossingRef/>
      </flankElements>
      <routePriorities>
    </segment>
  </segments>
</interlocking>
    
```

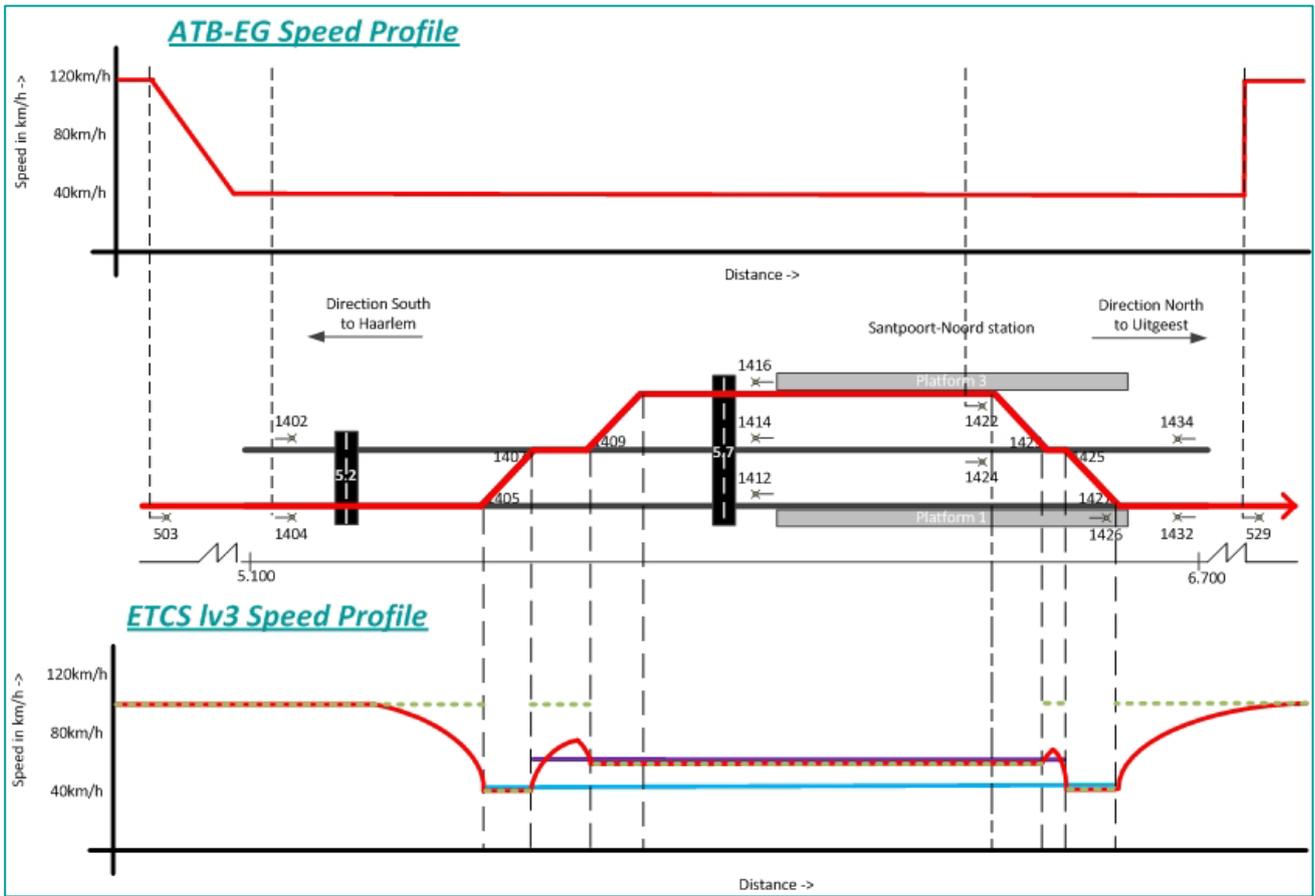


Figure 64: The differences between speed control with ATB and ETCS level 3 when driving the red route. The figure is not on scale. The red line in the speed profile diagrams indicate the maximum allowed speed (called EBIC, Emergency Braking Indication Curve, at ETCS). The black dotted lines represent signals or other rail elements like switches that influence the maximum allowed speed. The green dotted lines in the ETCS diagram represent the maximum speed in a section of track. The figure shows two short peaks that may confuse the driver because of quickly changing speed levels. The onboard speed indicator could use a speed cap to solve this issue. For instance, when driving from switch 1405 to 1427 via platform 3, the blue line could be used as a speed cap of 40km/h. Alternatively, every train going from switch 1407 to 1423 via platform 3 could get a cap of 60km/h shown with the purple line.

The above interlocking formalization defines a segment from virtual signal to virtual signal. This route approach bases on the current practice of ETCS level 2. One big advantage of ETCS level 3 however, arises from the possibility to operate with so-called moving blocks. In other words, movement authority does not depend on fixed blocks / segments but on the preceding train's position. As such, the interlocking formalization could take three different shapes:

1. segments of movable track elements for station areas only (and consequently no moving block operation in station areas);
2. a large amount of very short segments of movable track elements;
3. no definition of segments and direct request of elements from the RBC.

The first approach may follow the same formalization as for ETCS level 2. The second approach, assuming that each segment only contains one movable track element for optimal track allocation, makes segments out of the elements. Furthermore, route priorities do not play a role anymore since the segments are that small, that the interlocking never requires to decide on a certain route. In

addition, the interlocking formalization does not need to include signals and detection points since ETCS level 3 does not contain any. The interlocking formalization of the second approach in XML structure:

```

<interlocking>
  <segments>
    <trackSection>
      </flankElements>
      <signalRef/>
      <trackSection/>
      <switchRef/>
      <crossingRef/>
      <derailerRef/>
      <trainDetectorRef/>
      <levelCrossingRef/>
    </flankElements>
  </trackSection>
  <switchRef>
    </flankElements>
  </switchRef>
  <crossingRef>
    </flankElements>
  </crossingRef>
  <derailerRef/>
  <levelCrossingRef/>
</segments>
</interlocking>
    
```

In the case of the third approach, the RBC directly requests movable track elements, the interlocking gets more of a role like the detection system has with ATB. The only difference is that the interlocking also needs to take action instead of only providing data. The RailML infrastructure schema provides RBC development with sufficient data to locate the relevant movable track elements in an interlocking area. Then, it seems redundant to use a separate formalization for the purposes of interlocking.

4.3.4 The Main Changes to the Engineering Design Chain with ETCS Level 3

Changes to the engineering design chain compared to the ATB-EG signaling structure arise when the new interlocking structure requires additional or less data. In the case of the first approach with virtual signals, the interlocking formalization would not need data on signal aspect dependencies anymore. The OS currently represents signal aspect dependencies and would therefore lose its purpose, at least for interlocking engineering. RailML mitigated the importance of TOS engineering at Siemens, though it makes TOS engineering completely superfluous without the need for OS files. Besides a reduction in data and processes, ETCS development does need data about virtual signals. Therefore the development of OS maps could perhaps change to focus on those virtual signals instead of eliminating it from the chain.

The second approach with a large amount of small segments results in the same changes as the first approach. One difference is the exclusion of route priorities. Therefore, the SVA does not need to include that topic anymore. The third approach in which the RBC directly controls the interlocking system's actions, would change a lot. Interlocking engineering would now limit itself to connecting the movable track elements and reading/communicating their status: the operational layer in Figure 5. The functional layer shifts almost completely to the RBC. As a result, an OBE would probably suffice as input data. Furthermore, the interlocking engineering processes at Siemens probably reduce significantly in time and can maybe merge into just one process. As a consequence of this approach, RBC development would need to include the mitigated interlocking processes to sufficiently control the interlocking. In addition, the interlocking engineers need to know how the RBC desires to communicate with the interlocking, e.g. how will it call the movable track elements, to exclude errors.

5. RailML's Improvement of the Interlocking Engineering Design Process

The interlocking formalization theoretically allows partners in the engineering design chain to exchange most data in a machine readable format for interlocking development. RailML as a whole can however not exchange all data and information that partners need during the interlocking design and engineering. In order to establish a judgment on the improved (lean) process performance, one needs to know exactly which of today's processes RailML can change or eliminate. Section 5.1 describes where RailML supports the engineering design process and qualitatively concludes the 'waste' it mitigates. Section 5.2 elaborates on the developed simulation model to quantitatively measure the performance of RailML in section 5.3. Section 5.4 discusses those results.

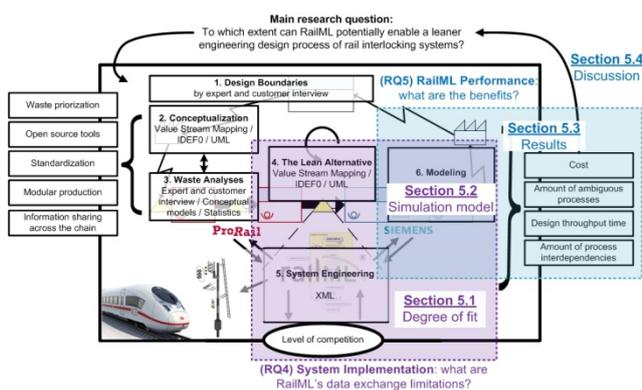


Figure 65: The purple (research question 4) and blue (research question 5) boxes visualize the research topics covered in this chapter.

5.1 RailML's Degree of Fit in the Interlocking Engineering Design Chain

On the basis of the Sptn RailML with interlocking, this chapter assesses the extent to which RailML can function as a data exchange tool. Figure 8 illustrates the web of IT tools and applications that RailML needs to connect. The RailML web contains four layers: the interlocking design process, the interlocking engineering process, applications and the (GIS) database. The next subsections discuss each layer's connection to the RailML formalization and general shortcomings in RailML v2.2 for the purposes of interlocking engineering design.

5.1.1 General Shortcomings in the RailML v2.2 Schema for the Purposes of Interlocking

Subsection 4.2.2 defined the elements that a RailML interlocking schema should encompass. The interlocking engineering process also requires additional data that belongs to the infrastructure schema. The v2.2 infrastructure

schema does lack some elements to fully cover the interlocking engineering process and result in an EVP. The next main elements currently miss in RailML v2.2 for the purposes of interlocking:

- Level crossing characteristics. Currently, RailML only allows for localization of a level crossing. The interlocking however also needs to know which track detectors will activate it. Furthermore, the Dutch railway network also knows a so-called 'stop/door' characteristic. The "stop/door" characteristic ensures that when a level crossing is directly positioned after a station, the level crossing will close in time for a through train and close later in the case of a train that stops at the particular station. Elements and attributes to describe the "stop/door" characteristic miss. Finally, de-activation of a level crossing requires a formalization in RailML too, e.g. by means of a timer duration.
- Signal types differ per railway organization. Chapter 4 explained why the possible aspects that a signal may show prove important for an interlocking. Therefore, RailML needs to know the locally used signal types and aspect definitions. In order to keep a RailML file concise and standardized, it might be wise to use local specific XML files as input to the RailML for this purpose.
- RailML currently does not distinguish and describe relations between station areas, junctions and open track. An interlocking needs to communicate with neighboring interlocking areas or open track to announce incoming trains and consequent track actions.
- RailML does not include logic to ensure failsafe operation. In relation to the interlocking formalization in subsection 4.2.3 for example, when a signal malfunctions, the preceding signal should directly go to stop. RailML however has no means to indicate that a signal malfunctions.
- Detection points, especially in the case of track circuit sections, divide a track into sections that can have relations on their own. Modern interlocking systems monitor the sequence and correct clearance of those track sections. On the one hand to improve train detection and on the other hand to easily allow for splitting, merging or reversal of trains at stations (also called a TVD section in an EVP). In the latter case, an interlocking would e.g. demand an attribute to describe an active or passive TVD.
- RailML does not contain attributes to describe the redundant system setup of certain elements. Examples

could be level crossing activation detectors and axle counter detectors.

- RailML does not contain the ability to describe whether an interlocking may control a switch on its own instead of for setting a route. This situation prevails with shunting activities.

5.1.2 The Use of RailML in the Interlocking Design Process

Recall from Figure 19 that the interlocking design process encompasses six main sequential steps in which data transforms. In order to understand whether RailML fits, the current input-output (data) products need to complement with the possible entries in RailML. This also identifies the potential lean effects of RailML that reduce 'waste' by making certain processes redundant or by altering the performance.

Appendix K, RailML's coverage of interlocking data products, shows that especially the RVTO, SWOD and test processes exchange and transform a lot of data. Consequently, RailML has most lean potential during those processes. The products in the CRS and FIS contain quite some written project guidelines not suitable for RailML. In general, RailML has the potential to:

- provide the CRS group with current track topology data having as much detail as the OBE, OS and OR maps;
- provide the FIS party with current track topology data having as much detail as the OBE, OS and OR maps;
- allow the FIS party to improve the development of topology alternatives at shorter throughput times by using machine readable track topology data with route analysis tools;
- transfer track topology data in a machine readable format between the FIS and RVTO with multiple parties;
- when not assuming a new rail corridor, base VT-OBE and OS map development directly on current track topology data, axle load characteristics and some route calculation data;
- transfer VT-OBE and OS map data in a machine readable format between the RVTO and SWOD when executed by different parties;
- base OBE map development directly on VT-OBE data, axle load characteristics and some route calculation data;
- base an OR map on the (VT-)OBE data by the mean of an IT tool;
- transfer OBE and OS data in a machine readable format between the SWOD, test protocol development and interlocking engineer;
- validate RVTO and SWOD results on the basis of comparison when having the right tools,
- develop larger part of test protocol via an IT tool from OBE, OS and OR;
- provide complete input for the dry test.

The new data exchange possibilities allow processes to get leaner by tackling various sorts of 'waste' as defined in Table 13. Appendix K and Table 14 indicate that RailML can transform the interlocking design chain in the direction of the lean state described in section 3.4. The FIS' main task comprehends the development and selection of various alternatives that suit CRS' requirements. Exchanging track topology data to FIS and CRS enables easy modifications to develop and compare future topology strategies. As a result, the stakeholder analysis takes most effort during both stages. Since stakeholder preferences may change after the development of concept topology strategies, it makes sense to combine both stakeholder management (CRS) and solution development (FIS) in the case that alternative development is a less intense process.

The RVTO aims to detail FIS's choice of topology with the required data for train operation like signal aspect dependencies and the rail elements in the interlocking area. RailML shortens this process substantially by providing former OBE and OS maps, if existent. The right IT tool eliminates OBE and OS development from scratch and makes sure that both are aligned. In that case, most development time remains to develop the testing and planning files and update the GIS and traction data. The SWOD focuses on RVTO's alignment with track circuits and return maps. RailML does not include any OR and traction related data due to which this process' focus remains the same. RailML could therefore potentially proof a catalyst to merge the RVTO part development of OBE and OS with SWOD; leave RVTO as the preparation of data that concerns mainly traction, GIS and current OBE/OS/OR maps. The main advantage results from a process in which a former RVTO OBE or OS development does not require changes (read: rework) on the basis of e.g. track circuit related issues during SWOD.

The test protocol development mostly comes down on interpreting OS, OBE and OR maps with some additional rules from the SVA. A link between RailML and BITS could almost completely develop a test protocol with the exceptions mentioned in the SVA. Therefore, test protocol development could form one process with the dry test to enlarge flexibility.

Although RailML could provide a catalyst to make the interlocking design leaner, it also leads to mainly four disadvantages. First, RailML at least adds another IT tool in the process and probably many others to ensure conversion. Tools require maintenance while not directly adding value to the data product: not lean. Second, and partly as a result from the first, every conversion may lead to an error. Every once in a while, a computer can make calculation, read and write mistakes too; even with the perfect tool. A designer has more troubles to discover errors in a data file than on a map because programming language makes face validation hard (Palacios, 2013). Third, RailML does not solve various type 1 'waste' factors. The most stringent factors include ambiguity caused by using the wrong data source and far-going planning improvements during FIS and CRS. Fourth, RailML cannot include all data products (consider Appendix K's red data labels).

When thinking outside the "RailML-box", one could argue why not to focus on minimizing the amount of tools and

databases and directly connect the remaining. When RailML can depict a substantial amount of data in the interlocking data process, another architecture may as well. In fact, Prorail already plays with a new database idea based on InfraAtlas: INCA (Middelkoop, 2013). INCA would use the same structure as InfraAtlas which multiple parties in the chain already extract data from. INCA aims to provide each party in the chain with the necessary data and feeds it back when data transfers to the next party. In this way, INCA also records a log file and makes it more of a project management database. INCA does however not become a threat for RailML development since it can only be used in a local interlocking design process like Prorail's. In order to share the required data between local infrastructure managers, global industrial parties like Siemens and global analysis programs like Opentrack, RailML still has most potential to exchange data.

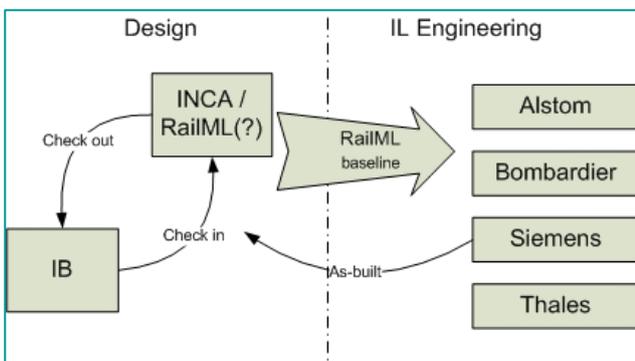


Figure 66: A possible system architecture of RailML and INCA in the engineering design chain of interlocking systems.

5.1.3 The Use of RailML in the Interlocking Engineering Process

The previous subsection indicated that especially the data transfer from infrastructure manager to industry benefits from RailML. Question arises: does RailML align with the standards and tools used at industry for interlocking engineering? This subsection investigates Siemens' specific requirements for the data products from Prorail and whether RailML could exchange that data. Furthermore, this subsection looks into the role of RailML during the final test processes.

The engineering of a SIMIS-C electronic interlocking system provides insight in the data requirements from the infrastructure manager. SIMIS-C is the older version of SIMIS-W which Siemens installs these days. SIMIS-W does not have such an elaborated documentation yet while the engineering process of both systems is almost identical. Siemens starts with the preparation of the data that it gets from the infrastructure manager. This data includes the OS and OBE. Data preparation mainly comes down to two things: one, validate the data of the OS and two, since the OS and OBE come as a static map, insert all elements depicted on an OS and OBE into a database.

The engineering of an ElementVerbindungsPlan (EVP) forms the heart of SIMIS-C's development process (Kanoun, 2011). The EVP describes all rail elements with their characteristics and relations at an interlocking area. As such, every part of an interlocking architecture basis

itself on this EVP. The input for EVP engineering contains OBE, OS, SVA and the Technical Overview map of Signals ("Technisch Overzichtsblad Seinen" or TOS). The TOS uses the OS and OBE to describe the rail elements and speed profiles, a combination of a speed level at the begin signal

Table 14: Shows which of the 'waste', discovered in section 3.3, RailML can tackle during the interlocking design process. The table categorizes the 'waste' by the three levels of priority.

First priority of 'waste'	
<p>Eliminate ambiguity in RVTO and SWOD</p>	<ul style="list-style-type: none"> No rework due to data misinterpretations from non machine readable data. No rework due to inconsistent data between parties that operate in an isolated data environment. No rework due to use of outdated data thanks to parties that were unable to share (in time).
<p>Reduce the number of data translations, interpretation mistakes and rework due to missing data</p>	<ul style="list-style-type: none"> The introduction of a machine readable format that all interlocking design parties use, simplifies design processes while increasing quality. Therefore, the separation between CRS&FIS and mainly RVTO&SWOD becomes less clear; which reduces the need for multiple engineering agencies.
Second priority of 'waste'	
<p>Improve planning of CRS and FIS</p>	<ul style="list-style-type: none"> Reduce time variability by eliminating the need to develop various track topologies by hand.
<p>Merge or eliminate processes to reduce complexity; especially within RVTO, SWOD and test processes</p>	<ul style="list-style-type: none"> No need to interpret non machine readable maps. Only new interlocking areas still require OBE / OS development from scratch. Tier rejection rates will barely exist anymore since all parties deliver in the same format and they can quickly validate their work at all times due to the machine readable format. Developing a test protocol for BITS will almost completely go automated; reducing the need for a separate process.
<p>Reduce amount of different (IT) protocols and non machine readable data transfers</p>	<ul style="list-style-type: none"> In theory, RailML eliminates all non machine readable data transfers between green and orange labeled products in Appendix K.
Third priority of 'waste'	
<p>Increase productivity of RVTO and SWOD by reducing complexity</p>	<ul style="list-style-type: none"> The execution of a substantial amount of design rules will shift from humans to RailML or corresponding IT tools. This table clarifies that RailML eliminates and merges processes; reduces the amount of interactions and thus complexity.

and a speed level at the end signal, that define signal aspect dependencies and aspect characteristics (Figure 67) (Kanoun, 2003). In other words, a TOS depicts the possible speed profiles between rail elements in an interlocking area, not just signals but also speed signs, switches and special occasions like buffer stops, and the possible speed profiles before and after a pair of such rail elements. A TOS models the effect of switches and special occasions on speed profiles as fictive signals. A TOS eventually aims to provide signals with the right aspects, a train with the right speed data and an ATP with control data to take corrective action if necessary.

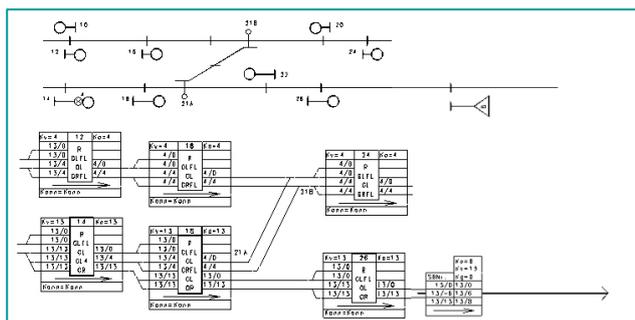


Figure 67: A TOS of a fictive Dutch track topology. The TOS shows the speed profiles depending on route and signals' aspects. The speed sign actually is modeled as a signal without aspects.

The EVP process starts with the development of a node-branch model that connects all relevant rail elements presented in an OBE. In addition, ATB data from the OS and speed profiles from the TOS enrich this first model. Second, the EVP engineer will provide the elements with characteristics like activation points, times, element relations and route relations. The characterization of elements occurs on the basis of the OBE, OS, TOS and SVA. Third, the EVP engineer will divide the elements for processing in the central interlocking unit or local interlocking units. Fourth, every interlocking process unit classifies each of its own elements with a dedicated hexadecimal element number for relative identification. This concludes the EVP process and results in a map like the one in Figure 68.

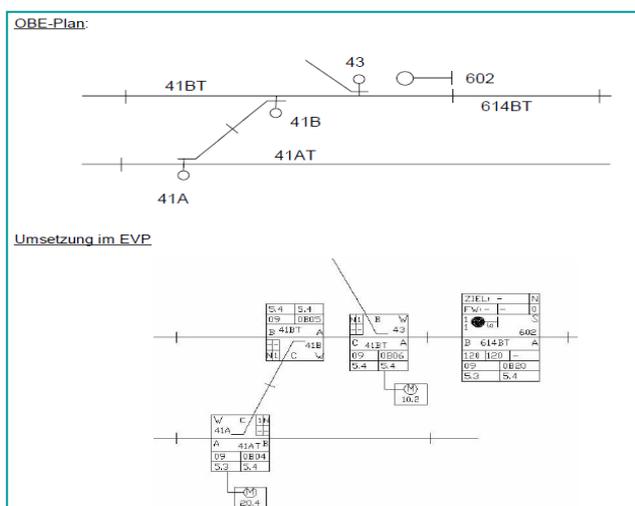


Figure 68: An OBE turned into an EVP by connecting various rail elements that are required for an interlocking system.

Chapter 3 explained that after EVP engineering, Siemens starts an EVP conversion process. Here, the EVP is loaded into test hardware to eventually, in the fourth engineering process, test whether the EVP contains all data to produce an interlocking system. The EVP conversion elaborates on four sub processes. First, Siemens can engineer a major part of the interlocking IT architecture on the basis of element division over central and local interlocking process units. Second, the EVP conversion process develops visualization software that shows the operational status of the elements in the interlocking area mainly for the purposes of traffic control and maintenance. Third, Siemens projects the data of an EVP into hardware units/channels on the basis of a hardware relation analysis that basis itself on the EVP relations. Last, the test hardware towers are allocated, the channels installed and connected by wires. The EVP test checks whether the interlocking test setup derived from the conversion process shows the behavior as expected. A tester could for example manipulate a signal and see whether the related elements and signals change according to the OBE, OS and SVA. When a test proves engineering mistakes, the TOS and/or EVP require adjustments. In an extreme case even the OBE, OS and/or SVA require changes. The result of the EVP test contains an improved interlocking design and a report that goes to Prorail. As explained in subsection 5.1.2, a dry test at Prorail compares their simulation results with the report of Siemens. In the case that no significant issues arise, the interlocking production may start.

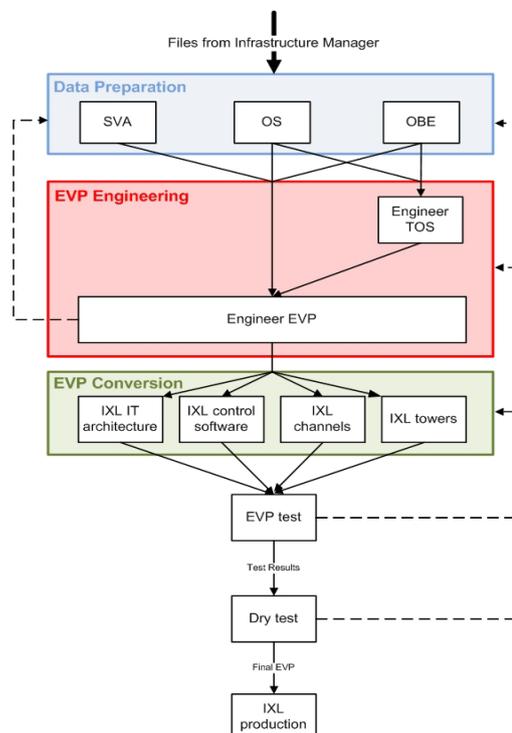


Figure 69: The chain of events during the interlocking engineering process with corresponding input and output data products.

Appendix K, RailML's coverage of interlocking data products, indicates that RailML can somewhat transform the interlocking engineering chain towards the lean state described in chapter 3. RailML features most potential to support the data preparation process. Currently, Siemens

engineers need to interpret OBE and OS, extract the right data and put it into a database. RailML offers the OBE and OS data in a machine readable format as stated in subsection 5.1.2. Modelers in the EVP engineering process can thus derive the relevant data from the RailML with the push of a button. Furthermore, the OS validation sub process seems superfluous with the introduction of RailML. An additional IT tool could easily check whether the OS and OBE correspond to each other and the set requirements. Probably, a party in the interlocking design process will already take care of such an OS validation, due to which Siemens does not have a need to validate it anymore. To summarize: there does not seem a reason to keep the data preparation process after introduction of RailML. RailML will make the EVP engineering process leaner by shortening engineering time, reducing the probability of errors and reducing the need for rework. The signal aspect dependencies in the RailML interlocking formalization seem to correspond with the modeling approach in the TOS. The TOS engineers only need to add virtual signals and static speed signs to make a TOS complete. A large extent of the EVP results from the OBE, OS and TOS which can, with the right tools, be inserted automatically. EVP engineering still needs to take care of adjustments imposed by the SVA. Especially in the case of completely new interlocking areas. In addition, an EVP engineer probably still needs to call on the division of elements per central and local process units.

The EVP conversion will not directly change due to the introduction of RailML. Although RailML could maybe support the EVP format, question arises what the practical need would be. Siemens probably already has tools that align the EVP with IXL tower, channel, software and IT architecture engineering. Therefore, RailML would only result in an additional conversion process; increasing the probability of system errors.

RailML delivers the same contribution to the EVP test as it does to the EVP conversion. RailML could maybe exchange the EVP input and updated EVP output between the EVP engineering and dry test process but there does not seem an added value of that action.

On the other hand, RailML could in the long term perhaps lead to the elimination of the EVP conversion and EVP test process. The need for a prototype testing process originates from a substantial amount of non machine readable data transfers and the varying signaling system environments. RailML standardizes the way to describe signaling systems, reduces the amount of data transfers and automates those data transfers. With ETCS in the distant future the signaling system would not differ across many countries and might also make interlocking engineering less complex. Therefore, with sufficient experience from best practices, standard test deficiencies can automatically be incorporated into EVP engineering. As a result, a large part of the EVP conversion and EVP test becomes obsolete.

The dry test process does benefit from the introduction of RailML. Siemens could deliver a RailML interlocking file on the basis of the final EVP. Prorail can compare this interlocking file with previous versions to study whether the changes meet the requirements. Furthermore, Prorail could develop an interlocking RailML on the basis of an OS, OBE and SVA themselves for verification purposes.

These RailML files improve the reliability of the test and could maybe eliminate part of the behavioral comparison by means of BITS simulation.

Table 15: Shows which of the 'waste', discovered in section 3.3, RailML can tackle during the interlocking engineering process. The table categorizes the 'waste' by the three levels of priority.

First priority of 'waste'	
Eliminate Siemens' data preparation and EVP conversion processes	<ul style="list-style-type: none"> RailML enables machine readable transfer of OBE and OS. This allows for easy verification and validation during the design process. Therefore, there does not seem a reason why the data preparation process would still exist after implementation of RailML.
Make interlocking tests (more) dependent on project size	<ul style="list-style-type: none"> With the use of an interlocking RailML file from both Siemens (on the basis of the EVP) and Prorail (on the basis of merely OS, OBE and SVA), the dry test process could be substantially shortened and have smaller fixed testing time component that is independent of the project size. The contribution of dry test preparation from Siemens is however slim compared to their EVP test.
Second priority of 'waste'	
Minimize overhead costs of Siemens' activities since they form a fixed cost share independent of project type or corridor	<ul style="list-style-type: none"> Overhead costs will reduce due to the probable elimination of the data preparation process. Overhead costs will reduce due to the faster, more automated engineering of the EVP.
Reduce amount of different (IT) protocols and non machine readable data transfers	<ul style="list-style-type: none"> In theory, RailML eliminates all non machine readable data transfers between green and orange labeled products in Appendix K.
Third priority of 'waste'	
Increases productivity and utilization rates of the testing processes	<ul style="list-style-type: none"> Productivity of the dry test increases due to support of RailML with the interlocking formalization.
Increase the average productivity of data preparation at Siemens	<ul style="list-style-type: none"> This will not be of concern anymore when the data preparation process would be eliminated.

Table 13 and Table 15 indicate that RailML can tackle all of the type 1 'waste' factors related to interlocking engineering. Although the effect on a more variable cost structure of testing is slim. RailML will also tackle both of the type 2 'waste' factors related to interlocking engineering. RailML can however not address all type 3 'waste' factors related to interlocking engineering. The importance of hardware does not decrease due to an unaltered EVP conversion and EVP test process. Consequently, project management costs at those processes will not decline either. Furthermore, the productivity of the testing process only improves somewhat due to an improved dry test.

5.1.4 The Added Value of RailML for the Purposes of Other Applications

Besides the engineering design chain, a RailML interlocking formalization could also have benefits for applications that use RailML. This section briefly touches the various options in the context of interlocking.

Radtke (2008) describes eight modeling areas that use the railway infrastructure as a basis:

“(1) running time calculation, (2) calculation of headways, (3) depiction of the signaling system, (4) block occupation for conflict detection and timetable construction, (5) searching for train routes in networks, (6) capacity calculation, (7) planning of possession and track work and (8) railway operational simulation”. In order to assess which of these modeling areas might benefit from the RailML interlocking formalization, Radtke (2008) defines three levels of infrastructure detail: the macroscopic, mesoscopic and microscopic level. Middelkoop (2013) adds a fourth level: the nanoscopic level. Normally, a microscopic model contains track topology with detailed track data like gradients and speed levels, signaling system data e.g. the position of signals and some operational data that relates to a timetable for example (Radtke, 2008). Interlocking would a.o. add signal aspect dependencies, aspect interdependencies with rail elements and the route life cycle process. Therefore, Middelkoop argues that interlocking logic adds such a substantial amount of extra detail to an average microscopic model, that the name would not justify it anymore. Middelkoop states that a four level model changes the spectrum of model applications (Figure 70) and seriously questions whether an interlocking formalization may benefit the results of current applications. He expresses his point by an argument based on a pilot project and an argument based on current calculation methods.

Middelkoop (2013) and Schut and Dragt (2013) explain that Prorail had a RailML pilot in the past to investigate the opportunities of RailML. The pilot encompassed the development of a tool to convert data from InfaAtlas to RailML v1.0. Prorail merely developed the tool for the purposes of track analysis studies. Since a RailML v1.0 to OpenTrack, a commonly used railway micro simulation program, tool existed, easy conversion from InfaAtlas to OpenTrack was achieved. RailML v1.0 did not deliver improved analysis results over existing tools that Prorail used. In fact, RailML v1.0 proved to lack a critical amount of characteristics for simulation purposes. Furthermore, Middelkoop argues that RailML requires an additional conversion over a direct conversion from InfaAtlas to OpenTrack. An additional conversion, as explained earlier, may lead to errors and requires people to maintain the tool.

Besides discouraging experiences from a pilot project, Middelkoop (2013) questions whether a detailed interlocking formalization will deliver improved analysis results. One could argue that a more realistic description of the interlocking in a model could for instance lead to improved analysis results regarding block occupation and running times. Radtke (2008) explains that a way to mimic an interlocking in a model is by representing each node, or switch, in a network as a single server queue according queuing theory. Wendler (2008) elaborates that queuing

theory’s estimations are fairly accurate; especially in the case of regular interval timetables because it allows for an accurate formulation of the arrival distribution. As a result, a detailed formalization of the interlocking would probably only slightly improve analysis results. This however comes at the cost of longer analysis specification and, in the case of simulation, calculation times.

One application might however benefit for sure from the interlocking formalization according to Middelkoop (2013): the signaling simulation software BITS. As explained earlier, BITS emulates the behavior of the interlocking area. RailML input reduces the specification time of a BITS model and increases the reliability.

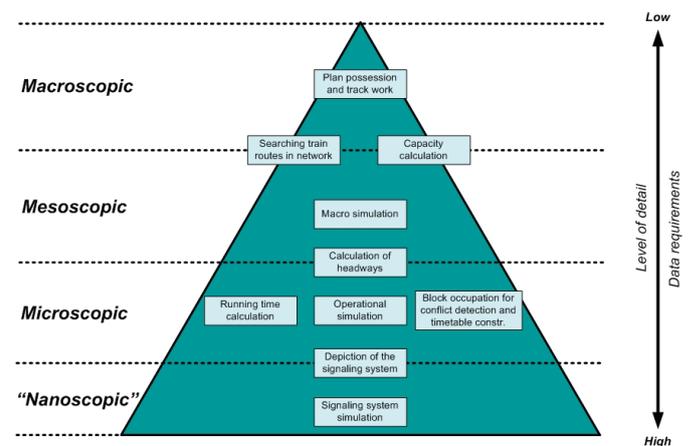


Figure 70: The proposed four levels of modeling detail with corresponding railway modeling areas that use infrastructure data (Middelkoop, 2013; Radtke, 2008).

5.1.5 A Single Data Source Approach with a Geographic Information System

Even though RailML enables a link between various data types, a big challenge arises in data version management. After all, which source contains the most recent data and who manages this? In order for RailML to support each process in the engineering design chain, the input and output destination of data needs to be known; otherwise most of RailML’s benefits mentioned in the previous subsections will not hold.

Prorail intends to solve this issue by means of a Geographic Information System (GIS) (Dragt, 2013). A GIS stores about any data and information that relates to a geographical position (ESRI, 2013). A GIS is not just a database, it can also be a combination of a database with connected hardware and software. As such, a GIS allows for various functions that range from measuring data to data analyses and from system visualization to operational control. ESRI (2013) state five main benefits of using a GIS approach: “(1) cost savings, (2) better decision making, (3) improved communication, (4) better recordkeeping and (5) managing geographically”. In the context of the interlocking engineering design chain and RailML, a GIS allows to store the infrastructure and interlocking elements tied to coordinates. In addition, a GIS does contain version management. Furthermore, a GIS can easily visualize data on maps and even visualize the result of analyses for validation purposes. Palacios (2013) stresses the importance of

such visualization opportunities in order to prevent errors and inconsistencies in data processing.

The main bottleneck in GIS development comes from the geographical position measurement. Every object needs a geographical reference and that requires an enormous measurement job. Furthermore, especially in the case of the interlocking engineering design, accuracy of the coordinates is of vital importance to the safety of the final system. A measurement error of a few meters could for example provide a train with a too short overlap track length and lead to a collision in the case the train overshoots its signal.

In addition, the role of InfraAtlas with a GIS becomes questionable as today, InfraAtlas stores the latest data for the purposes of mainly calculation programs/models like FRISO or ROBERTO. InfraAtlas does not provide the engineering design chain with data. Due to this different nature and architecture of the EAI's, it becomes a challenge to run both alongside each other or to change the InfraAtlas architecture completely to GIS.

A detailed elaboration on the architecture of a GIS in the engineering design process is outside the scope of this research but Figure 71 closes with a possible visualization of the RailML constellation with a GIS.

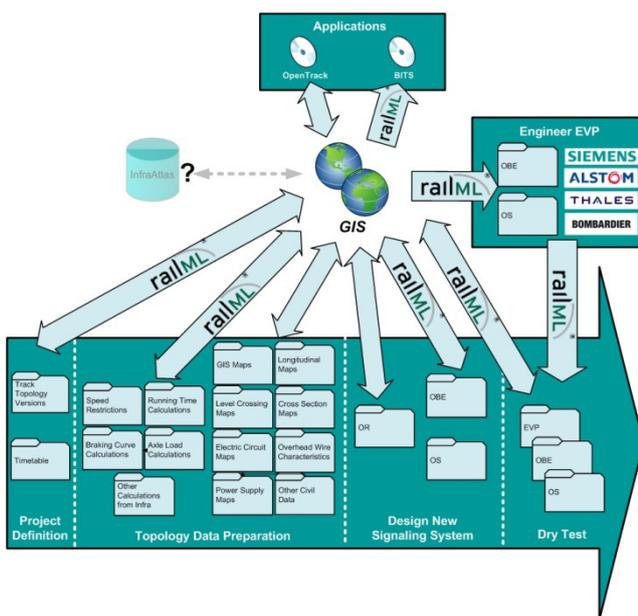


Figure 71: The constellation of RailML, GIS and the most relevant data products and applications divided over the potential lean interlocking engineering design chain (section 3.4).

5.2 The RailML Performance Model

This section describes the main concepts used to develop models that assess the lean performance of RailML in section 5.3. Subsection 5.2.1 first discusses the modeling approach. Subsection 5.2.2 continues with a description of the status quo simulation model. Subsection 5.2.3 discusses the changes made to derive the status quo model with support of RailML. Subsection 5.2.4 ends with a description of the lean simulation model.

5.2.1 Justification for the Simulation Approach

Verbraeck and Valentin (2006) define simulation as “*replicating a system found in reality over time in order to understand its behavior, mitigate undesirable situations or investigate the effects of new policies*”. Simulation constitutes the opposite of analytical modeling. The general downsides of a classic analytical approach comprise (1) the limited applicability to derive exact solutions and (2) the time-intensive, opaque and (near) impracticable nature when dealing with very complex issues. A complex issue usually deals with parallel processes, process interdependencies, time relevancy, process variations and parameter uncertainty. Furthermore, multiple actors impose different representations of the system which a study needs to account for. Verbraeck and Valentin (2006) provide six common reasons to start a simulation study:

1. the system in question does not exist and the development of a real one / pilot for the purposes of studying, imposes insurmountable issues;
2. the system in question does exist but ‘playing around’ with the system imposes insurmountable issues;
3. the system administration lacks crucial data to improve the system or develop a new one;
4. the study requires to investigate system behavior non real time;
5. an analytic modeling approach cannot or barely provide insights or answers;
6. the process elements of the system do not have a straightforward mathematical solution.

The modeling of the interlocking engineering design touches points 2 through 6:

2. The engineering design of an interlocking consumes a substantial investment and demands a lot from project management to keep real projects running. Furthermore, the safety of the final design may never come into danger. Therefore, executing one or two test/pilot projects seems plausible but not an extensive optimization study that involves multiple project types and process structures.
3. The research in previous chapters revealed value added and non value added process times for the design stage and value added cost and time figures of a few interlocking projects regarding the engineering stage. A distinction misses of non value added time into queue time, NVA-WIP time, NVAW time, mistake time and additional time of other ‘waste’ factors. In fact, the non value added time of the engineering process misses in its entirety. Besides, the effect of possible replications sticks to general probability indications.
4. RailML and the ideal lean approach require (relative) insight into that data in order to derive conclusions on their performance. Simulation enables the estimation of the distinction by evaluating the effect of project arrival patterns, consequent queues and insertion of variable/stochastic process characteristics like time and errors.
4. The total processing time of an engineering design project ranges from 2 to 10 years. Studying a multitude of project types would thus take decades in real

time. Simulation allows to review several decades in a few minutes.

5. The lean philosophy aims to achieve as much continuous project flow as possible. The factors that influence flow and the depiction of flow behavior per project cannot or very hard be described using analytic modeling. Examples of factors that influence flow in the interlocking engineering design include allocation of process resources, interdependencies (e.g. process A may only continue upon completion of process B) and project specific decisions as process repetitions. Relevant flow behavior encompasses performance statistics like half width and variability/extremes, resource utilization rates, the effect of different project type ratios and so on.
6. Well known mathematical theories can compute / estimate process structures like parallel processing during the RVTO and SWOD, probable process replications after validation, process variability due to normally distributed process times and queues due to a random project arrival pattern over time. However when a system includes multiple of these structures and processes, mathematical modeling quickly gets very complex. In that case, simulation, especially with the use of a program with convenient GUI, makes modeling less time intensive and decreases the probability of modeling errors. The amount of modeling errors also drops because most simulation programs validate definitions. The program would e.g. recognize that the model does not connect two processes while a mathematical approach does not.

Chapter 3 summarizes a comparison of different (simulation) modeling programs to tackle an engineering design issue. ARENA resulted as the best option available for this study.

5.2.2 The ARENA Base Model

In general, a base or status quo model reflects the current engineering design chain and allows to assess the relative performance improvement of a status quo with RailML. The base model also deals with validation of the modeling strategy since it needs to match today's performance figures. A second derivative of the base model comprises an ideal lean process structure following chapter 3; which serves as a benchmark for RailML's improvement. This chapter aims to picture the improvement of RailML to the current situation and the leanness of RailML in section 5.3.

The ARENA modeling followed the discrete simulation modeling cycle as proposed by Verbraeck and Valentin (2006): start with a model conceptualization, accordingly specify your model in the simulation program environment, then verify and validate the model and finally develop models that reflect solutions and measure their performance; if added insight, alter the conceptualization. This subsection explains the main modeling concepts of the base model and the two derivatives since each has been elaborated on extensively in previous chapters. Appendix L documents the model setup by elaborating on the conceptualization, verification, validation and model parameters.

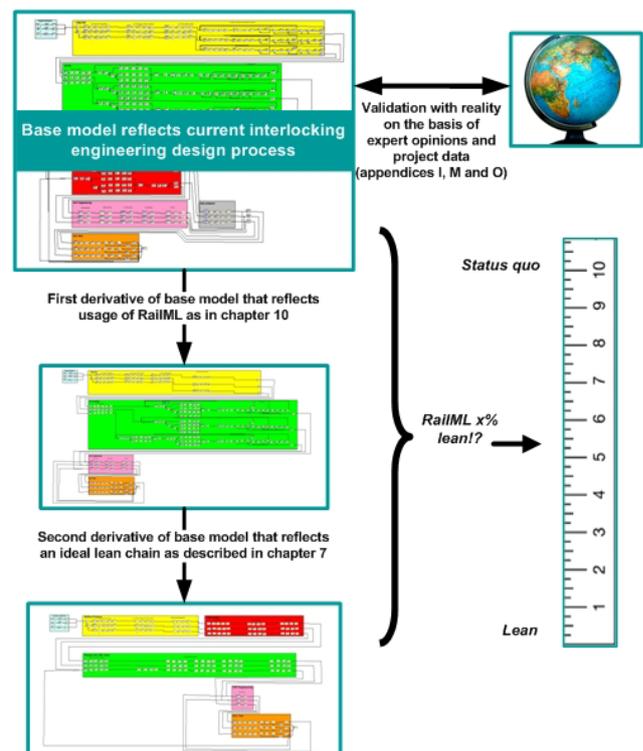


Figure 72: The modeling strategy of this section. The strategy aims to define the extent to which RailML achieves an ideal lean interlocking engineering design structure.

The base model bases itself on chapters 2 and 3 and corresponding appendices in order to reflect the current process structure and performance (Figure 72 & Figure 73). The base model contains eight parts: the creation of projects, the CRS/FIS processes, the RVTO processes, the SWOD processes, parallel EVP engineering and test protocol processes, the dry test process and disposal of the projects from the simulation. A transaction characterizes the distinction between processes which includes a data transfer, queue time and costs.

The creation of projects consists out of three creation possibilities: low complexity projects, medium complexity projects and high complexity projects. In order to align the definition of project complexities with Siemens' project data that relates only to the engineering processes, small interlocking projects correspond to low complexity projects in the model, big and ETCS interlocking projects correspond to medium complexity projects and new interlocking projects to high complexity projects. The project generation bases itself on the average amount of projects over the last three years and the coming year.

For the process representations in the ARENA model generally counts that it composes out of three computations: first a project claims a resource, second the project has some non value added process time and third the effective process times. The first computation works on the basis of queuing theory; the latter two computations on the basis of a normal distribution. The validation made sure that queuing time plus non value added processing time equals the non value added time as indicated by the experts in Appendix D. The only exception includes the EVP engineering processes since Siemens did not provide non value added processing times. Besides computation structure,

each process demands a resource from the same pool. This implies that project teams can handle a low, medium and high complexity project. The amount of resources does not directly reflect the amount of project teams or employees as group sizes vary and employees may work on multiple projects at the same time.

The CRS/FIS processes in ARENA miss a replication possibility between CRS development and FIS alternative development. The reason is that the CRS changes iteratively and to include this in the model would require extensive knowledge to derive a representative picture. The validation of the project can however result in replication of FIS' system requirements since a more transparent replication behavior. In general, the FIS never gets accepted at once and there exists a small possibility for another replication. The probability for a fourth replication is set to zero.

The RVTO process contains parallel processing of the rail maps, GIS maps and RVTO report; a project may only continue when ARENA finishes every parallel process. Furthermore, the state of final design and the three parallel processes use the same resources. The model includes the possibility of replicating the parallel processes after RVTO validation. ARENA represents the replication probability in the same way as after the FIS.

The SWOD has a parallel process structure of OBE, OR and SVA development; comparable to the parallel processes during the RVTO. The project seizes one resource before processing and may only release it / continue to the next process when all three processes are finished.

The EVP engineering and test protocol processes run parallel too. The project teams however need to finish the test protocol and EVP engineering before starting with the dry test. Therefore, the test protocol could actually be left out of the model since the process times are much shorter than the average time it takes to accomplish all EVP engineering activities. The model still contains the test protocol to account for the exceptional case of a strong process deviation, although the effect would probably be negligible.

The dry test process could in rare events conclude that the EVP does not match the requirements. Such an event could occur from the start of the SWOD since Prorail does not execute a validation process after the RVTO. Therefore, worst case scenario, the SWOD, EVP engineering and test protocol require reconsideration.

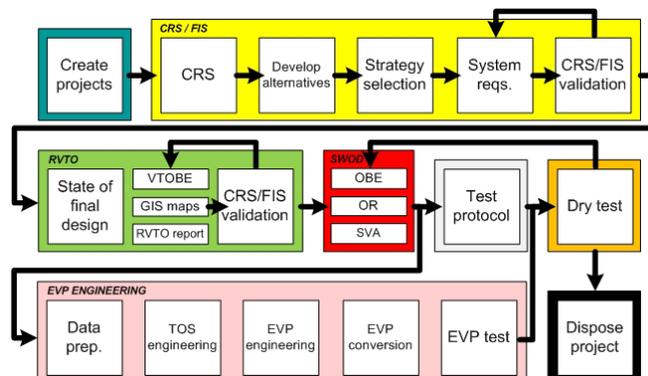


Figure 73: A strongly simplified overview of the ARENA base model. Each arrow represents a transaction and/or replication.

5.2.3 The ARENA RailML Model

The RailML process model changes the structure in Figure 73 on the basis of section 5.1 (Figure 74). An adjustment to the base model is chosen over a completely new model for three reasons. First, only structural changes, e.g. RailML that makes data preparation during the EVP engineering superfluous, will happen with reasonable certainty. In contrast, the extent in which process times change, cannot be concluded with reasonable certainty. Second, changing the base model allows for easy and fair performance comparison. Third, building a new model requires substantially more time. **The model assumes the existence of an EAI (e.g. a GIS) next to RailML because RailML cannot provide full functionality in the absence of one central data source (consider section 5.1).**

Discussing the changes from the model's start to the end, the first change eliminates the CRS/FIS validation process. Section 5.1 identified that by working with RailML, the system requirements process adjusts the former OBE and OS data into a final future concept exchangeable by RailML. RailML, or an associated tool, allows for automatic validation of the concept due to which it becomes superfluous to check whether the format is correct.

Second, the RailML model includes a combination of the SWOD and RVTO. RailML only enables this combination for the low and medium complexity projects since a high complexity project encompasses new interlocking areas that cannot rely on past data. Therefore, in the case of low and medium complexity projects, the model runs the processes of RVTO's interlocking area specification in parallel with the SWOD processes. RailML allows the elimination of some of those processes as well. A small part of developing future operational track maps remains, now that the process can rely on existing data. Therefore, the model combines future track development with SWOD's OBE development since both correspond to a big extent. RailML reduces the creation of geomaps and data during the RVTO as well but since it does not correspond to another process, it will stay a separate process. RailML and the combination of RVTO and SWOD consequently eliminate the need to document the RVTO processes. In case of high complexity projects, there still remains a need to develop OBE, OS and geographical data from scratch. Therefore, a separation between RVTO and SWOD still remains in the model for high complexity projects although RailML makes the RVTO documentation redundant.

Third, section 5.1 identifies that RailML will eliminate the need to prepare data and engineer a TOS during the EVP engineering stage. The model therefore excludes those processes.

Fourth, section 5.1 also identified that the test protocol development almost becomes insignificant because RailML with the right IT tools can automate this process. Therefore, the RailML model does not contain a test protocol process anymore.

Fifth, the result of an insufficient dry test does not lead to a replication of the SWOD since that process now occurs before a design validation. Therefore, an insufficient SWOD may only lead to replication of the EVP engineering processes.

Table 16: The parameter differences for the RailML model in a worst case run and an optimistic run. The amount of days concern the average non value added time plus value added time.

Parameter in RailML model	Worst case	Optimistic case
General		
Probability for one replication of RVTO / SWOD	100%	50%
Probability for second replication of RVTO / SWOD	25%	12.5%
Probability for third replication of RVTO / SWOD	5%	2.5%
Probability for replication of EVP engineering	15%	7.5%
Low complexity projects		
OBE development	24.7 days	18.53 days
OR development	24.7 days	18.53 days
SVA development	24.7 days	20.65 days
Geodata development	36.2 days	20.65 days
Medium complexity projects		
OBE development	51.8 days	38.85 days
OR development	51.8 days	38.85 days
SVA development	51.8 days	41.5 days
Geodata development	83.3 days	41.5 days
High complexity projects		
Track map development	994 days	745.5 days
Geodata development	994 days	745.5 days
OBE development	185 days	138.75 days
OR development	185 days	138.75 days
SVA development	185 days	153.5 days

Besides structural changes, the parameters in the RailML model change as well (Table 16). Two parameter variants are developed: a worst case scenario model and an optimistic scenario model. These two scenarios allow the representation of a performance bandwidth in section 12.3. The first parameter difference occurs during high complexity's development of track maps and geodata or RVTO in the current process structure. In a worst case scenario, those processes take as long as the status quo. Although with the elimination of a RVTO report, an optimistic scenario would value the process time at two thirds of the original. The second parameter difference occurs a process step later, when specifying the interlocking area. In general, RailML could speed up this process due to the use of IT tools. In an optimistic scenario, a 25% reduction of process times seems plausible. Furthermore, in a worst case scenario for low and medium complexity projects, the parallel processing of geo maps with OBE, OR and SVA development, takes as long as the original sequential process structure. In an optimistic scenario the development of geo maps does not exceed the average time to process the OBE, OR and SVA. The process time of the SVA equals two thirds of the new OBE or OR development time plus a third of the current SVA development time.

The third parameter difference concerns the probability that validation leads to a replication of the interlocking area specification. In a worst case scenario, the probabilities stay the same as in the status quo. RailML however reduces the probability of making errors mainly by standardizing transfers and reducing the amount of transfers. Therefore, half the current probabilities seems possible. The fourth parameter concerns the dry test validation's probability to result in replication of the EVP engineering. For the same reasons as with the RVTO/SWOD validation it makes sense that fewer replications are required.

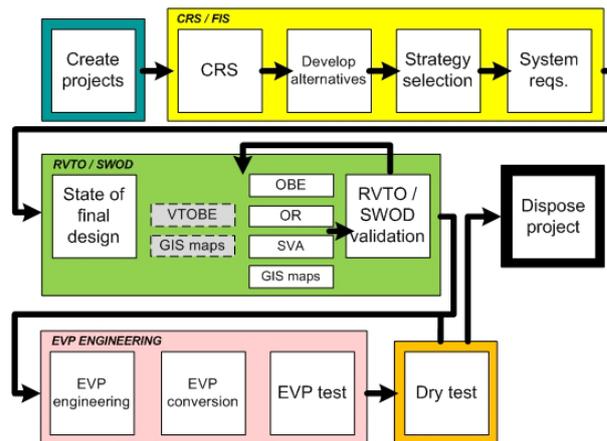


Figure 74: A strongly simplified overview of the ARENA RailML model. Each arrow represents a transaction and/or replication.

5.2.4 The ARENA Lean Model

The second variation to the base model concerns a lean engineering design chain to serve as a benchmark according to chapter 3 (Figure 75). The next describes the changes compared to the RailML model that lead to the lean model. Whereas the introduction of RailML directly tackles four of the five high potential transformation directions (consider chapter 3: 'waste' prioritization, process standardization, open source EAI and information sharing), the modular process design results from process management decisions that RailML does not affect. In production systems, a lean modular design approach comes down on resources, whether machine, robot or human, aligned in a continuous process that deal with a small part of the product. The automotive industry applies this strategy for example by creating a sequence of jobs, sometimes along a conveyor, that consume the same or an integer divisible average process time. On average, each process ends at the same time and all parts move together to the next process. As a result, the structure almost eliminates all queues and specializes employees to reduce errors. A modular design approach in an engineering design process imposes some difficulties. From an infinite amount of the same product type, it is straightforward to determine average process times and corresponding variation. That does not count for an engineering design process in which projects vary greatly by a wide range of factors. Furthermore, one cannot as easily split a design process into parts / modules as with a product. In addition, little to no literature exists about a modular design process. The lean model therefore tries a strategy on the basis of sepa-

rately processing project categories and minimizing the amount of process flows by reducing the amount of different project teams. The model reflects separate processing by using pools of resources per level of project complexity instead of one pool for the three levels. The model also redefines the process stages into five parts of flow on the basis of chapter 3. In short, this implies a project definition flow with representatives from all stakeholders, a data preparation flow with employees from Prorail, an interlocking area design process at an engineering agency, EVP engineering at Siemens and a dry test at Prorail. Figure 75 visualizes the flow difference by the amount of arrows compared to Figure 73 and Figure 74.

Chapter 3 explains how all parties in the chain could theoretically achieve such a lean state; the model makes six structural and two main parameter changes for that purpose. The first structural change removes the Prorail selection process from the CRS/FIS because with the right IT tools, Prorail can execute these processes themselves and a transfer for selection purposes becomes obsolete. The second structural change splits a part of the current FIS and RVTO's state of final design to develop a digital workspace that contains all necessary files for the engineering design chain. Third, the model changes the parallel process structure of the combined RVTO/SWOD in the RailML model to a sequential one. Fourth, a lean interlocking area design stage does not involve a validation process anymore since it does not directly add value to the final product. Validation remains important but one time, just before production, seems sufficient. Fifth, section 5.1 indicates that in the long run, the EVP conversion and EVP test processes would not add additional value to the final project as a standardized RailML interlocking approach might always result in the desired interlocking architecture. Lean theory calls this learning effect 'perfection'. Therefore, the lean model eliminates both processes. The sixth structural change occurs at the dry test where a replication of high complexity projects does not only include the EVP engineering but also the interlocking area design stage. This results from the fact that a new interlocking project cannot be based on previous interlocking data. Furthermore, with the absence of an interlocking area design validation process, the dry test's inconsistency could result from the interlocking area design.

The lean model comprises two general changes of parameters. First, the new processes during the data preparation stage require parameters. The 'data requirements' and 'gather and relate data' processes currently form a part of the system requirement definition. Therefore, it is assumed that each of these three processes consumes a third of the system requirement definition. The data gap analysis now forms a part of RVTO's state of final design and the lean model equally shares the process time between the two processes. The sequential structure of the interlocking area design causes the second change of parameters. Parallel processes benefit from a synergy effect that makes a sequential process take longer. The model assumes this synergy effect to reduce process time by one third. Therefore, it is assumed that the (VT-)OBE, OR, geo data and SVA processes consume half of their associated total parallel time.

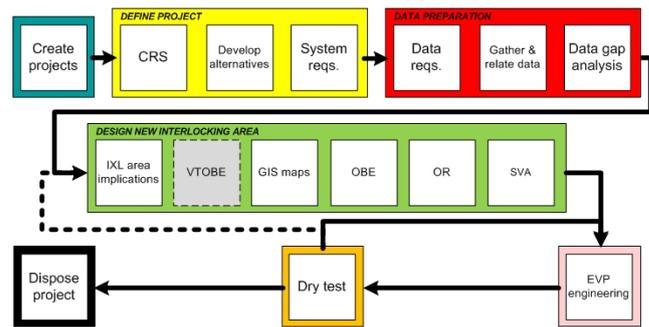


Figure 75: A strongly simplified overview of the ARENA RailML model. Each arrow represents a transaction and/or replication

5.2.5 ARENA Model Specification

The specification discusses the three main steps in the chain (project initiation, design and engineering) and the general simulation setup.

Project Definition

The interlocking engineering design chain starts when Prorail initiates a project. This means that Prorail finished the legal procedures and closed project budget. The simulation starts at that point in time. This means that ARENA needs to know the arrival rate of interlocking projects that run as entities through the simulation. An interlocking project being defined in three levels of complexity. Furthermore, projects do not always arrive exactly as planned and nicely divided over the year. Therefore, ARENA also needs to know the arrival distribution in order to mimic reality.

Prorail provides a public list of rail projects on their website which provides a fair indication of the amount of interlocking projects (Prorail, 2013b). An assessment of each project's planning and trace decision leads to a yearly project overview as defined in Table 17. Assuming an average of 2 switches for low complexity, 10 for medium complexity and 30 for new complexity projects, leads to 140 and 164 replaced switches for 2013 and 2014 respectively. Prorail defines an average of 163 replaced switches over the last 3 years (Prorail, 2013a), which validates the figures of Table 17. Due to conflicting values, the worst case scenario defines the parameters (i.e. the 2014 figures).

Table 17: The amount of interlocking projects per complexity level for 2013 and 2014 based on Prorail (2013b).

Complexity	2013		2014	
	Amount	Daily rate	Amount	Daily rate
Low	10	0.027	17	0.047
Medium	6	0.016	7	0.019
High	2	0.005	2	0.005

A constant daily rate as shown in Table 17 does not hold in reality. When assuming that the projects arrive independently divided over the year, projects can start somewhat earlier or later than the average daily inter arrival times. Usually five families of distributions from queuing theory apply in the case of simulation: (1) exponential/gamma/Erlang/Weibull, (2) normal/lognormal/Johnson, (3) uniform, (4) triangular and (5) constant/discrete (Duinkerken, 2011; Hillier & Liebermann, 2001; Rockwell

Automation Inc, 2010). A constant distribution, as well as a uniform and triangular distribution, cannot represent the distribution well due to a specifically limited arrival interval. A distribution of the normal family could not represent the arrival rate either as it allows for negative values. The exponential distribution then seems the most obvious distribution as one can easily interpret the meaning (only requires the mean inter arrival rate). Furthermore, modelers very often use the distribution to model the inter arrival time between two events with the same rate of occurrence. In addition, the distribution results in a higher probability of very short inter arrival times compared to very long ones. In other words, a consecutive project would have a higher probability of arriving in the second rather than the 10th year; which makes sense. The gamma/Erlang/Weibull distributions would apply in the case that the inter arrival time of consecutive arrivals after two or more than two projects should be assessed. In this case, the simulation program needs to know when to create the next project in the system, which makes the gamma/Erlang/Weibull distribution inadequate.

Interlocking Design Processes

The design process contains three main sub processes (CRS/FIS, RVTO and SWOD) of which the specification follows mostly from the mapped value stream (Appendix D). The specification challenge comprises that each complexity level approximates the stated idle and process times with the introduction of process time variability. ARENA complicates the challenge by generally facilitating a traditional FIFO queue configuration, i.e. when an entity requires a resource to undergo a process, it stands in line until a resource becomes available and gets processed until it is finished. In a design process, such a clear structure does not hold. Usually, project members “hop” from project to project. As such, a project arrives at a worker, waits, gets processed a bit, waits, get processed a bit and so on. A standard seize resource, delay entity for processing and release resource approach can therefore not reflect the current design structure. The queue waiting times will namely be equal (in the long run) for each project when sharing the same resource or project team; regardless of project complexity. In reality, as the figures of section 3.3 prove and Figure 76 illustrates, this is not the case as designers put complex projects more aside than simpler projects due to the longer value added process times.

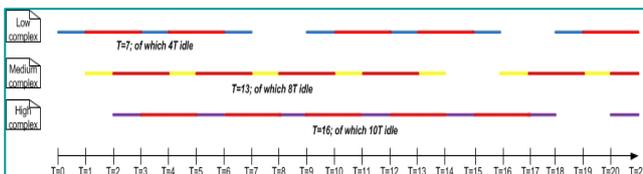


Figure 76: Hypothetical schedule of an employee that works on three different projects. Each project of different project complexity. A red line corresponds to NVA WIP. The blue, yellow and purple indicate value added WIP for a low, medium and high complexity project respectively. One project corresponds to a connected chain of red and blue/yellow/purple lines.

In order to specify the design processes of projects according to Figure 76, the ARENA model separates the queue, NVA WIP and value adding parts of a process (Figure 77).

Queuing represents a design process’ waiting time in case of full occupation of resources / project teams. The NVA WIP part reflects the non-processing time of projects after initiation of the design process. Both waiting times should match the indicated non value added process times projected in Appendix D. The value adding part represents the effective transformation after initiation of the design process. The latter two parts of the design process follow a normal distribution in order to account for variability. The mean of the normal distribution equals the time as indicated by Prorail employees (section 3.3 and Appendix F). The standard deviation bases on an educated guess of 10% the average processing time. In that case a range from ‘mean-10%’ till ‘mean+10%’ covers 95.45% of all process times, which seems reasonable.

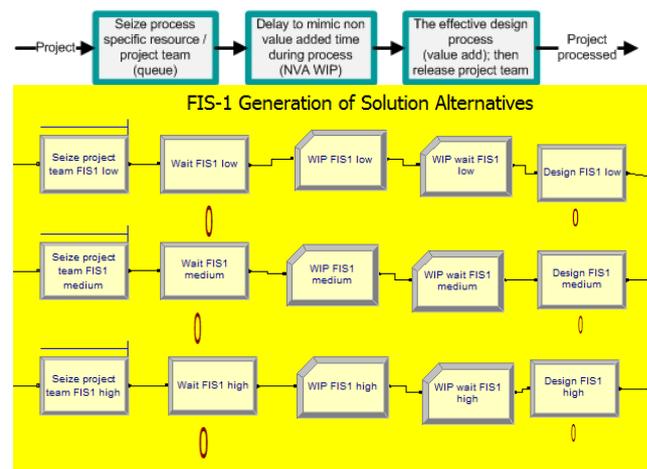


Figure 77: Shows the basic modeling structure of each design process in ARENA. Top picture shows the conceptual process; bottom picture shows how it looks in ARENA. A square in ARENA equals a process that can seize, delay or release an entity. A line on top of a square visualizes a queue. A square with a top left cove is a counter for performance evaluation.

The amount of resources / project teams per project matches the minimal amount to mimic reality. After all, each additional project member raises costs of the engineering design. The specification and validation aim to match the sum of the waiting and idle process times with the total non value adding time as stated by experts and visualized in the VSM of Appendix D. A rule of thumb for the amount of resources sums the number of days required to process all projects in a year (Equation 3) and divides this over 365 days. Then, the result gets multiplied by 30% to account for variability in arrival times. A utilization rate of 100% will namely always be accompanied with massive queues; which does not occur. Table 18 shows this process.

Equation 3: The applied rule of thumb for the amount of resources in the simulation model

$$Amount\ of\ resources_p = \frac{((17 * LCT_p) + (7 * MCT_p) + (2 * HCT_p))}{365} * 1,3$$

Where:
 p = a particular process
 LCT = a low complexity projects’ total process duration [days]
 MCT = a medium complexity projects’ total process duration [days]
 HCT = a high complexity projects’ total process duration [days]

Table 18: The results when applying Equation 3 for the interlocking engineering design process

Resources	# of resources at 100% occupancy	Rounded # of resources at 70% occupancy
CRS	9.01369863	12
FIS alternatives	10.26794521	14
FIS selection	6.850410959	9
FIS requirements	2.704109589	4
FIS validation	1.687671233	2
RVTO	22.01643836	29
RVTO validation	2.991780822	2
SWOD	3.452054795	5
Data prep	1.243835616	2
TOS	1.436986301	2
EVP	3.358082192	5
EVP conversion	0.943287671	2
EVP test	7.642739726	10
Test protocol	1.476712329	2
Dry test	0.709589041	1

The specification of some processes differs from the value stream map in Appendix D. First, the VSM does not specify Prorail’s alternative selection process and Prorail’s FIS validation. The alternative generation process also covers alternative selection; the system requirements also cover validation. Experts (Koelewijn et al., 2013; Storck & Dragt, 2013) revealed that the selection of the final alternative takes about 40% of total alternative development time. The relatively large share results from an elaborate shareholder process to reach an agreement on one way forward. Furthermore, the experts also indicated that they will almost always reject a FIS and RVTO once. One FIS and RVTO validation process takes about two weeks for medium complexity projects; of which 5 days of effective processing / adding value. The model assumes that a low complexity project needs half of that time and a high complexity project double that validation time. In addition to a specification of the validation time, the validation process requires a specification of the probabilities that a project gets rejected more than once. The model assumes that Prorail always rejects a project once, then with a 25% chance for a second rejection and a 5% chance on a third rejection (Figure 78). More than three project rejections seems a very exceptional case.

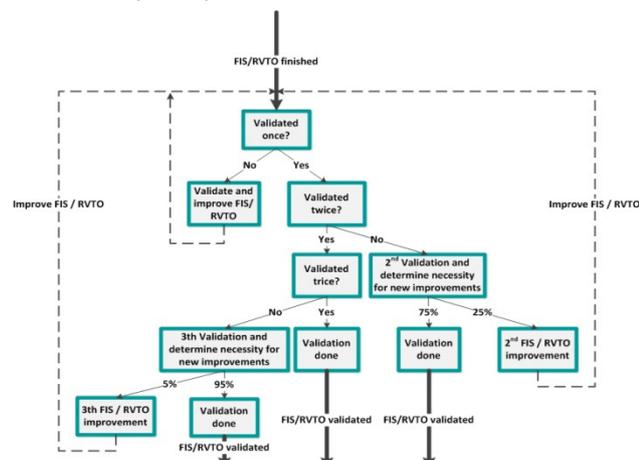


Figure 78: The probability tree that shows when the model assigns a project to be validated / redo the RVTO and SWOD processes.

Second, one project team executes all RVTO (sub) processes with the exception of validation. Therefore, the modeled structure slightly differs from that in Figure 77. In this case, a project seizes a project team and releases it only after completion of all RVTO processes instead of after one sub process (Figure 79).

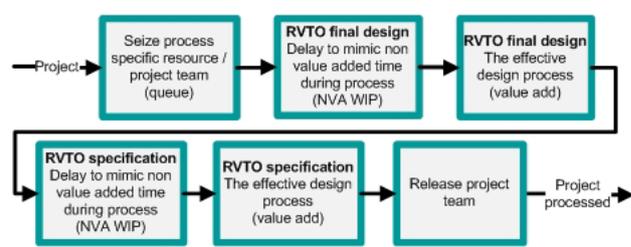


Figure 79: The model uses one resource to process the RVTO completely by first seizing a project team, then running the various processes and afterwards releasing the project team again.

Third, the RVTO interlocking specification’s sub processes and the SWOD processes occur at the same time instead of sequentially. Therefore, the model generates two duplicates of a project entity to allow for parallel processes. The project needs to wait for the last of these processes to finish before the project may continue to the SWOD process (Figure 80). An ARENA batch module allows for this function.

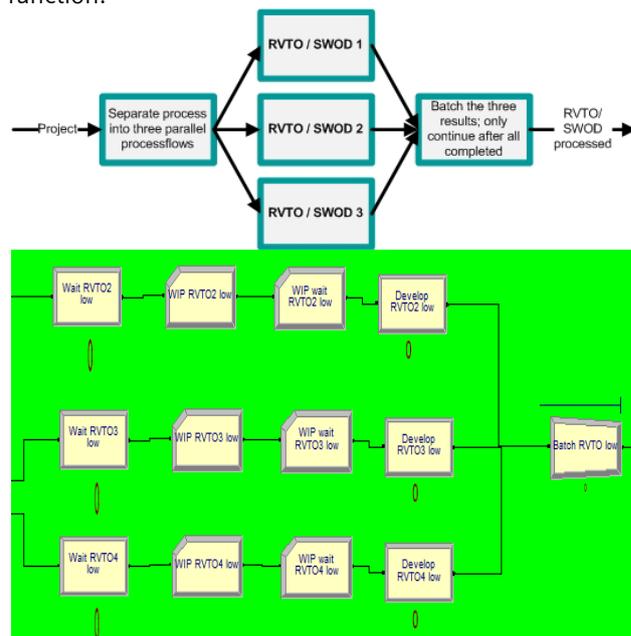


Figure 80: The top picture shows the schematic idea of the parallel RVTO / SWOD processes shown in the lower picture. A square in ARENA equals a process that can seize, delay or release an entity. A line on top of a square visualizes a queue. A square with a top left cover is a counter for performance evaluation. A converging block merges the duplicates into one again.

The Engineering Process

The modeling structure of the engineering process differs somewhat to that of the design process due to a different nature of the available process data. Siemens delivers data on the process times and associated costs for each of their engineering processes. The data does specify the share of idle time per process. Therefore, a standard seize project team, delay for processing and release project team struc-

ture reflects each engineering process. This structure does only quantify the process time and not the delay as occurs in the modeled design processes. The process times do still follow a normal distribution with the average process times as mean and 10% of the process time as standard deviation. A low complexity project corresponds to a small modification project in the Siemens data. A medium complexity project corresponds to a big modification project and an ETCS project. A high complexity project corresponds to a completely new interlocking project. The model includes the time and costs per level of complexity by taking the average over the various corridors per engineering process. The resulting time and cost values represent the total amount of hours made and does not correct for the amount of workers. An expert (van 't Hoff, 2013) and interlocking project planning schemas identify that big interlocking modifications take about a year and new interlocking projects take about 1.5 years at Siemens'. The average process times have been scaled to those durations so that they account for the amount of employees that work on a project. The figures for small interlocking modifications correspond to reality and have not been scaled. Due to reasons of confidentiality, this report does not present the averages.

Apart from the different modeling structure of Siemens' engineering processes, it does not count for the test protocol development and dry test processes since Prorail executes those. Prorail provided the same type of figures for those processes as with the design processes. The test protocol process occurs parallel to the engineering activities at Siemens. As such the model represents the same parallel modeling structure as with the RVTO and SWOD (Figure 81).

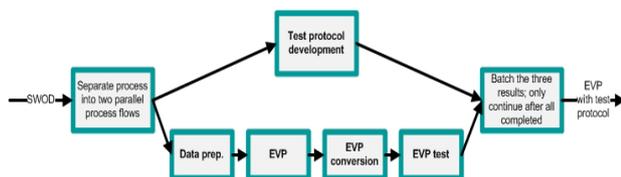


Figure 81: The interlocking engineering and test protocol process flows that both need to be finished before a final dry test can start.

One final remarkable modeling structure locates at the end of the dry test. A dry test could result in a rejection of the EVP. Since the cause could arise during the SWOD and EVP engineering because both do not have a validation process by Prorail, the project needs to undergo all SWOD and EVP engineering processes again. Experts confirm that rejection occurs every once in a while but could not specify the amount of projects that get rejected at this point. Therefore, an educated guess specifies the amount at 15%.

Simulation Setup

The simulation requires a running length, warm-up period and amount of replications before it can actually start. The running length defines the time in which the simulation runs entities through the processes. The warm-up period defines the initial time to bring a system to its steady state. Since the measurement report does not include statistics of the warm-up period, the running length minus

the warm-up period defines the time that the simulation actually measures statistics. The simulation program assigns a random seed to each run that results in different outcome of the probability functions. Therefore, multiple repetitions or replications of the simulation ensure a representative collection of statistics.

The warm-up period is a part of the running length and therefore needs to be defined first. The simulation continuously executes the arrival process according to Figure 17. Furthermore, the model includes only one arrival point of entities. This implies that when the first generated project entity reaches the end of the model, all processes are activated at least once. A first run of 10950 days (30 years) with 10 replications identifies that the maximum time for a high complexity project is about 9000 days. In a worst case scenario, that particular entity could have been generated at the start of the simulation. Therefore, the model applies a warm-up period of 10000 days to account for a long run of the first entity plus an uncertainty margin. The simulation model applies a warm-up period for each replication over a continuous simulation with one warm-up period but multiple replication sections because that delivers the best results. The disadvantage however is a longer running time but since each warm-up period takes about 30 seconds of calculation time with animation, that is not a major issue.

The simulation model runs for 20950 days. The 20950 days includes a warm-up period of 10000 days and a statistic collection time of 10950 days (30 years). The first estimate of 30 years proved sufficient because:

- the average process time of a low complexity project is about 550 days; on average, this results in $((10950-550)/365) * 17 = 484$ low complexity project entities;
- the average process time of a medium complexity project is about 1200 days; on average, this results in $((10950-1200)/365) * 7 = 186$ medium complexity project entities;
- the average process time of a high complexity project is about 4500 days; on average, this results in $((10950-4500)/365) * 2 = 35$ high complexity project entities.

Statistical theory usually requires a minimum of 30 cases from one experiment to assume a normal distribution (Vocht, 2008). Of course, one should still test whether the normal distribution holds by means of a Kolmogorov-Smirnov test for example. When it does, the interpretation becomes more straightforward and analyses more reliable. Therefore, an average minimum of 35 high complexity projects for one run is just sufficient.

The amount of replications depends on two factors: the real life simulation time and the size of the half width. Twice the half width indicates the 95% confidence interval of process times. A lower half width thus increases the certainty of the estimated performance. Half width reduction however requires more replications. Verbraeck and Valentin (2006) define that the relation between half width and amount of replications goes as in Equation 4.

Equation 4: The mathematical estimation between the half width of a process result and the amount of simulation replications (Verbraeck & Valentin, 2006).

$$n' = \left\lceil n \left(\frac{h}{h'} \right)^2 \right\rceil$$

Where:

- n' = the needed number of replications
- n = the current number of replications
- h = the half width at n replications
- h' = the desired half width

Equation 4 however estimates the amount of replications. The formula does not provide an exact solution because of the stochastic nature. Equation 4 therefore presents the estimated required amount of replications to achieve a certain half width at three levels of complexity. A test simulation of 10 replications provides a half width of 23 days of low complexity projects' total process time, 34 days of medium complexity projects' total process time and 86 days of high complexity projects' total process time. These results form the basis of Figure 81.

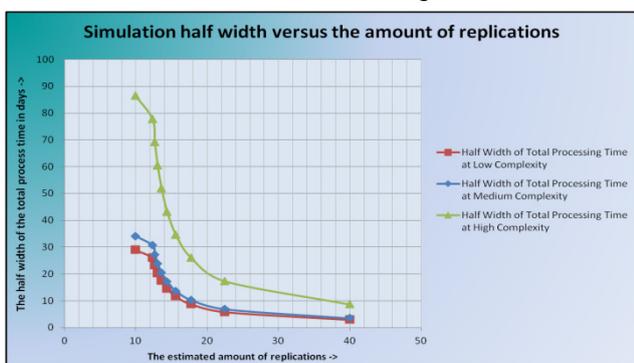


Figure 82: The target half width versus the estimated amount of replications based on Equation 4 and a start simulation with 10 replications. Each point represents a decrease of half width by 10%.

Figure 82 shows that the required number of additional replications to lower the half width by 10% becomes very high from 22.5 replications. Furthermore, each additional replication takes 1 minute of animated simulation time and 7 seconds of non animated simulation time. Since a substantial number of validation tests are required, the increase of replications from 22.5 to 40 does not outweigh the 10% reduction in half width anymore. Besides, the largest half width, of the high complexity project, is 17.3 days. This means that the 95% of the high complexity projects' total process times will either be 17.3 days lower or higher than the mean. At an average total processing time of about 4500 days, this is a very decent estimate. Therefore, the simulation replicates the running time 23 times, rounded upwards.

5.3 The Lean Performance of RailML

This section compares the effects of an interlocking engineering design process using RailML with the status quo and the lean structure. The results follow from the simulation models as defined in the previous sections. This section elaborates on the effects per performance indicator as defined in section 2.6. Subsection 5.3.1 discusses the results regarding cost figures. Subsection 5.3.2 continues to elaborate on the amount of ambiguous processes between the three models. Subsection 5.3.3 elaborates on

the throughput time figures and subsection 5.3.4 presents the differences in process interdependencies.

5.3.1 The Differences in Cost Figures between the Status Quo, RailML and Lean Models

Table 5 provides performance metrics to assess the cost criterion in the case of the particular engineering design process; project cost, productivity rates and the 3C value leveraging model assess Siemens' activities in particular and the lean transaction cost efficiency method assesses the cost performance of the chain.

Figure 83 shows that RailML will reduce Siemens' costs associated to interlocking engineering. The reduction comes primarily from the elimination of the data preparation and TOS engineering processes. Although RailML leads to a significant cost reduction of somewhat over 17% to 27%, a lean structured engineering design reduces costs about four times more. The lean process elimination of EVP conversion and EVP test cause this difference.

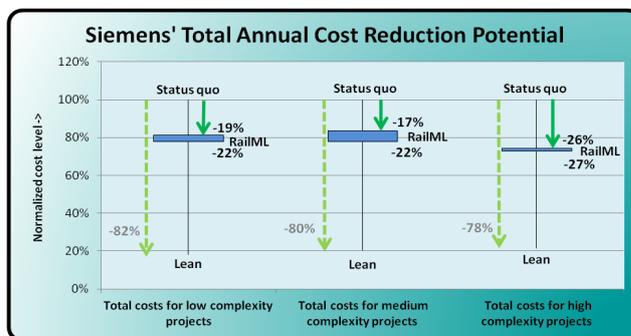


Figure 83: The cost reduction range with the introduction of RailML and the potential lean cost reduction for the total of engineering activities at Siemens. The dotted light green arrows indicate the lean performance improvement and the dark green arrows indicate RailML's performance improvement.

Figure 84 shows that RailML increases the productivity of resources mainly by a mix of reducing process times and resources. Productivity is measured as the amount of Euros generated per day of work per resource. The lean case however, improves productivity nine to ten times better than the RailML models. The most likely cause of this strong improvement again results from the process elimination of EVP test and EVP conversion. As a result, process times and the amount of resources drastically decline.

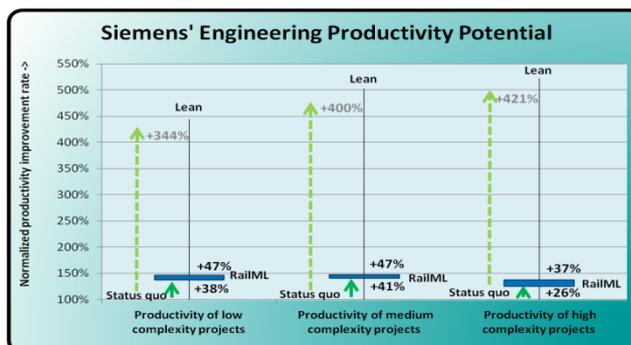


Figure 84: The productivity improvement range with the introduction of RailML and the potential lean cost improvement for the total of engineering activities at Siemens. The dotted light green arrows indicate the lean performance improvement and the dark green arrows indicate RailML's performance improvement.

Appendix C describes the 3C value leveraging model of W. Beelaerts van Blokland et al. (2009). The 3C model describes that three value drivers, conception, continuation and configuration, define the value-time curve. One may consider the value-time curve as the life-cycle curve but related to value generation along the project's lifespan instead of sales. The right balance of the three value drivers determines the competitive advantage and risk of the company in the long run. Competitive advantage would result in higher market share and risk on shorter break even time and lower debt rates.

The ARENA model allows measurement of the three value drivers in the following way:

- Conception - the investment multiplier (IMP). The IMP states the company's investment to develop a project or product. The higher the IMP, the lower the negative value during the project and the lower the project risk. Furthermore, lower cost for Siemens probably results in a lower price for the infrastructure manager as well. This competitive advantage may drive market share. This research estimates the IMP by taking the average cost reduction ratios between the models.
- Continuation - time to market (TTM), market share (MS) and break even time (BET). The TTM equals the average process time to engineer design a project. The market share reflects the amount of interlocking projects that Siemens deals with compared to the total amount of projects. A shorter TTM, just like a lower IMP, improves the competitive advantage and may increase market share. The BET indicates the time within the TTM from where the project generates net value / profit instead of loss. In an interlocking project, the infrastructure manager pays upfront to prevent a negative cash flow at industry. The industrial party may however legally receive a payment share after complementing a milestone. Therefore, accounting rules prevent a negative cash flow, though practically the company breaks even after reaching a certain amount of milestones. Then, limited margins locate the current breakeven point somewhere near the end of the curve. The BET position relative to TTM results from the production multiplier discussed at configuration.
- Configuration - the production multiplier (PMP). The production multiplier indicates Siemens' work share compared to the total work in the chain. The PMP therefore follows the ratio in the next equation:

Equation 5: the production multiplier function as applied in this research project.

$$PMP = \frac{\text{Chain's accumulated total process time in days}}{\text{Siemens' accumulated total process time in days}}$$

A higher PMP means that Siemens takes fewer risk and breaks even quicker.

Figure 85 shows the effects of the RailML and lean models in contrast to the status quo. The TTM gets shorter due to the elimination of processes. As a result, the IMP reduces as well since fewer processes require less investments. The simulation however also leads to the conclusion that Siemens' share of work relative to that of the chain goes down. In other words, the effect of RailML and the lean approach facilitate themselves better during the engineering stage than the design stage. The rate in which the RailML models improve the value-time curve however, is about a third to a fourth of the lean model. The elimination of the EVP conversion and EVP test contribute much to the significantly shorter TTM and IMP of the lean model compared to the RailML models. The PMP on the other hand, strengthens the conclusion that the design stage cannot incorporate the lean philosophy as well as the engineering stage does. The MS probably gets affected by the presented results as well. The model and the available data however provide insufficient ground to give a reliable estimate of the market share.

The lean transaction cost efficiency method enables an assessment of the chain's cost performance instead of only Siemens'. de Jong et al. (2013) develops the lean transaction cost efficiency method on the basis of the transaction cost theory of Oliver E. Williamson. The general definition of a transaction defines that a transaction occurs every time that "a good or service is transferred across a technologically separable interface" (Williamson, 1981). The lean transaction cost efficiency method aims to decrease the transaction costs for the purposes of a lean transformation. This implies that transaction costs should not just decrease in absolute terms but most and foremost in relative terms. In order to measure whether the RailML and lean models achieve that goal, de Jong et al. (2013) define three relations based on three performance metrics:

1. turnover per unit of fixed asset;
2. gross margin per unit of fixed asset;
3. inventory per unit of fixed asset.

Table 19: The quantitative results of the 3C-value method approach for the interlocking engineering process at Siemens. The table does not include absolute project TTM figures due to reasons of confidentiality.

	Market share		IMP		PM		TTM
	Absolute	Relative increase i.r.t. status quo [%]	Absolute	Relative improvement i.r.t. status quo [%]	Absolute	Relative improvement i.r.t. status quo [%]	Relative accumulated decrease i.r.t. status quo [%]
Current	17.6%	0%	1.00	0.0%	3.70	0.0%	0%
RailML worst	17.6%	0%	1.24	19.2%	4.43	16.4%	21%
RailML best	17.6%	0%	1.30	23.2%	4.03	8.0%	28%
Lean	17.6%	0%	5.00	80.0%	8.84	58.1%	78%

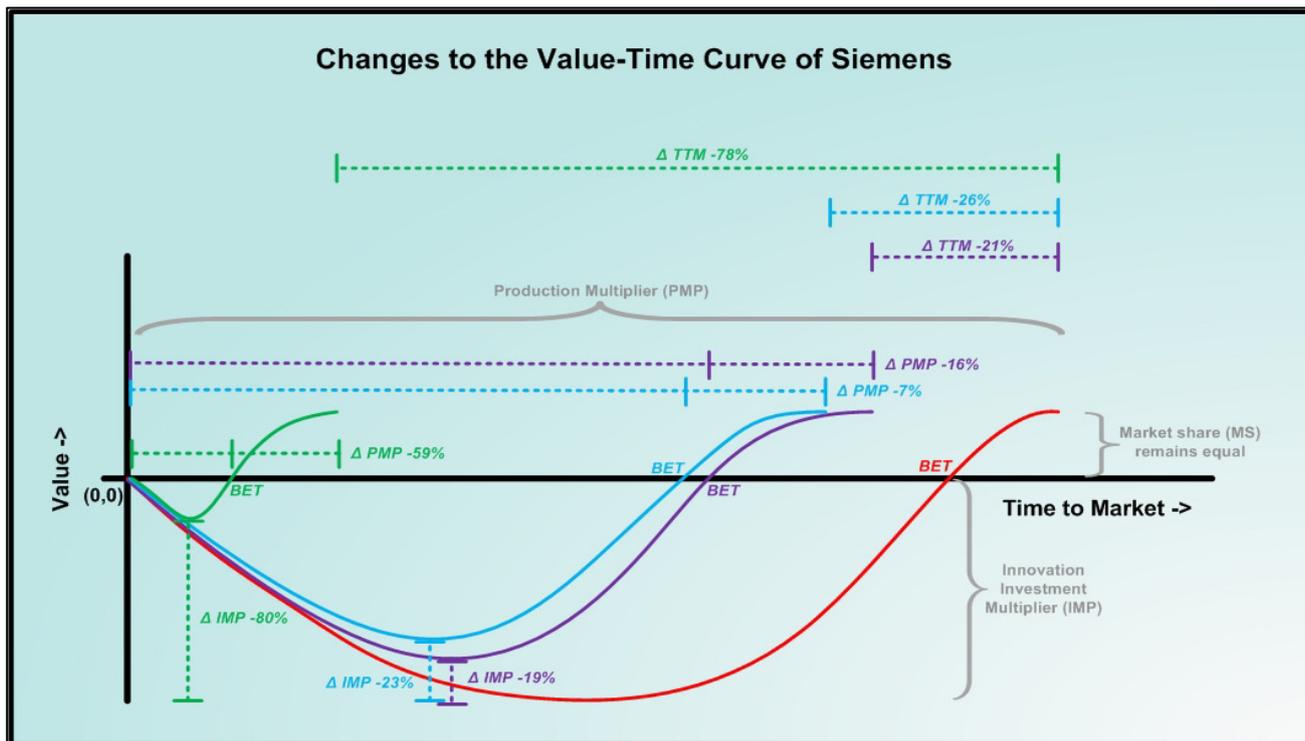


Figure 85: The value-time curve of the four different models on the basis of Table 19. The red line indicates the status quo, the purple line indicates the RailML worst case scenario, the blue line indicates the RailML best case scenario and the green line indicates the optimal lean situation. The dotted lines represent the improvement of the scenarios compared to the status quo. The figure is drawn on scale.

The data in this research does not contain margins; let alone reliable assumptions. Therefore, the model cannot determine the gross margin per unit of fixed asset. Turnover figures miss as well but total accumulated process time figures are used instead. Statistical analyses in section 3.3 identified that process times form by far the most important cost driver to achieve cost reductions. Therefore, process times provide a fair representation of the relative, not absolute, cost differences. Although costs differ from turnover, they are closely related since costs for one party means turnover for another. One should however inversely interpret costs: turnover needs to increase whereas costs need to decrease. The model does result in the size of non value added work (more in the next subsection) which is the inventory equivalent in an engineering design process. The denominator fixed assets usually include the value of property, plant and equipment (PPE). Production processes clearly have such PPE, engineering design processes do not. One could argue that office space could decline in the long run, but reliable estimations miss. Therefore, the application of the lean transaction cost metrics assumes a constant value for fixed assets.

The model allows to study one combination of performance metrics with the absence of the gross margin metric. Figure 86 illustrates that the quadrant graph allows for lean efficiency conclusions on the basis of a model's non value added work versus accumulated process time combination. The alpha angle of the status quo versus the beta angle of the final (model) concept describe the performance differences. These differences are also categorized by means of quadrants. In this particular comparison, the 1 quadrant indicates a cost improvement at the cost of having more NVAW. Although the relative performance in

cost per NVAW scores better than the status quo. The 2a quadrant indicates a worse NVAW and cost performance in absolute terms, but a relatively better one. The 2b quadrant indicates that cost and NVAW perform worse in relative terms too. The 3 quadrant indicates an improved NVAW level though the cost versus NVAW performance scores worse in relative terms. The 4a quadrant indicates an improved NVAW and cost level in absolute terms, but not in relative terms. A result in the 4b quadrant also performs better in relative terms. As a result of this classification, the preferred quadrant result according to the lean philosophy from best to worse is: 4b; 1; 2a; 4a; 3; 2b.

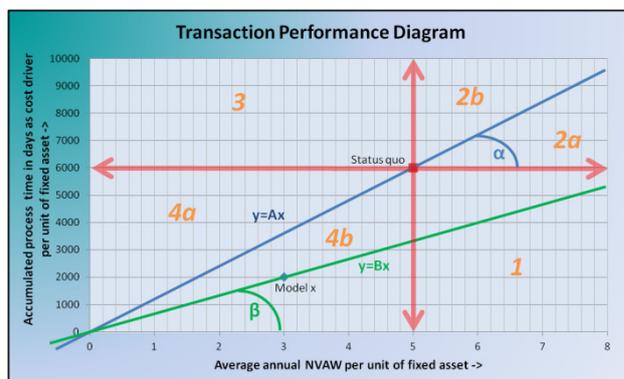


Figure 86: The transaction quadrant of NVAW per fixed asset versus cost per fixed asset based on de Jong et al. (2013).

Figure 87 presents that the transaction performance of low complexity projects in the RailML and lean models scores better than the status quo ($\beta < \alpha$). Transaction costs decrease for all three models though the RailML models

show that they require a larger NVAW in return; especially in a worst case scenario. The RailML models eliminate some processes, get longer (virtual) queues at other processes and consequently increase NVAW. The modular approach of the lean model mitigates this increase in NVAW by changing the project flow (section 5.2). Instead of a discrete flow structure where each project demands a resource and thus waits in a (virtual) queue at every process, the lean model combines the processes in one sequence per party in the chain. Furthermore, resources only process one type of project complexity instead of all three. Therefore, the number of queues reduces substantially and project flow increases.

The best alternative to the status quo results in 1.74 times less costs at the same NVAW levels. The RailML models relatively decrease costs by a factor 1.24 (RailML best case) to 1.43 (RailML worst case).

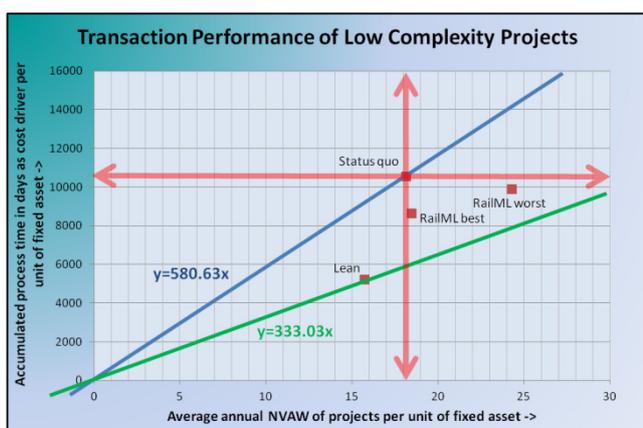


Figure 87: The transaction quadrant of low complexity projects in which the status quo is compared to the RailML and lean process chain models. The graph depicts a line from the origin to the most optimal, lean, scenario to illustrate difference with that of the status quo.

Figure 88 depicts that the transaction performance of medium complexity projects is comparable to that of low complexity projects. The best alternative to the status quo results in 1.85 times less costs at the same NVAW levels. The RailML models relatively decrease costs by a factor 1.28 (RailML best case) to 1.52 (RailML worst case). The worst case RailML model however again comes at the cost of a larger NVAW level in absolute terms.

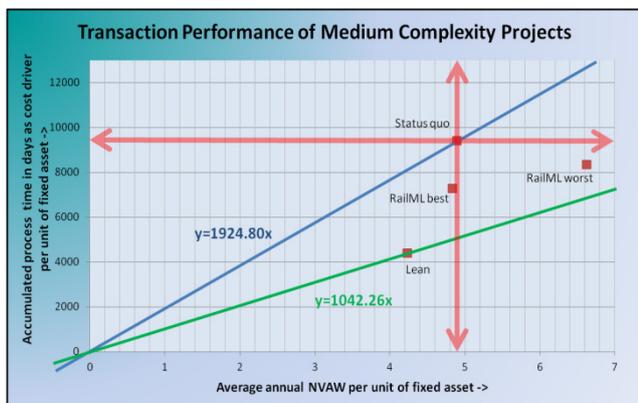


Figure 88: The transaction quadrant of medium complexity projects in which the status quo is compared to the RailML and lean process chain models. The graph depicts a line from the origin to the most optimal, lean, scenario to illustrate difference with that of the status quo.

Figure 89 shows different behavior from high complexity projects compared to low or medium complexity projects in terms of transaction performance. The lean model does not position itself in the favorable 4b quadrant (Figure 86) anymore. Therefore, the RailML model does require more NVAW when dealing with high complexity projects in order to achieve a better transaction performance. This result makes sense as high complexity projects cannot rely on past data. As a result, the interlocking area design requires more processes and allows for the possibility of an interlocking area design replication (section 5.2). This leads to a difference in transaction performance between the lean and RailML best case model. The lean model reduces costs by a factor 1.74 and RailML's best case model by 1.58. The RailML worst case scenario scores worse in absolute terms since the NVAW level goes up while not achieving a much better cost level. The relative difference is however better than the lean model with a cost factor difference of 2.09.

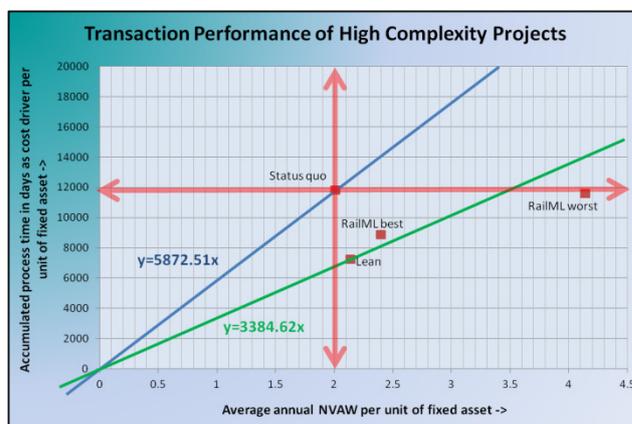


Figure 89: The transaction quadrant of high complexity projects in which the status quo is compared to the RailML and lean process chain models. The graph depicts a line from the origin to the most optimal, lean, scenario to illustrate difference with that of the status quo.

Table 20 concludes this section. In general, the lean performance of RailML is decent: the cost improvement covers quite a substantial portion of a lean chain.

Table 20: Conclusion of section 12.1 by scoring RailML's lean performance on a three point scale.

Performance metric	RailML's lean performance	Comment
Siemens' costs	+	RailML covers about a fifth of the lean cost reduction.
Siemens' productivity	-	RailML improves productivity but misses lean potential.
Siemens' value-time cycle	+	RailML covers about a quarter of the lean value-time cycle improvement.
Transaction cost efficiency	+/-	RailML shows a decent transaction performance at the cost of more NVAW in absolute terms.

5.3.2 The Differences in Ambiguity Figures between the Status Quo, RailML and Lean Models

The model allows to assess the models' ambiguity by the number of ambiguous processes, the size of NVAW and the tier rejection rate. The resource utilization rate could result from the model as well although the comparison would not provide usable information. The amount of resources follows a rule of thumb rather than real data (section 5.2). That allows for relative comparisons of indirect results like the productivity rates in Figure 84, but it does not allow for a straight interpretation in absolute terms. Especially not since the process and resource structure changes per model.

Figure 90 visualizes that the lean state deals with almost all ambiguous aspects in the model. The only ambiguity that remains arises from actors that suddenly desire to change the CRS while the FIS or RVTO already started. RailML most and foremost deals with ambiguous processes during RVTO, SWOD and EVP engineering like non machine readable data transfers and interpretation, interlocking area design from scratch and a multitude of validation/test processes. RailML however still leaves some validation processes, a.o. during RVTO and EVP engineering, and does have the means to improve the CRS/FIS' ambiguities that mostly arise from stakeholder interaction instead of data. With 13 main ambiguities as status quo, 7 when introducing RailML and 2 in an optimal lean model, RailML covers about the desired lean performance.

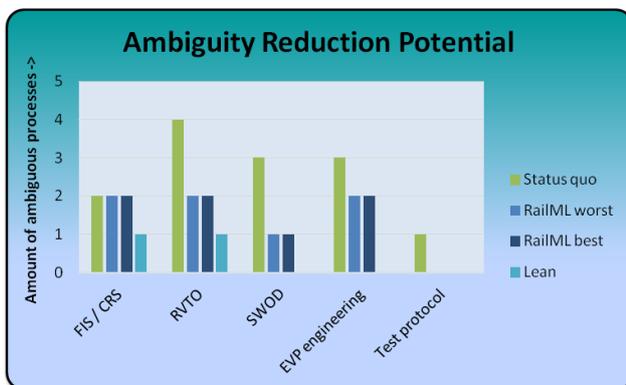


Figure 90: The amount of ambiguities as found in section 3.3 that each model deals with.

Figure 91 illustrates that RailML's potential to reduce the amount of replications approaches that of an optimal lean model. Each level of complexity achieves half to two thirds of the lean performance. Although the lean model only allows projects to replicate the EVP engineering, the best case RailML model reduces the probabilities of both the interlocking area design and EVP engineering (Table 16). This probably explains why especially the best case RailML scenario approaches the lean model so well.

Figure 92 shows that a best case RailML model can again cover about half of a lean transformation in terms of total NVAW flow. Total NVAW flow differs from average NVAW by accumulating the amount of projects and processes. In an unfortunate scenario however, RailML only covers about a tenth or may even worsen the status quo in the case of high complexity projects. A worsened total of

NVAW flow from high complexity projects results from a standard replication of the entire RVTO plus SWOD process whereas the status quo only demands a replication of the RVTO and with a small probability that of the SWOD as well. Only high complexity projects are affected since they still require to execute the VT-OBE and GIS map development due to the absence of base data (figure 68).

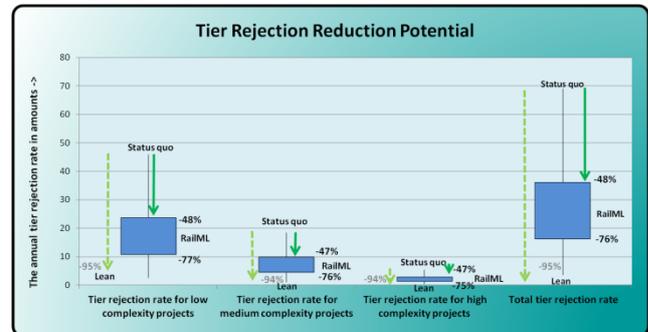


Figure 91: The amount of times that a model requires a project to replicate a certain process for each of the models. The dotted light green arrows indicate the lean performance improvement and the dark green arrows indicate RailML's performance improvement.

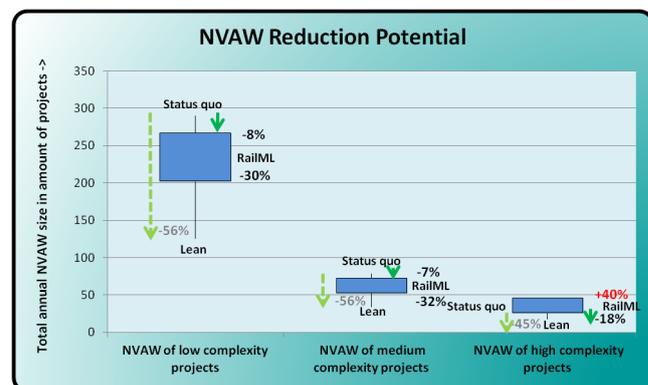


Figure 92: The change in accumulated NVAW over a year compared for each model. The dotted light green arrows indicate the lean performance improvement, the dark green arrows indicate RailML's performance improvement and the red arrows indicate RailML's process deterioration.

Table 21 concludes this section. In general, the lean performance of RailML is decent: the ambiguity improvement goes in the same direction as the lean one with the exception of high complexity projects' NVAW reduction.

Table 21: Conclusion of section 12.2 by scoring RailML's lean performance on a three point scale.

Performance metric	RailML's lean performance	Comment
Ambiguity reduction potential	+	RailML's amount of eliminated ambiguous processes is half the lean amount.
The tier rejection reduction potential	+	Even in a worst case RailML scenario, half lean's tier rejection rate improvement is meet.
NVAW reduction potential	-	The best RailML scenario is half the lean performance but large variety caused by the worst case scenario, with a possibility to worsen the status quo, question RailML's lean's performance.

5.3.3 The Differences in Time Figures between the Status Quo, RailML and Lean Models

The model assesses the time criterion by means of a comparison of lead times and cycle times; decomposed by value creating time and non value adding time / waiting time.

Figure 93 illustrates that in a worst case scenario, RailML would not radically improve the lead times of projects. In such a case, the chain that uses RailML only benefits from the eliminated FIS validation and, in the case of low and medium complexity projects, the SWOD. The high complexity case does not really eliminate the SWOD because of the remaining need to map the interlocking area from scratch. In the event that the probabilities of validation decrease, simulated by the best case RailML model, the lead time reductions prove more substantial although only best high complexity case achieves more than half the lean performance. The lean model benefits from additional elimination of processes; especially during the engineering stage.

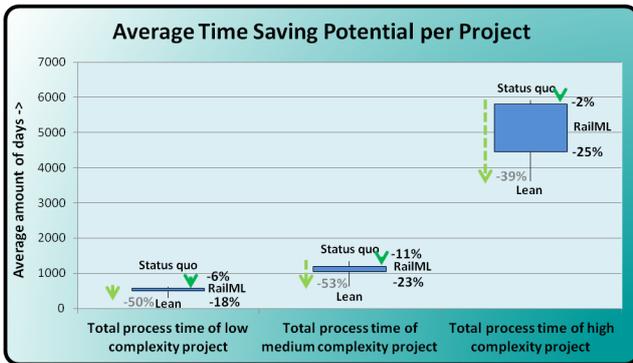


Figure 93: The reduction in average project lead time per type of complexity. The dotted light green arrows indicate the lean performance improvement and the dark green arrows indicate RailML's performance improvement.

Figure 94 in comparison to Figure 93 shows that the value added time per project decreases more than the non value added component in the case of low and medium complexity projects. The most likely explanation is that the elimination of processes led to a mix of processes which contain less value added time compared to non value added time. The elimination of RVTO processes for example, which has much value added process time and relatively low non value added process times, supports this reasoning. Since the RVTO processes mostly remain in place when design engineering high complexity projects, this would also explain why the high complexity projects do not show a bigger reduction of value added time compared to non value added time.

Figure 95 shows that RailML reduces the process times; test protocol/dry test and RVTO/SWOD can meet the lean performance. That the test protocol/dry test shows the same performance between the RailML and lean model makes sense because their structure is identical by eliminating the test protocol process. The RVTO/SWOD reduction that meets the lean performance most and foremost results from the low and medium complexity projects that eliminate quite some RVTO processes. The RailML model might even outpace the lean model since it operates in parallel instead of sequence. The gain however comes at

the cost of variability in the other direction as RailML might also lead to a deterioration of the status quo. The previous section indicated that the RailML and lean model demand at least one replication of the RVTO/SWOD instead of only at the RVTO. Therefore, the value and non value (Figure 96) added times will increase if the replication probabilities remain the same. Remarkable is that the non value adding times will deteriorate substantially more than the value adding times. Probably caused by the worst case scenario increase of NVAW as shown in Figure 96. Furthermore, the substantial non value added time improvements at FIS/CRS and test protocol/dry test consume just a small proportion of the total amount of time.

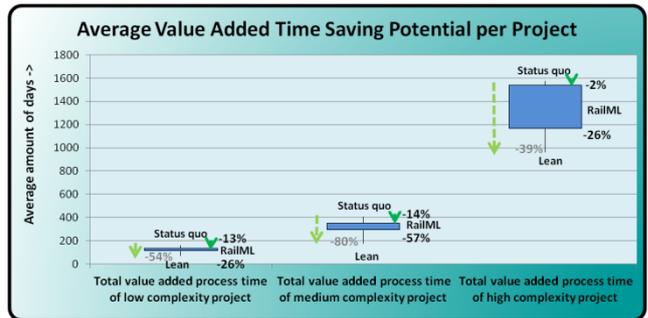


Figure 94: The reduction in average project value added lead time per type of complexity. The dotted light green arrows indicate the lean performance improvement and the dark green arrows indicate RailML's performance improvement.

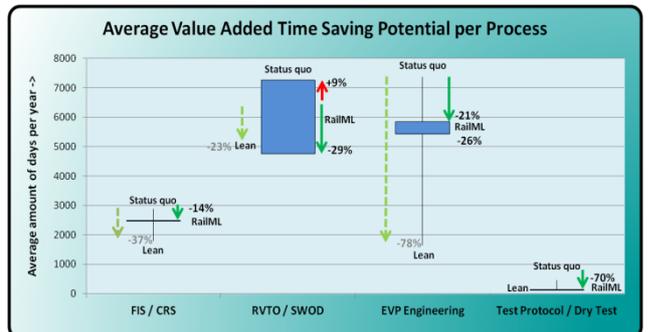


Figure 95: The reduction in average project value added cycle time per group of processes. The dotted light green arrows indicate the lean performance improvement, the dark green arrows indicate RailML's performance improvement and the red arrows indicate RailML's process deterioration.

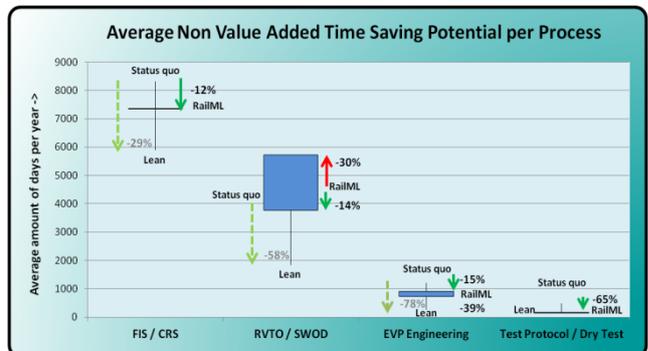


Figure 96: The reduction in average project non value added cycle time per group of processes. The dotted light green arrows indicate the lean performance improvement, the dark green arrows indicate RailML's performance improvement and the red arrows indicate RailML's process deterioration.

Table 22 concludes this section. In general, the lean performance of RailML is rather average: the time improvement goes in the same direction as the lean one but at a slow pace. Furthermore, some processes show a deteriorated time picture instead of an improved one.

Table 22: Conclusion of section 12.3 by scoring RailML's lean performance on a three point scale.

Performance metric	RailML's lean performance	Comment
Lead times	+-	Process times decrease well towards the lean model in an optimistic RailML scenario though barely in a worst case scenario.
Cycle times	+-	Process times reduce well in the direction of the lean model. RailML however deteriorates RVTO/SWOD on average.

5.3.4 The Differences in Interdependency Figures between the Status Quo, RailML and Lean Models

The System Level Coupling Index (SCLI), the amount of design rules, the amount of separate processes and combinations of complexity figures assess the amount of interdependencies criterion.

Figure 97 shows that the average SCLI decreases with the lean model but increases with the RailML model when compared to the status quo. This result makes sense as the SCLI depends on the number of separate processes. RailML eliminates some processes though it leaves the main process structure the same. Especially the parallel process during the RVTO/SWOD contributes to a high SCLI value since it contains four parallel processes instead of three which have relations to each other and the next processes. The SCLI can also be larger in practice due the introduction of RailML that allows for automation. A higher SCLI however does not correspond to the lean philosophy.

from the data it can exchange, the format it can validate and the linked IT tools that automatically produce output files or export files via RailML.

The number of rules are based on Prorail design guides (Prorail, 2010a, 2010b, 2010c, 2010d, 2012b, 2012c). The reduction of design rules in comparison to Table 8 depends on the ability of RailML to standardize and enforce rules. For example: an interlocking sketch needs to depict certain elements that someone needs to draw and RailML, or an associated tool, can enforce. The amount of design rules in the lean benchmark further results from changes in the process structure, e.g. that Prorail executes the current FIS instead of engineering agencies.

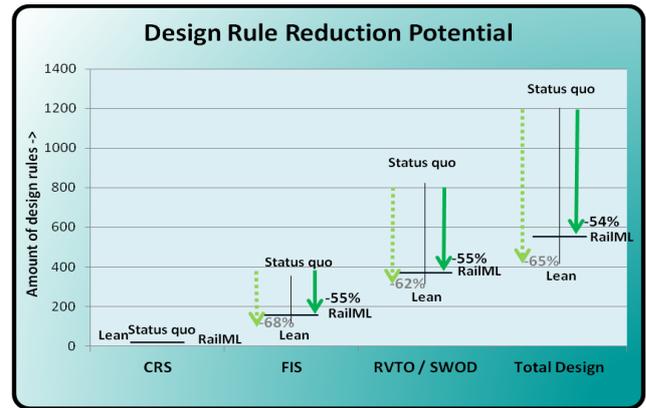


Figure 98: The number of design rules during the design stage.

Figure 99 shows that RailML reduces the amount of separate process steps from 20 to 16. The lean case enables further reduction to 14 process steps. The main disadvantage of RailML's new approach is the strong RVTO, called interlocking area design in the lean model, focus. This implies that the other processes strongly depend on this process. Furthermore, discovering errors becomes harder in one bundled process than small separate process entities. The status quo and especially lean model perform better with this respect.

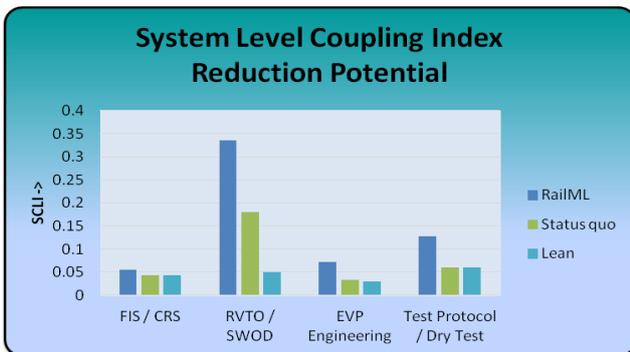


Figure 97: The average system coupling level index for each of the four models categorized for the main processes. The system coupling level index follows Equation 2. The number of modules results from the number of processes as shown per model in section 5.2.

Figure 98 leads to the conclusion that RailML can take a big share in the lean reduction of the amount of design rules. Most of RailML's reduction in design rules results

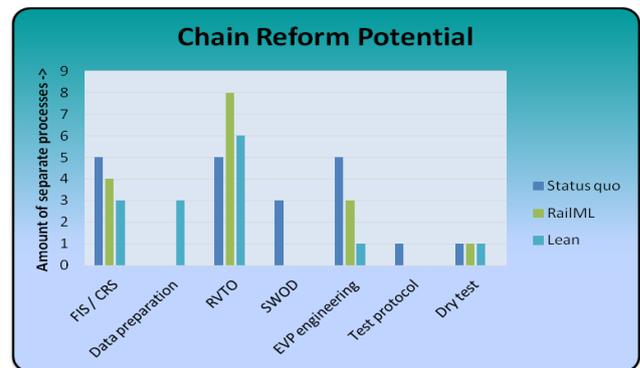


Figure 99: The amount of separate processes per group of processes. The diagram counts each parallel process as a separate process. The amount of processes follows from section 5.2.

Figure 100, Figure 101 and Figure 102 prove that there exists no clear relation between the SCLI measure of complexity and the total accumulated time as cost driver. This result implies that the earlier finding related to Figure 97, higher SCLI values for RailML, barely has an impact on the behavior of costs.

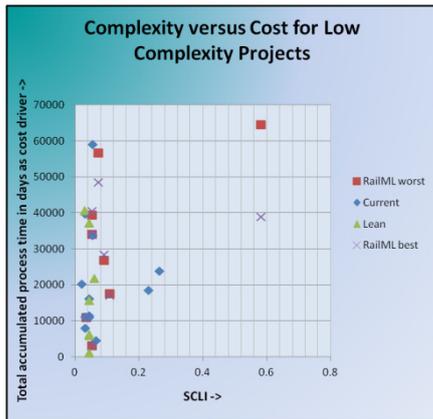


Figure 100: The SCLI with the associated accumulated process time concerning low complexity projects.

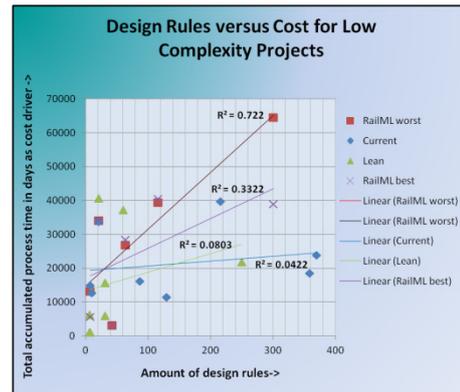


Figure 103: The amount of design rules with the associated accumulated process time concerning low complexity projects.

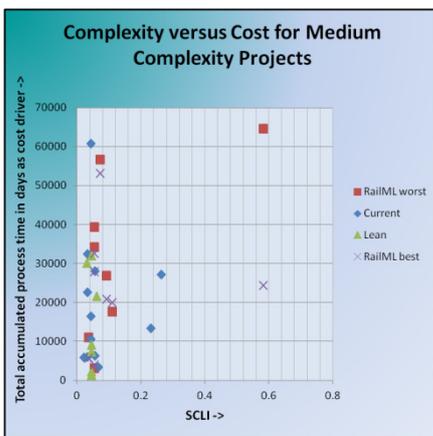


Figure 101: The SCLI with the associated accumulated process time concerning medium complexity projects.

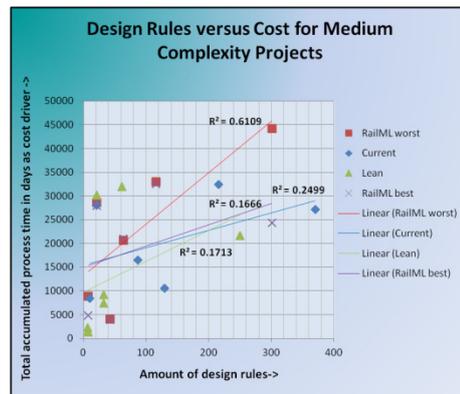


Figure 104: The amount of design rules with the associated accumulated process time concerning medium complexity projects.

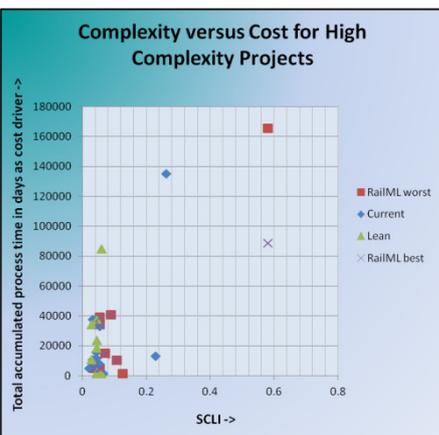


Figure 102: The SCLI with the associated accumulated process time concerning high complexity projects.

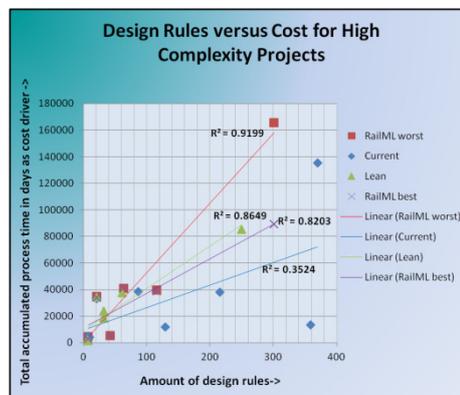


Figure 105: The amount of design rules with the associated accumulated process time concerning high complexity projects.

Figure 103, Figure 104 and Figure 105 show increasing relations between the amount of design rules and the total accumulated time per model. Although the few data points raise uncertainty, the general trend increases time with increasing design rules when taking all data points of one complexity type into account. Therefore, lowering the amount of design rules forms an important driver in the development of a lean transformation approach. Figure 106 furthermore shows that the amount of design rules does not have cross relation with the SCLI.

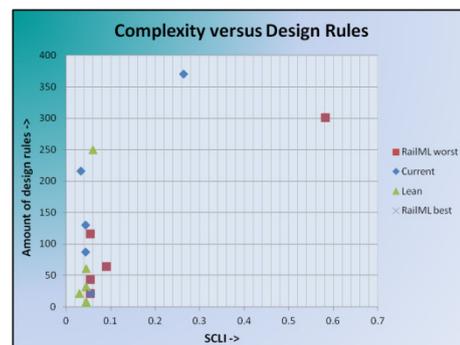


Figure 106: The SCLI values plotted against those the associated amount of design rules for each of the four models.

Table 23 concludes this section. In general, the lean performance of RailML is decent: the effects of the amount of interdependencies goes down.

Table 23: Conclusion of section 12.4 by scoring RailML's lean performance on a three point scale.

Performance metric	RailML's lean performance	Comment
SCLI	+-	RailML increases the SCLI though the consequences seem negligible.
Amount of design rules	+	RailML significantly reduces the amount of design rules. Furthermore, a lower amount of design rules decreases accumulated time.
Amount of separate processes	+-	RailML reduces the amount of separate processes but, as a result, puts a strong and undesired focus at the RVTO / interlocking area design.

5.4 Discussion of Results

This section discusses the research approach in the light of the study's goal, the results, the study limitations, data reliability, result generalization and so on. The discussion touches the two main topics of this thesis: the development of interlocking in RailML (subsection 5.4.1) and the lean transformation approach for an engineering design process (subsection 5.4.2). Subsection 5.4.3 discusses the topics that hit both topics.

5.4.1 Discussion of the RailML Interlocking Formalization Approach

The next discusses seven main RailML formalization challenges:

- Align RailML with a variety of railway businesses. This research focused on formalizing the interlocking for the Dutch railway network. RailML however intends to support several rail processes of various rail businesses; especially between industry and infrastructure managers when it concerns interlocking. The interlocking formalization of figure 57 contains two sub elements: signals and segments. The segment sub element would probably not cause any issues to completely describe movable track elements. The signaling structure however, might pose issues in different signaling regimes abroad. The Dutch signaling system ATB is quite concise. Therefore, the logic might turn out to become cumbersome or even impossible to use for different signaling systems. For instance a signaling system that works with approach signals instead of only main signals, could pose challenges. Besides the signaling structure, also a different interlocking process (consider figure 45) might pose difficulties. The interlocking formalization bases on SIMIS W's irreversible route interlocking before signal opening meth-

od. An interlocking system that irreversible interlocks routes after signal opening, might change the formalization too.

- Evaluate the implications of a bigger railway network on the interlocking formalization. The interlocking formalization relies on the Santpoort Noord interlocking area as best practice. In combination with RailML group's current work on the interlocking schema and expert opinions, most areas in the formalization are covered. Some exceptions that need a formalization in the interlocking schema could however arise when investigating different interlocking areas. Furthermore, complications could arise when formalizing a larger part of the (Dutch) network.
- RailML misses some essential data. Section 4.2 identified six shortcomings for the purposes of interlocking. RailML should at least encompass three of them (level crossing characteristics, signal types and track section characteristics) in order to prevent a substantial performance reduction compared to section 5.3. Two of which (level crossing characteristics and signal types) might cause difficulties to develop since they are railway specific. The RailML organization could for instance solve this issue by introducing an XML namespace per railway.
- RailML has only few times been applied in reality. Most of those times, RailML suited for the purposes of simulation. In that particular field, RailML mostly took care of depicting the rail infrastructure in the model. As far as known to the author, RailML only played a role in a few interlocking engineering design projects (Hürlimann & Krauss, 2003). Those projects however focused on train protection systems. Therefore, especially with the new RailML interlocking schema, data exchange issues can arise that require the RailML (interlocking) schema to change. Such issues would typically arise in the data transfers between an existing database and RailML.
- RailML needs to remain standardized. In order for RailML to achieve full potential in interlocking engineering design, the RailML files need to use the same schema. Especially for the data transfer between infrastructure managers and industrial parties like Siemens, a standardized RailML file is of vital importance. Section 5.1 makes clear that the RailML interlocking formalization allows to eliminate the data preparation and TOS engineering. In addition, the potential opens to eliminate EVP conversion and EVP test processes in the long term too. For reasons of design validity, process elimination only works when standardizing the process and the consequent processed data. In the case that RailML schemas differ from time to time, the process elimination potential vanishes.
- ETCS will change the RailML schema. Section 4.3 showed the implications of ETCS on the RailML interlocking formalization. In addition, the infrastructure sub schema of RailML needs schema or element representation alterations. The inclusion of Cartesian coordinates (x, y, z) is one of the additional necessities for ETCS and currently

not represented in the RailML infrastructure sub element. Whether RailML catalyzes the development of ETCS by simplifying its data engineering process, thus largely depends on the infrastructure sub element and not the interlocking formalization.

- RailML puts limits to the inclusion of data in the engineering design chain. Although an XML structure can theoretically include several kinds of data (after all, websites base themselves on XML), one should trade-off the benefits of data concentration versus the ease of data processing. A SVA report could be formatted to RailML, although a tool needs to convert it back for an interlocking engineer to understand it. This sequence does not enrich the data and only raises the probability of inconsistencies and interpretation mistakes.

5.4.2 Discussion of the Lean Process Transformation Approach

The next discusses nine main lean process transformation challenges:

- The expectations of RailML as lean transformer do not hold. Very strictly seen, the application of just RailML would barely lead to process improvements. This follows a discussion point in the next section: without a clear data source, RailML can only provide some performance improvements at the transfer from infrastructure manager to industrial engineering party.
- There seem limitations to apply the lean philosophy in an engineering design system. The high complexity projects in the lean model perform considerably worse compared to low and medium complexity projects in transaction costs and time savings: two of the most vital aspects considered in a lean transformation. In a way, this finding makes sense. A lean design philosophy strongly focuses on standardization and a modular processing approach to enforce project flow. In the first place, such an approach could pose issues in a design process because each design is unique. Furthermore, in the case of high complexity projects, one still needs to design an interlocking area from scratch which barely leaves room to fundamentally change the process.
- One should view the lean model as a hypothetical, though realizable scenario. Section 5.1 identifies how an interlocking engineering design chain can overcome most 'waste' factors. The lean model structure does not rely on a precise description of the means, like RailML. The lean model mainly functions as a way to benchmark the lean performance of the RailML tool.
- Performance figures can only be interpreted with certainty in a relative way. A project management database misses in the chain. The design part of the simulation model relies on process characteristic estimations from experts. This research did not have sufficient time (the lead time of a low complexity project could easily take two years) to study each process of a project; let alone of different complexity types. The performance figures in the engineering stage of the chain rely on four project offers executed by Siemens. Although more comprehensive than the figures of Prorail, analysis of variable relations needs a critical reflection for review in absolute terms. The analyses do however align with expectations of various Siemens engineers. In addition, the figures of Prorail are also validated by five involved employees; before and after the simulation process. Therefore, the values present a fair picture of the most critical processes to discover 'waste' and, with careful reasoning, describe the relative performance when altering the status quo model. Since this study aims to assess the relative performance of RailML versus the status quo and a lean approach, approaching the Pareto principle is sufficient.
- The reliability of the model mostly depends on the model running time and the amount of replications as described in appendix L. The half width of each measurement determines the confidence interval. The half width of most individual process measurements, including the total process times and costs, typically ranges around 1% to 2% or lower. The half width of accumulated process characteristics may hit the 5%. This implies a confidence interval of 10% which one might consider large. Appendix L did also find that reducing that half width figure would require a substantial amount of additional replications. The author considered the additional time per simulation run too large for the reduction of values that are barely used in analyses.
- The lean design transformation methodology proves decent, though requires a substantial amount of data. The previous considerations made clear that half a year of study provided sufficient data to map an engineering design process of 20 main process activities. The main issue in value chains that investigate the lean philosophy however, whether it concerns a design or an engineering processes, remains the lack of chain coordination. As a consequence, parties barely monitor performance indicators of interest for the chain. An analyst could relatively easily measure process characteristics in a production, although this does not apply for a design process. Therefore, the lean design transformation methodology should especially improve in that area.
- ARENA allows for the modeling of engineering design processes, although puts limits on resource dynamics. Resources in engineering design processes mostly concern employees. ARENA cannot represent the iterative nature in which multiple employees work together and process work gets reproduced really well. In other words: on a macro scale, ARENA allows for engineering design process modeling, on a micro scale, other modeling methods need to be found / developed.
- Competitors' approach to engineer an interlocking system. Siemens' engineering process structure determines the way interlocking systems get engineered in this study. The SIMIS W interlocking focuses on a node-branch structure to find the relevant rail elements to interlock a route. Experts from both Siemens and Prorail expect that interlocking systems from competitors work

in the same way. In that case, the process proposal and related performance merely apply for those parties as well.

- Infrastructure managers' approach to design an interlocking system. Besides differences between engineering parties, infrastructure managers may design the interlocking system differently too. Although of less interest to the interlocking system itself, other infrastructure managers might pose some interesting ideas to structure the design process.

5.4.3 Discussion of the Combined Lean RailML Approach

The next discusses four main lean RailML challenges:

- Interlocking safety. Chapter 2 clarified that infrastructure managers highly value a (near) failsafe interlocking design. This raises the questions whether the new RailML process structure can also ensure such high quality levels, and if yes, how to ensure them? The RailML process model in Figure 72 only eliminates the CRS/FIS validation stage since RailML allows to check whether the developed data format by an XSD. The data preparation's data validation sub process does, in theory, not add value anymore because the status quo's SWOD process gets validated in the RailML model. Furthermore, RailML allows for the possibility to continuously validate the changes made to the existing interlocking area files (low and medium complexity projects only). Therefore, safety levels probably remain the same or improve. A design process, compared to a production process, however does not allow to easily compute a confidence interval or mathematical statement for proof of safety. In order to test whether the safety statement holds, requires additional research that is outside the scope of this project.
- Visual representation for verification and validation purposes. Section 5.1 stated that RailML should not turn the engineering design process into a black box. The availability to visualize and manually validate data remains very important because of mainly two reasons. One, involved employees state that they have a hard time to check database structured data. Two, it is an Utopia to assume that RailML will eventually generate an interlocking production file with the 'push of a button'. RailML cannot process all kinds of data, designers and engineers will always confront data version conflicts and they need to deal with unique characteristics per interlocking area.
- Where can the chain, and especially the infrastructure manager, find its source data? RailML, being a data exchange tool, makes no sense without a main data source. Prorail works on the development of a GIS that allows for that possibility. This research therefore assumes the existence of a GIS as data source. It remains however to be seen whether the GIS can really function well as main data source in reality.
- Limit the amount of tools; especially during the interlocking design process. Prorail and partners already use a wide variety of IT tools during the interlocking engineer-

ing design process. RailML not only adds a tool by being a tool, conversion processes from and to RailML with add a bunch of new IT tools too. The infrastructure manager should guard itself against an abundance of tools. After all, every tool requires maintenance and every tool imposes the additional probability of making errors.

6. Conclusion & Recommendations

This chapter first concludes the individual research questions (section 6.1) to derive an answer to the main research question: *“to which extent can RailML potentially enable a leaner engineering design process of rail interlocking systems?”* (section 6.2). Besides, section 6.3 gives recommendations for future strategies and research.

6.1 Conclusions on Sub Research Questions

The first sub research question addresses the methodology to transform an engineering design chain into a lean one. This sub research question aims to provide a guideline for achieving a lean engineering design process structure that allows to benchmark the lean performance of RailML. In addition, this sub research question aims to find whether the lean theory applies for engineering design processes as well, since practitioners (currently) apply the lean theory almost only to production processes.

The developed lean engineering design methodology bases itself on a system analysis framework and its associated representation of a system diagram. A system diagram structures system mechanisms in its simplest form as input, conversion, external influences and output. Input concerns measures that parties in the interlocking engineering design chain can take to make their processes lean. Various literature studies that investigate lean transformations in the production of transport systems and one design process, indicate five high potential transformation directions for the interlocking engineering design chain: ‘waste’ prioritization and elimination, the introduction of open source tools, standardization of processes, a modular process approach and information sharing across the chain. The conversion entails how the transformation directions influence the status quo. Literature analysis leads to a seven step sequential transformation methodology to (1) define customers’ requirements, (2) conceptualize the current state of the system, (3) identify its “waste”, (4) define the lean state, (5) engineer the proposed system (RailML in this case), (6) model and measure the effects of (1), (4) and (5), (7) evaluate and compare the results and (8) design the final state of the proposed system that mitigates undesired effects. The methodology needs to account for external influences imposed by the presence of multiple parties that use varying engineering design procedures. The outcome of the system concerns measurement of the new lean / RailML approach to a set of performance indicators. These performance indicators contain metrics that align with the lean definition of a value added focus, ‘waste’ elimination, continuous process flow, customer pull and perfection. The main goal of the methodology is to express process complexity in standardized requirements so that IT tools have high potential to mitigate a lot of ‘waste’.

The methodology proved effective to design and test a (conceptual) lean process structure. This lean process structure proved useful as a benchmark that allows to

asses the leanness of transformation strategies. Furthermore, the lean theory aligns very well with the ‘waste’ found in the engineering design chain. The methodology’s main disadvantage comes from its large demand for data. Engineering design processes usually lack data which one cannot easily measure as well. This might especially pose issues when an engineering design structure gets more complex.

The second research question addresses what a lean interlocking engineering design process encompasses. The methodology gets applied to state the status quo, find “waste” and design a process structure that deals with most of that “waste” in a realizable way.

The design stage of the chain covers definition of the interlocking requirements (FIS), specification of the interlocking area (RVTO) and design of the interlocking engineering initiation files. Engineering agencies take care of the design; the infrastructure manager controls the process. Engineers from an industrial party like Siemens then prepare the data from non machine readable maps to their systems, develop the interlocking IT (TOS and EVP) and test it on a test machine (EVP conversion and test). Before the interlocking system gets produced, the infrastructure manager tests whether industry’s interlocking system behaves as expected.

A comparison between this process structure and the interlocking requirements allows to mark process characteristics as “waste” or not. Every interlocking project contains general system (a.o. RAMS), general logic (a.o. simple switch), signaling system specific (a.o. ATB-EG with ETCS) and hardware related requirements (relay or electronic). Besides standard requirements, track topology and station area requirements depend on the interlocking area and opinion of the involved experts.

Data analysis proves that the design stage mostly suffers from considerable non value added work-in-progress, high work complexity measured in interdependencies and design rules, many (non machine readable) data transfers and involved parties, ambiguous interlocking area design from scratch during the RVTO/SWOD and fluctuating client requirements. The engineering stage mostly deals with cost variability, considerable fixed costs, an ambiguous data preparation, ambiguous amount of total validation processes in the chain and low productivity rates. In general, the chain benefits from a shorter lead time as data analysis proves it to be the most important cost driver.

The lean state or benchmark mitigates that ‘waste’ by narrowing down the structure to five main processes: project definition, data preparation, interlocking area design, interlocking engineering and a dry test. Ideally, the infrastructure manager executes the first two and the last of those processes, an engineering agency executes the data preparation process and industry engineers the interlocking IT. The lean state can only mitigate more processes with a reduction in variable requirements. This could for

instance occur when interlocking areas of stations become more standardized.

Experts at Prorail estimated the main time characteristics of the process. Siemens delivered the specification of four interlocking offers. The limited amount of data cases also limited the application of statistical analyses. The study therefore mostly applied descriptive statistics to find 'waste'. As a result, the analyses could have missed some 'waste'. In addition, the interlocking engineering design processes could differ depending on the infrastructure manager and industrial engineering party. The nature of the data therefore allows well for relative performance comparison though should be assessed critically in an absolute way.

The third research question assesses how RailML models a representative Dutch interlocking area, Santpoort Noord, for the purposes of interlocking engineering design. The research question aims to develop one vital shortcoming of RailML: the interlocking schema. Three interlocking formalization studies provide the basis for that schema. One of these studies does not use RailML but a database. The other two propose a RailML formalization; one from the perspective of the interlocking hardware and the other from interlockable routes. This study bases the interlocking formalization from the perspective of a train requesting a route because of two reasons. First, eight gaps in RailML v2.2 to engineer an interlocking system, align quite strongly to the Dutch route interlocking process. These gaps include (1) active or passive route indications, (2) the route activation points, (3) the status of detection, (4) the status of movable track elements, (5) the status of level crossings, (6) rail element relations for flank protection, (7) route preferences and (8) signal aspect dependencies. Second, RailML group's most recent attempt got close by using a route based approach. The result however, contains some redundant data, cannot capture all aspect relations and has an opaque structure; especially concerning signal aspect dependencies.

A new signal based interlocking formalization resulted in a concise formalization based on combinations of only entrance speed profiles, signal aspects and target signals per signal. A combination of the route based and signal based approaches resulted in a complete, concise and RailML group coherent interlocking formalization that allows to capture all necessary rail elements for (Siemens') interlocking engineering. The structure contains two main sub elements: signals and segments. Signals encompasses the signal aspect dependencies. The segments element contains the rail elements to check on the route, the rail elements to check on the flanks and the priority route to set. The interlocking formalization holds for Santpoort Noord, although different interlocking areas, interlocking processes and signaling systems might pose additional formalization demands. Furthermore, the inclusion of a larger network than just one station area might provide issues as well. Besides, RailML lacks the experience from a best practice in an engineering design process.

The fourth research question investigates how RailML changes the engineering design process structure on the basis of the newly defined interlocking formalization. The

research question aims to derive a realizable process structure of the status quo with RailML. RailML improves the interlocking design stage by transfers of machine readable data, a less cumbersome and more accurate interlocking strategy development during CRS/FIS, elimination of OBE and OS map development from scratch with modification projects and automates test protocol development. As a(n) (indirect) result, the less complex processes allow the FIS to merge with the CRS and the RVTO to merge with the SWOD. Furthermore, the FIS validation, RVTO report and test development protocol processes become superfluous. At the engineering stage, RailML eliminates the data preparation process because only an insignificant task remains having a machine readable exchanged OBE and OS. Furthermore, RailML also eliminates the TOS engineering process since the formalized signal aspect dependencies largely correspond to the TOS.

In respect to a completely lean transformation, RailML lacks the ability to tackle various first out of three priority types of 'waste' during the design stage. Second, RailML introduces a new tool in an already tool rich chain. Third, RailML does not prove very conducive to transfer every single type of file. Fourth, RailML barely tackles the considerable fixed costs associated to accounts as project management and testing hardware.

The described chain with RailML assumes the introduction of an Enterprise Application Integration system (e.g. Geographic Information System or similar) as the main data source in the chain. This assumption forms an essential part of the RailML strategy and RailML cannot exploit its potential when partners in the chain do not know where to derive and/or store their data. In addition, RailML v2.2 misses some essential data for the engineering design of interlocking systems that does not belong to the interlocking formalization. Within this respect, RailML v2.2 might also require significant changes for the introduction of ETCS level 3. A scenario exists that the RBC controls the interlocking and questions the existence of a separate interlocking formalization. A last consideration comprises a trade-off between data concentration versus the ease of data processing. As a result, parties might consider to directly align their internal data and apply RailML for party to party exchanges only.

The fifth research question measures the improved performance of RailML with the status quo and investigates how close it gets to the lean process structure from research question 2. An ARENA simulation model assesses the cost, ambiguous process, throughput time and process interdependency performance indicators for the status quo, RailML and lean scenarios. In terms of cost, RailML reduces Siemens' costs between 17% to 27% depending on the level of complexity. This performance achieves 21% to 34% of the lean performance. In terms of productivity, RailML achieves a lean performance of 6.5% in a worst case to 12% in a best case scenario. Transaction cost analysis shows that RailML scores relatively better than the status quo on transaction costs although it comes at the cost of larger non value added work in absolute terms where a full lean approach would not for low and medium complexity projects. In terms of ambiguous processes, RailML reduces the tier rejection rate by 47% to 77% which

covers 49% to 81% of a completely lean transformation. RailML however does not reduce non value added work that well with a wide range from -40% (an increase in NVAW) to 32%. The time criterion per project shows that RailML reduces the lead time between 2% and 25% which covers 2% to 47% of a lean state. The time performance per process (cycle times) generally shows a decent performance of RailML versus lean with exception of the RVTO/SWOD. In terms of process interdependencies, RailML substantially reduces the amount of design rules per design process (55%) with exception of the CRS and approaches the lean performance (85% lean). The amount of separable processes reduces from 20 to 16 in the RailML situation and reduces further to 14 in the lean structure. ARENA proves a decent tool to evaluate the differences in lean design alternatives' performance. Especially when one wants to investigate the effects of an interarrival process and process / replication variability. Furthermore, ARENA greatly contributes to gaining process insight. The model also generally attains a high level of reliability with a common confidence interval of 2% to 4%. ARENA does however not have the ability to reflect engineering design's strong iterative and interacting nature of employees on a process.

6.2 Conclusion on Main Research Question

This section provides an answer to the main research question:

"To which extent can RailML potentially enable a leaner engineering design process of rail interlocking systems?"

The answer to this question contains a general and an academic part.

In general, RailML can improve the interlocking engineering design chain by mainly decreasing process times, costs and complexity measured in design rules, tier rejection rates and ambiguous processes. Figure 107 shows that the degree to which this happens, covers 43% of a full lean improvement when considering 11 out of 12 metrics for a medium complexity interlocking area like Santpoort Noord. The System Level Coupling Index is left out as it did not provide a clear relation with time as the main cost driver. Low complexity projects show a similar performance improvement as medium complexity projects. High complexity projects experience more variety, a weaker transaction cost effect, a shorter value added time reduction and a lower productivity improvement, while the total size of non value added high complexity work could become worse than the status quo.

The lean benchmark shows that there exists potential to further improve (1) productivity, (2) non value added work, (3) cycle times per process, (4) lean transaction costs in absolute terms and (5) the value-time effect. The effectiveness of RailML depends on a lot of factors. In the first place, RailML can only achieve its potential with the clear introduction of a main data source for the entire chain like a GIS. Second, RailML should not lead to a black box process chain. Third, RailML should not lead to an increase in the amount of IT tools in the chain. Fourth, the safety case should be guaranteed. Fifth, RailML needs to

align with various industrial parties and infrastructure managers other than the Dutch. Sixth, the standard RailML needs adjustments and consider ETCS level 3. Seventh, RailML needs a best practice: it proves much potential in the executed case study, though an engineering design chain has never put it to practice.

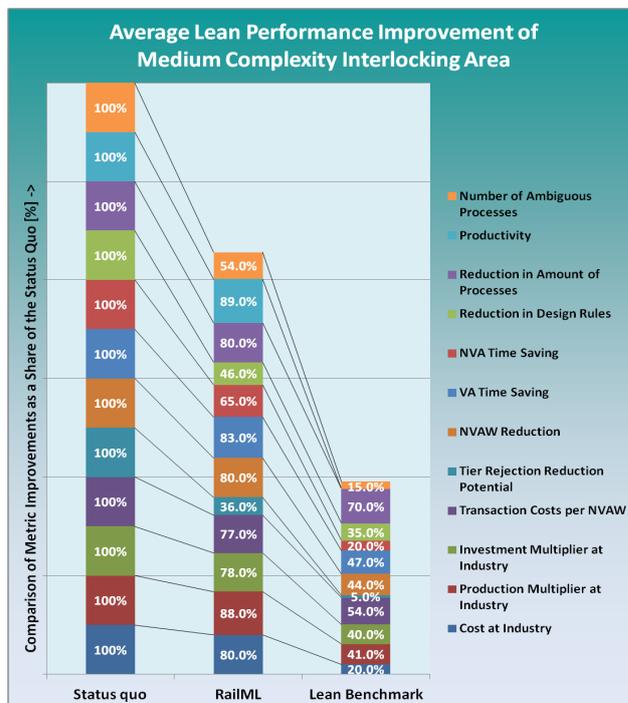


Figure 107: The performance improvement of an interlocking engineering design chain with RailML and the lean benchmark compared to the status quo. Productivity is interpreted inversely: the lower the better where lean equals 0%, status quo 100% and RailML the relative importance within this range.

From an academic standpoint, the RailML interlocking formalization makes a considerable leap forward. The two existing proposals do not address several for interlocking engineering relevant elements and base on an outdated version of RailML. The RailML interlocking group's progress is still quite premature, misses some essential data and knows a comprehensive, sometimes redundant, structure. Furthermore, questions remain on which object relations to include and how to represent them concisely. The proposed interlocking formalization in this study excels in its development approach from a real interlocking area on the basis of real and up-to-date engineering design files. Furthermore, cooperation with an industrial interlocking engineering party learnt how to present that data such, that the interlocking engineer gets precisely sufficient data in a way that promotes engineering. The result formalizes the interlocking RailML in a very concise way mainly because of grouping the objects from the standpoint of a train requesting a route, a new signal aspect dependencies approach and the bare minimum addition of other elements. In fact, at the RailML interlocking conference in Paris of September 2013, the RailML group showed much interest in the approach. Each participant of various infrastructure managers and industrial parties will investigate the interlocking approach for their signaling and/or engi-

near system to derive a widely accepted interlocking formalization in the near future.

The second academic contribution of this research concerns the development of a lean transformation methodology for engineering design processes. The lean management philosophy led to a wide range of transformation approaches that focus on production systems; an even smaller share focused on transportation related businesses. This raised the questions how to transform an engineering design chain into a lean one and whether lean works in an engineering design process. The developed lean engineering design transformation methodology combines lean thinking, lean best practices, system analysis, conceptual modeling techniques and analytical modeling techniques. The methodology leads to a scientific benchmark of a strategy's, e.g. RailML, lean performance. The lean tool's (e.g. RailML) lean performance mainly depends on its ability to translate complexity into standardized 'building blocks' of requirements. The study raised three considerations with respect to a lean engineering design process. First, the considered techniques can model an engineering design process on a macro level, not a micro level of project iterations. Second, the methodology requires a substantial amount of data that might be hard to get in a design process. Third, the degree in which engineering design processes can become lean, appears to have its limits. The data in this research suggest that the more complex a project, the more unique the aspects, the more required process steps, the more difficult to achieve continuous flow and the less lean.

6.3 Recommendations

This section defines recommendations from a business perspective and an academic perspective. The latter distinguishes RailML and the lean engineering design methodology.

Business recommendations for Siemens, Prorail and other related rail parties:

- Continue with the investigation and development of RailML based on the Dutch interlocking engineering design chain. RailML has the potential to considerably improve performance and the lean benchmark showed that this potential may grow even further. An important task now relies in expanding the RailML and investigating the influence of a bigger network and (possibly) new interlocking situations on the RailML structure. In addition, start the development of a RailML with ETCS level 3, preferably based on a real interlocking area.
- Make a cost-benefit analysis which mainly compares the costs of RailML implementation with the performance improvement. The results provide reason to implement RailML at a large scale or not.
- Develop and integrate an IT constellation strategy with RailML for the entire engineering design chain. RailML's potential depends on this constellation and especially three aspects: a clear and complete data source, e.g. GIS,

a minimization of IT tools and the prevention of a black box IT system, e.g. by allowing for visualizations.

- Start a pilot project with RailML. A small scale interlocking project may reveal essential aspects to the formalization that do not arise as quick when modeling. Furthermore, an equal interlocking project that runs parallel, can benchmark the performance of the RailML approach.
- Investigate the use of RailML in other railway signaling development projects. RailML shows beneficial for interlocking engineering design, though it may also prove beneficial in other areas such as safety system development. Furthermore, synergy opportunities might arise.
- Investigate co-development of other tools or process structures that mitigate 'waste' that RailML cannot. The lean process structure identified areas that RailML does not improve as well as the lean approach. A combined approach of other tools and structures may lead to a lean approach after all. Suggestions include modular processing of projects, i.e. teams that focus on projects of a certain level of complexity, the introduction of Enterprise Application Integration (EAI) to achieve a shared work stream in the cloud across partners in the chain and project management tools and/or approaches to improve the client requirement specification.
- Standardize interlocking requirements as much as possible. Currently, station and track topology requirements contain variability. Introducing identical station areas and/or fewer switches in the network could for instance substantially contribute to standardize the process and allow for more automation.

Academic recommendations concerning the RailML interlocking formalization:

- Develop the RailML v2.2 to mitigate the data gaps for interlocking engineering design.
- Conduct research to find out if and how the formalization can account for signaling systems outside the Netherlands. This research only accounts for signaling system that are in place or planned for in the Netherlands. The situation could arise that a different signaling system does not fit in the current formalization. The formalization then needs to change.
- Conduct research in order to find out whether different infrastructure managers or industrial parties work substantially different and require changes to the RailML.
- Conduct research to investigate the precise effects of RailML on the safety case. This research conservatively eliminated process tasks that RailML only could fulfill without doubt. Therefore, the design safety should not go down but only increase with the added validation options of RailML and decreased amount of non machine readable data transfer. Research however needs to quantify the effects.

- The RailML group should ensure the RailML standard. Especially for the data transfer between infrastructure manager and industry, a standard is of high importance to success. When an industrial party receives different RailML files, whether that comes from the structure or the way elements are interpreted, their engineering process cannot improve and change as described in this research.

Academic recommendations concerning the lean engineering design methodology:

- Apply the lean transformation methodology to other engineering design chains. This research may improve the reliability of the methodology and lead to improvements.
- Conduct research with more elaborated process data to investigate whether the 'waste' conclusions change. When significant differences arise, this may question the general applicability of the methodology.
- Conduct research to find modeling techniques that allow better for design process interactions. As stated in the conclusion, the methodology and especially ARENA may only model an engineering design process on a macro scale. The question arises whether there exist better alternatives.
- Conduct research to investigate the finding that high complexity engineering design processes do not allow well for a lean approach. The data in this report seems to support this statement, but it's insufficient that draw a strong conclusion.
- Investigate how the added performance improvement of the lean benchmark in addition to RailML can be achieved. The study pointed out that measures should be taken that allow for a modular engineering design process, continuous flow of processes, larger degree of standardized 'building blocks', improved project management and perfectionism. Possible solution directions include the elimination of more validation stages, less variety in interlocking areas, elimination of parties and/or consolidation of processes per party.

Appendix A - RailML Interlocking UML by RailML group

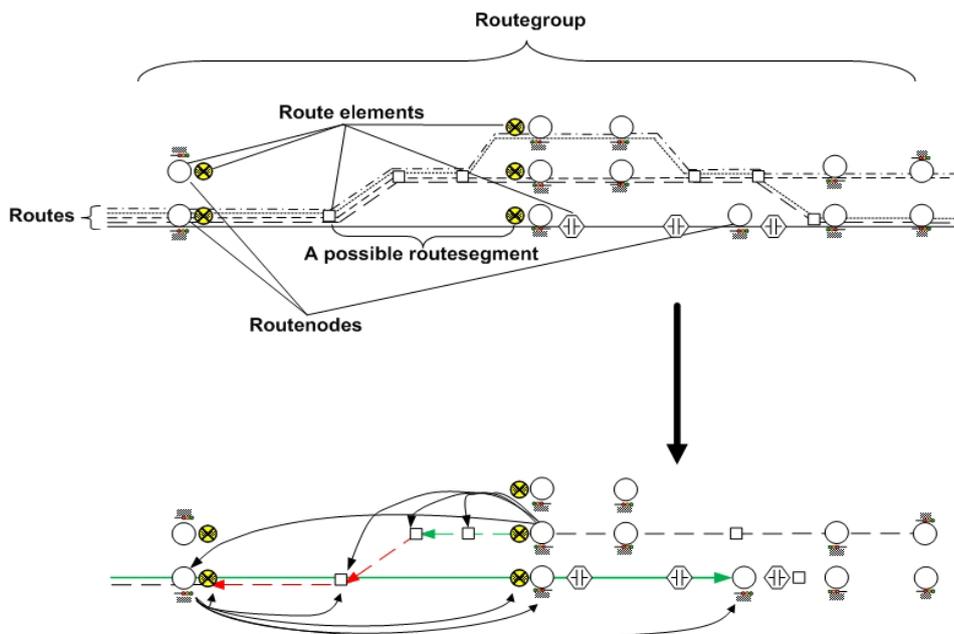
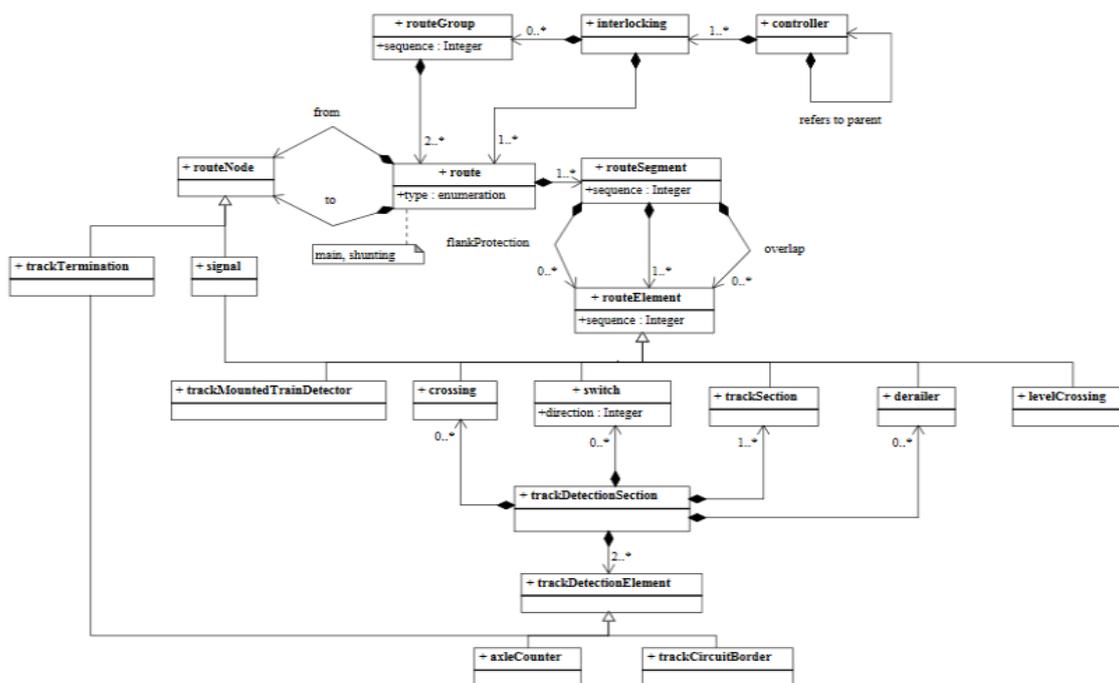


Figure 108: The UML shows the interlocking UML as agreed upon by the RailML group. The lower illustration shows the visual meaning of the object relations. A routegroup contains two or more different routes that a train can request to drive on. A route is defined as a realizable link between two routenodes. The Dutch Railway network defines a route as long as a block section due to which a node is always a signal (van der Meij et al., 2013). Every route then contains two or more rail elements. Rail elements in the visualization include signals (circles), level crossings (yellow circles), switches (squares) and detection (hexagons). The UML defines a link between two rail elements as a routesegment.

Appendix B - System Analysis

This chapter covers a more detailed look at the fundamental issue elaborated in 1.2 by identifying the questions that need to be researched. This chapter starts with a goal analysis, then an actor analysis, followed by an external factor analysis and closed by an impact analysis.

Goal Analysis

The goal analysis uses a goal-means diagram, supply chain overview and means-end diagram to get a clear picture of the scope and derive a set of performance-indicators.

Goal-Means Diagram

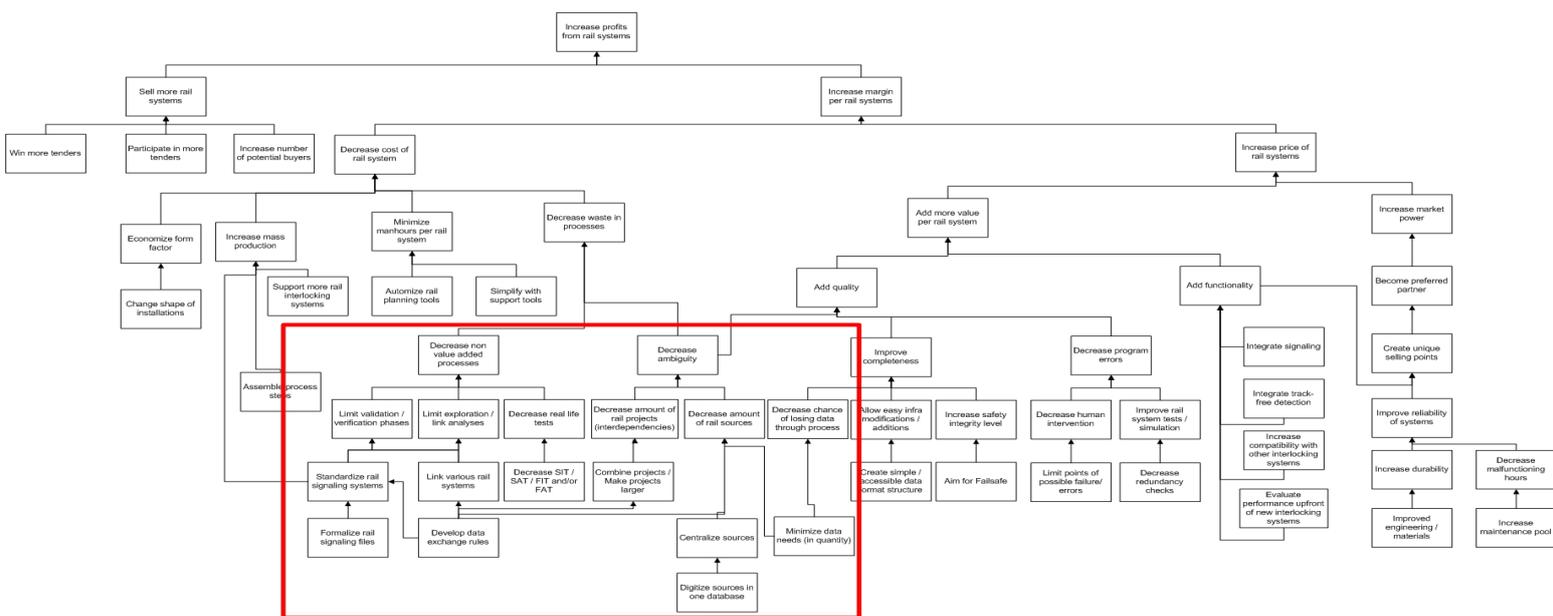


Figure 109: Goal-means diagram illustrating the main focus area of this research with the red box.

Figure 109 shows the goal-mean diagram that indicates the 'how' relations from top-goal to bottom-measure of Siemens' Rail Automation division, specified into rail signaling systems. Most important is to increase profits, which is a derivative of costs and income and can be decomposed further down.

The main takeaway from Figure 109 is the focus area. The two top goals of the focus group are to decrease non-value added activities and ambiguity during the development of signaling systems. This stretches as far as measures to standardize more processes and core data structure instead of tailored process structures for each project.

The issue gap described is explicitly not about adding functionality, expanding market share and/or marketing; it is purely about doing the current development process better, faster and cheaper. They could however become one of the ultimate indirect goals for Siemens. Furthermore, measures to decrease errors and tweak the available data are not a goal in itself as well. It would be nice to hit

two birds with one stone, although it seems better for a separate ICT / math study.

Means End

The means end diagram identifies possible ways to measure the success of the final project result. The scope determines the eventual selection of kpis. Some remark scan however already be given. First, cost and time are closely related. Major part of the costs is caused by the number of FTE involved in the process and the number of involved employees also causes throughput time to rise significantly. It thus makes sense to combine both as one kpi. Second, the computer calculation gain is probably something which will show its effect on large track sections. This study will not investigate a large section due to which the role of such a kpi is debatable. Third, conclusions on the amount of data errors requires a benchmark. It is yet questionable if enough information is available to derive a proper benchmark.

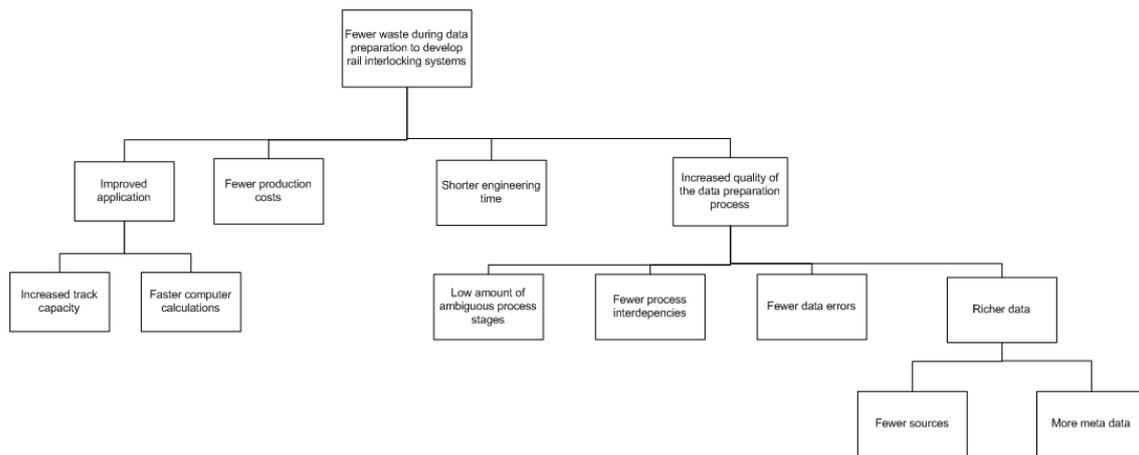


Figure 110: Project specific goals of Siemens and their definition relations decomposed until operational goals are derived.

Table 24: Possible performance indicators.

Criterion	Possible unit of measurement
Track capacity	Trains/hr
Computer calculation gain	Seconds/process
Cost	€ OR amount of employees involved OR Amount of FTE
Throughput time	Days/section length OR days/project€
Amount of ambiguous processes	# or additional time OR # translations
Fewer process interdependencies	#
Amount of data errors	#
Amount of sources	#
Level of meta data	Amount of data links OR Amount of classes

Actor Analysis

This section first shows the involved actors per class and also presents a mean end diagram for the most involved actor: Prorail. Prorail is by far the most important actor since they need to adopt the RailML methodology if Siemens wants to make it a success. From the means end diagram of Prorail in Figure 111 it can also be concluded that their desires are fundamentally different than those of Siemens. The essential difference is that Prorail wants a standard to make competition for rail signaling easier while Siemens wants to gain a competitive advantage by diving in now. What both have in common though is to make the current cumbersome process leaner.

Table 25: Actor classification.

	Dedicated Actor		Non dedicated Actor	
	Critical Actor	Non-critical actor	Critical Actor	Non-critical actor
Same interests	Siemens AG (Germany)		-RailML society	-Siemens rail traffic monitor institutions -IT developers -Rail maintenance companies
Different interests	Prorail	-Competitors -Traffic control -Operators	-Ministry of Infrastructure -Loxia -Engineering agencies	

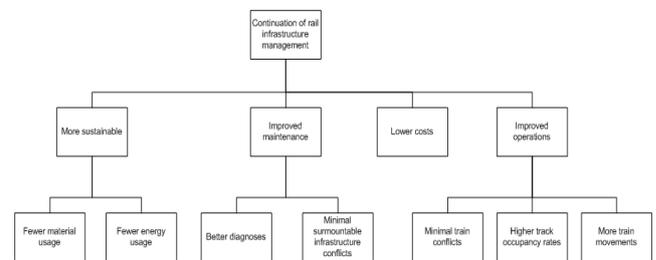


Figure 111: Means end diagram of Prorail.

External Factor Analysis

The next table proves that the major external factors, factors that Siemens cannot directly and reasonably indirectly influence, do not have much effect on this case as we are most interested in those external factors that are both highly uncertain and have a high impact. The only development that could really play a disturbing role in the development of this specific RailML project is competitors doing the same, something equivalent or having a better marketing position. This effect needs to be taken into account when developing RailML in order to make it flexible in case of desired changes towards contractor's desires.

Table 26: The most influential external factors classified by uncertainty and impact.

		Uncertainty	
		Low	High
Impact	Low	<ul style="list-style-type: none"> Railway organization Demographic changes New transport systems 	<ul style="list-style-type: none"> Economic situation Political willingness
	High	<ul style="list-style-type: none"> Modal split Train protection systems Legislation Technological innovations 	<ul style="list-style-type: none"> Level of competition

Impact Analysis

This analysis roughly puts various strategies, to improve Siemens' process chain, into perspective to each other. In general, a cost reduction strategy limits the use of resources and likely comprises on quality; keeping other factors equal. A price reduction strategy would leave product characteristics the same but put margins and long term continuation under pressure. A wider variety of interlocking products could result in simpler integration when dealing with various signaling systems. The downside is that each product needs its documentation, production line, experts, maintenance program and so on; thus increases costs. The opposite makes Siemens less flexible in certain markets which especially puts application under pressure. A technological innovation in the form of an IT tool like RailML increases work productivity at mainly the initial investment cost. Such a measure therefore theoretically allows for fewer trade-offs.

Table 27: Common process optimization strategies versus the aggregated goals of Siemens. Symbols reflect big deterioration (1) to big improvement (5) via no change (3). The total score between brackets reflects the total without taking application into consideration. In the end, the interlocking should remain the same.

Measures	Criteria				Score
	Application	Time	Costs	Quality	
Decrease costs	3	2	4	1	(7) 10
Reduce price	3	3	3	3	(9) 12
Increase amount of products	5	2	1	4	(7) 12
Decrease amount of products	1	3	4	3	(10) 11
RailML	3	5	4	5	(14) 17

Appendix C - Literature Review on Lean Transformations

This appendix contains the main conclusions of a literature review into lessons from lean transformation best practices, lean transformation methodologies, conceptual techniques to accomplish a lean transformation, quantitative techniques to accomplish a lean transformation and performance metrics to evaluate the effects of a lean transformation.

Lean Approach Applicable to Interlocking Engineering Design

To decrease the amount of non added value activities in the rail engineering process, W.W.A. Beelaerts van Blokland et al. (2010) proposes the use of a so-called lean integrator. A lean integrator is person that needs to stimulate and consult a chain's transition from a traditional

approach to a lean one. In addition, Beelaerts et al. suggest to make an agreement on sharing revenues across the chain. Such a policy stimulates the development of an end product where a customer pays for what he desires. Kolic et al. (2012) proposed one welding approach to deliver most value in producing steel plates. This statement can be translated to rail interlocking as having one system for a large market. Hallam (2010) mentions the importance of concurrent design choices during engineering in order to best determine what is considered added value or not. In order to do such, a clear definition of value is necessary. Marzouk et al. (2012) decompose value in internal or external, product or process and soft or hard.

Table 28: 'waste' reduction matrix based on literature.

Best practice	Defects	Overproduction	Gain of lean engineering on type of 'waste'					
			Idle products	Redundant processes	Redundant movement	Redundant transport	Waiting time	Non-value added design
W.W.A. Beelaerts van Blokland et al. (2010) on aircraft manufacturing	-	Focus on economies of scale and substantial innovation packages which leads to few and large batch sizes of a small range in products	-	Lack of co development between suppliers and manufacturers increases need for validity checks	-	Many suppliers and a lack of co development between suppliers and manufacturers increases need for expensive good transport	Many suppliers and a lack of co development between suppliers and manufacturers increases chance for waiting times	Classic production in the form of a total assembly system discourages innovation at main manufacturer.
Hallam (2010) on defense aerospace systems	An increasing number of suppliers increased production errors or incompatibilities	-	The design change rate resulted in 10%-30% of the cases in stoppages of production varying in duration	Growth in program costs due to repeated implementation of small design changes and new technologies	-	50% of aerostructures is outsourced at tiers 2, 3 and 4 in the chain lead to many transport flows	The design change rate resulted in 10%-30% of the cases in stoppages of production varying in duration	Design changes throughout the process can generate product elements not suited to customer needs.
Kolic et al. (2012)	Large bed steel plates increase chance on bad welding as opposed to one sided welding; Using cut-outs for welds is sensitive for low quality welds	-	Flow stoppages because of turning and positioning steel plates along the line	Large bed steel plates require more welding as opposed to one sided welding; Various redundant quality checks	Own axis turns due to large bed steel plates as opposed to one sided welding; Turning the direction of steel plates along the line to allow for welds	-	-	Various quality checks along the line to guarantee a good welding; Many manhours per steel plate
W. Beelaerts van Blokland et al. (2009)	Porsche did not demand high quality of suppliers' parts with consequent defects and unusable parts	Especially in non boom economic times, Porsche cars were simply too expensive and production outpaced sales	Each Porsche design was perfected until engineers matched their desired plan, often making resources on the production line wait	The paint booth accepted that a large chance existed on painting multiple times because of contamination; Quality checks of cars were done multiple times along the line by different engineers that used varying protocols	Manual parts collection for engineers that assembled the engines with additional movements for new needs or part shortages	Due to a relative large amount of defective parts from suppliers, additional parts had to be shipped	Huge inventory of parts that remained a long time at the production plant	Porsche cars featured a lot of craftwork that was not always viewed as added value by the customer
Marzouk et al. (2012) on construction consulting	Inadequacy in technical knowledge increases chance on defects	Unbalanced resource allocation	Lack of confidence in pre-planning design work; unbalanced resource allocation	Poor briefing and communication requires redundant explanation; unclear restrictions and requirements	Lack of or missing information; much individual work;	-	Lack of confidence in pre-planning design work; unbalanced resource allocation; unclear and broad tasks	-

The literature indicates five potential ‘waste’ reducing transformation directions for a rail design process. First, Marzouk et al. (2012) shows that distinguishing ‘waste’ according to the three levels of Womack and Jones (2003a), from added to non added value, prioritizes the ‘waste’ to eliminate at start. Second, Hallam (2010) proposes the use of a design authority. A design authority manages the design choices that are made/need to be made across the process chain. The goal is to reduce design choices that prove complex for downstream tiers, the need to reverse choices downstream and/or implement forgotten design aspects at downstream tiers that might be less efficient in doing so. Third, Kolic et al. (2012) proves that welding more than two steel plates at the same time is more efficient than two. In a rail design process, this translates to processing multiple types of data (machine readable) while directly paying attention to interactions. An example would be to describe track topology while directly defining interlocking logic. Fourth, the Porsche case of Womack and Jones (2003b) emphasizes the importance of standardization in deliverables and procedures. Last, W.W.A. Beelaerts van Blokland et al. (2010) recommends the role of a network cooperation initiative to achieve fair competition. Having such an initiative aligns parties on general agreements and protocols in the design process chain which enables the possibility for a downstream party to provide the means for an upstream party’s specifications. Therefore, it is likely that more party’s can engage in a tender and stimulate competitive tenders. RailML could exactly become such a network cooperation initiative.

One way to stimulate continuously flowing engineering processes is by introducing the already proposed design authority. In addition, Kolic et al. (2012) describes the importance of a level production setup when producing steel plates for ships. In a rail engineering process, this can be translated as modular processing of data. Furthermore, Marzouk et al. (2012) stresses the importance of sharing information across the chain by reducing four information barriers: non-generated information, inaccurate information, incompatible information and excessive information.

Customer pull in rail engineering processes can first be achieved by a clear market focus which, according to Womack and Jones (2003b), enables a sufficient understanding of customer preferences which drives an efficient customer pull. Second, Marzouk et al. (2012) explain what an efficient customer pull is. They show that when information planning is not done backwards, as driven by customer demand, so-called information inventories rise when a server or person is fully occupied with another information process. Therefore it is key to eliminate such information inventories.

Marzouk et al. (2012) and W.W.A. Beelaerts van Blokland et al. (2010) explain that achieving perfection in an engineering process is mostly benefitted from an open source approach. Open source enables the implementation of new innovation without barriers from institutions while all parties can access the latest system at all times.

Table 29: Transformation strategies per lean aspect based on various papers.

Best practice	Key elements of lean engineering				
	Value-added activities	‘waste’ elimination	Flowing processes	Customer pull	Perfection
W.W.A. Beelaerts van Blokland et al. (2010) on aircraft manufacturing	Develop best value supply network by a.o. creating tier structure and a high relationship level. Lean integrator plays an important role to optimize cooperation of parties in chain. Incentive to focus on value added activities by sharing revenues across the chain.	Successful cooperation within the network to innovate and compete. Classify suppliers by means of Kraljic matrix. In addition, suppliers can be rearranged in a pre-assembling, upstream integrating or lowest tier eliminating way to improve performance.	The configuration value driver indicates the degree to which developments and operation processes can be shared with partners in the chain. This is part of the value-leveraging model (3C) and the driver can be measured by using the production multiplier and turnover per capita performance indicators.	A continuation value driver indicates possibilities to change a product in such a way that it adds value to the customer. This is part of the value-leveraging model (3C) and the driver can be measured by means of time to market, market share and break even time performance indicators.	Develop knowledge sharing network. Focus on outbound innovation to find businesses that can better exploit new technology.
Hallam (2010) on defense aerospace systems	The concurrent incorporation of design choices along each tier of the chain prevent an incremental loop of changes at one tier that prevent superfluous work and support of tier’s suppliers.	The concurrent incorporation of design choices along each tier of the chain can shorten lead times significantly. A design authority that governs all changes will improve lead times even further.	A design authority can manage the chain, foresee bottlenecks and make sure the process never stops flowing.	Balance between customer takt time minus the expected value of delay and the planned cycle time should result in optimal productivity to meet customer demand. Furthermore, upstream suppliers should make sure their cycle time is always shorter than the consumer takt time.	Continuous effort to match change incorporation with the required design change rate and achieve concurrent implementation
Kolic et al. (2012) on shipyard manufacturing	High performing accuracy and dimensional control systems allow for further automation and fewer activities per worker. The one sided welding using slots eliminates quality checks as they only allow for welding within a specific dimension.	High performing accuracy and dimensional control systems allow for faster construction. Simultaneous fitting and welding of single steel plates. The use of slots instead of cut-outs to weld eliminate the need of lugs.	Panel assembly line able to facilitate one-piece flow. Level production set up. Equal takt time between work stations to align the processes.	Equal takt time to ensure JIT.	Much lean experience in Asia puts an incentive at mostly European shipyards to make a come-back and optimize the manufacturing line.

<p>W. Beelaerts van Blokland et al. (2009)</p>	<p>A quality offensive to show the value trade-offs between perfect cars vs late deliveries, suppliers' parts defects vs high upfront standards and direct solutions vs. design solutions at the end of production line.</p>	<p>Delaying of operation responsibilities and consequent cost centers and job activities. Reducing the average inventory time of items towards zero. Reducing the amount of jobs to match the value-added craft. Demand supplies only when exactly needed. Standardize the work processes.</p>	<p>A closed loop production line of parts kits collection and engine production with parts delivered at the rate of construction.</p>	<p>Making suppliers aware of the pull principle along the tiers by a special task team that governs the chain. Focusing on a clear market, in this case the relatively more expensive sportscar market</p>	<p>Porsche improvement process to continuously measure the current working performance and make it visible to anyone in the organization. A new suggestion system for employees.</p>
<p>Marzouk et al. (2012) on construction consulting</p>	<p>Value can be decomposed into external and internal value. Furthermore, there is a distinction between product and process value and between soft and hard values. Workshops can help to operationalize the values.</p>	<p>Classify process activities into the three priority levels of 'waste' (muda) and eliminate type 3 that is clearly non value adding.</p>	<p>Continuous information flow is hard in design/engineering, therefore focus on most valuable information. That kind of information must be shared directly although four main barriers prevent this: (1) info is not generated (2) info cannot be identified or is incompatible (3) excessive info or (4) inaccurate info. Goal is to eliminate these barriers.</p>	<p>The main goal is to remove inventory stations where information has to wait before someone will use it. For those purposes, the planning should be made backward from the moment the customer demands the final product instead of pushing it forward.</p>	<p>Endless process to identify new 'waste' in the process. The most important driver is considered transparency to encourage and simplify the findings of process improvements.</p>

Lean Methodologies from Literature

The eventual selection of lean transformation directions depends on the 'waste' discovered in the rail engineering process of Siemens and Prorail. In order to discover the 'waste', a dedicated lean analysis methodology needs to be applied. The next discusses some literature that involves methodology development.

Lu et al. (2011) investigate the implementation of a lean pull system for a TFT/LCD manufacturer. Their approach to find an effective lean pull system is by first identifying value added activities by using a value mapping technique. Then a framework is developed to develop an optimal lean-pull system. This step includes determination of the takt time, determination of system's scheduling point, measures to achieve continuous flow, investigation of required batch locations in the process and, when producing multiple products, measures to achieve a leveled production set up. Third, a simulation model provides insight in the quantification. Fourth, a multi-criteria analysis (MCA) weighs the simulation results after which a judgment, the fifth step, on the desired future state and transformation is given.

Jeong and Phillips (2011) applied a technique of Ulrich and Eppinger (2008) called the Concept Development Process (CDP) as a basis for lean engineering. The CDP entails generation of (1) customer needs, (2) target specification, (3) concept generation, (4) concept selection, (5) concept test and (6) final design specification. The application of the CDP model aims to describe the most important characteristics, goals and means to comply each process step. provides an example of this approach.

Stage	Descriptions
Customer needs identification	<ul style="list-style-type: none"> • Full automation • Highly configurable • Meet annual production requirement (8.5M)
Target specification	<ul style="list-style-type: none"> • Throughput = 8.5M units per year or 31 units/min • Define operation cycle time • Conveyor transportation speed (0.5 inch/sec)
Concept generation	<ul style="list-style-type: none"> • Current Layout (CL) • Express Lane Layout (ELL) • Independent Zone Layout (IZL)
Concept selection	<ul style="list-style-type: none"> • Modular design-based concept selection • Simulation-based concept selection
Concept test	<ul style="list-style-type: none"> • Diverse what-if analysis for the concept selected
Final spec and product development plan	<ul style="list-style-type: none"> • Test results and design spec ready for implementation

Figure 112: example of Jeong and Phillips' (2011) approach.

Dotoli et al. (2012) develop and test a five step iterative lean manufacturing strategy. Their approach starts with an as-is activity diagram to describe the manufacturing process in much detail. Second, an as-is state map is developed to visualize the process. Third, the ideal future state is described in a to-be state map and a to-be activity diagram. Fifth, a simulation study shows the performance of the system. When the performance proves insufficient, one needs to start from the third step until the system shows the desired (direction of) results.

Agyapong-Kodua et al. (2009) researched the transformation of static value chain models into dynamic simulation models for use in lean manufacturing enterprises with product variety. The authors developed a four step approach in which existing static value stream techniques are applied in a dynamic value stream context. The approach is initiated by identifying 'waste' throughout the process. Second, the impact of inventories, queues and delays on the system performance is evaluated to prioritize bottlenecks. Third, test combinations of parameter changes. Fourth, provide a recommendation on the basis of the results in the previous steps that lead towards a lean process structure.

Marzouk et al. (2012) developed and applied a lean transformation methodology specifically for lean design engineering processes. They propose to start with a flowchart method to visualize the process management structure of the engineering agencies. The flowchart data will mostly rely on interviews and monitoring. The second step entails a classification of the various project types that run through the process indentified in the flowchart. This second step relies mostly on expert judgment. The third step encompasses the built of a simulation model that encompasses the essential project development. The fourth step then focuses on making the current process structure lean. The last step compares the results of new proposed process structures with the current situation to make a second iteration of lean measures before a final alternative is proposed.

Parry, Mills, and Turner (2010) designed a 4-step methodology to make a production process leaner while keeping

organization's core competencies. The first step in their methodology is to perform a market analysis by means of Porter's competitive forces model to identify the organization's context, their products and market developments over time. The second step analyzes the value chain resulting in a clear distinction in value and non-value added activities, the flowability of the chain and the most important assets involved. This step needs to depict the complexity as a result of all interdependencies involved. The third step involves the understanding, incorporation and sustainment of customer value across the chain, mainly accomplished by interviews. Last, a cost centre analysis aims to put costs in the chain at positions where they are really spent. Thus cutting overhead as much as possible.

Conceptual Models from Literature to Achieve a Lean Transformation

Rother and Shook (2009) introduced Value Stream conceptual Mapping (VSM) in 1998 to identify 'waste' in a production or engineering process and provide leads to eliminate it. A value stream defines all required actions to design and produce a product. VSM is a commonly used technique when it comes to describing the value chain and achieving a lean transformation; especially in an early design state (W. W. A. Beelaerts van Blokland, 2013). Rother and Shook state mainly four advantages of VSM: an overall value chain view instead of a single process focus, face discovery of bottlenecks' sources, a general and straightforward process communication language and the basis for a change and implementation plan. The authors furthermore stress the importance of this qualitative tool to discover 'waste' and describe how a process should work. In contrast, a quantitative tool should ideally function as a way to show the urge of the problem upfront and the performance of a new approach.

VSM starts with a specification of the customer's needs in terms of the final design or product per time unit. VSM eventually results in a drawing which presents the various activities that lead to that final product. Each process consists out of several performance characteristics like the cycle time, number of involved workers, uptime of limited resources, changeover time and so forth. In addition, inventories between each process indicate how many products or much data waits before it actually gets processed. Furthermore, the VSM provides an overview of the lead time versus the actual processing time per activity and how production control of the specific institution manages each activity.

Lu et al. (2011) applied the VSM technique as a conceptual tool for a lean transformation. They confirm the tool's ability to get detailed insight in the (rail engineering) process, the rational discovery of 'waste' and the supportive function when rearranging the process structure. Furthermore, VSM enables a visualization of the required process characteristics to make a transformation feasible. The downside of VSM is however that the potential future states which all reduce 'waste', cannot easily be compared in effectiveness. Quantitative models, discussed later in this appendix, suit better for that purpose.

Jeong and Phillips (2011) studied how the shortcoming of VSM that Lu et al. among others identified, could be overcome. On the one hand they investigated the use of simulation to validate VSM and on the other hand they tried to merge VSM with simulation. Regarding conceptual modeling, the authors propose to generate future concepts from a brainstorm partly on the basis of VSM. Furthermore, they propose to split the concept selection process into a qualitative and quantitative analysis. The qualitative approach consists out of a brainstorm and experts' observation to identify the significance of the most crucial aspects in the developed concepts. Figure 113 provides an example of Jeong and Phillips' conceptual approach.

Concept	Characteristics	Potential Issues	Significance
CL	• Two loading machines feed entire systems	• Enough capacity to feed an entire system?	High
	• A pallet switching between PFS and TCS	• The pick & place could block pallet flow	Medium
	• Shared returning conveyor belt	• Congestion on the shared conveyor belt due to the mix of loaded and unloaded pallets	High
ELL	• Two loading machines feed entire systems	• Enough capacity to feed an entire system?	High
	• No pallet switching	• Longer pallet travel distance may impair the pallet distribution, which may exacerbate the congestion	High
	• The express lane added	• Expedited the pallet movement	High
IZL	• Each zone works independently with its own loading machine	• More time and effort to build the system • More space required	Low
		• No interaction between zones • Enough capacity to satisfy demand?	High

Figure 113: An example of Jeong and Philips' (2011) approach to conceptualize the process.

Dotoli et al. (2012) make a distinction between activity diagrams and state maps when conceptualizing manufacturing or design processes. The authors use the UML technique for activity diagrams and VSM for state maps. The UML technique is chosen being a standardized language to describe process activities and the involved actors. The last provides the input of an activity diagram that qualitatively describes system dynamics. VSM, as already discussed, visualizes process activities in such a way that a lot of 'waste' can be discovered by face.

Agyapong-Kodua et al. (2009) first apply the Computer Integrated Manufacturing Open System Architecture (CIMOSA) modeling technique followed by VSM to conceptualize their considered manufacturing processes. CIMOSA consists out of four coherent conceptual diagrams including a top-level index diagram, an interaction diagram, a structure diagram and an activity diagram. These diagrams form the backbone for a lean optimization as they decompose and visualize enterprises' activities as a network of dependent processes. The authors do however make a note that the development of an acceptable CIMOSA model takes a lot of time due to capturing, connecting and validating the data.

W.W.A. Beelaerts van Blokland et al. (2010) use various conceptual tools to achieve a lean value chain in the aerospace industry. First, a general supply chain visualization technique is applied with the focal company at the heart of the chain. Second, a value network visualization technique broadens the supply chain view by showing the interactions between suppliers in a tier. The third conceptualization tool Beelaerts van Blokland et al. uses is the

Kraljic matrix to categorize suppliers by supply risk and profit impact.

Mahfouz et al. (2011) choose the modeling technique IDEF0 for their packaging production simulation mostly thanks to its strong hierarchical nature. A hierarchical nature is easy to understand but also enables an unambiguous translation into the simulation model which improves verification of the model. An IDEF0 starts with the AO format which is the top layer and only consists out of one process block depicting all the ingoing and outgoing flows of the entire system. The A1 is a decomposed AO IDEF0 containing usually three to five process blocks that each have an ingoing and outgoing flow of input, output, support and/or information to visualize the production process. The decomposition can continue until the modeler sees a complete process fit.

The Supply Chain Operations Research (SCOR) model describes a supply chain by means of a performance indicator set (Supply Chain Council, 2010). The Supply Chain Council developed this model to derive a standardized approach in describing supply chains which would allow for easy comparisons, analyses and adaptations. The SCOR contains three levels: process types, process categories and process elements. SCOR defines five process types: plan, source, make, deliver and return. The process categories make sure the process types get planned, executed and supported. The process elements are dedicated to detailed performance measurement. The SCOR metrics focus on five focus areas: supply chain reliability, responsiveness, agility, costs and asset management. The Council stresses that SCOR does not provide any tools to optimize a chain but analysts should rather see it as a benchmarking tool.

Quantitative Methods from Literature to Achieve a Lean Transformation

Mahfouz et al. (2011) investigated simulation as a tool to optimize the production process of a packaging manufacturer. More specifically, the authors developed a discrete event simulation model with the aim to first find a relation between lean performance indicators and variables within the model, second to find the limiting factors in the model and third to optimize the system by altering the limiting factors. Mahfouz et al.'s model bases itself on resources and their utilization constraints, products and their process characteristics, process blocks with statistical deviations and logical points to allow for special production cases and consequent routes through process model. The authors were able to validate the model by means of face validation and comparison analysis. They discovered that simulation provides a business with the mean to identify the effects of proposed process changes ahead of implementation. Therefore simulation will rationalize business' decision making and allows it to take focused measures to mitigate risks when they prevail.

Jeong and Phillips (2011) use Flexsim software simulation package to perform a simulation study in which multiple concepts are compared on the basis of throughput, process times, queues and capacity. In order to determine the

parameters and probability distributions, mostly experts were approached due to a lack of full data.

Dotoli et al. (2012) use the ARENA software simulation package to execute the simulation study. The package has been chosen since an UML can easily be translated into the ARENA language. Furthermore, ARENA suits well with large modular processes. The construction of an ARENA model starts with the model translation from the UML. Then the parameters are put into ARENA model. Those include activity times, process probabilities, resource restrictions and average input rates. Last, a validation process is performed to evaluate the results of the model.

Agyapong-Kodua et al. (2009) apply system dynamics and discrete event simulation as tools to achieve a dynamic value stream model. The authors define a dynamic model as a model that incorporates interdependencies and interactions between variables to estimate system performance of future state value streams. Literature analysis of Agyapong-Kodua et al. pointed out that a system dynamics model based on causal loops and consequent differential equations distinguishes cause-effect relations, and as such, results in a set of potential change parameters. Furthermore, system dynamics provides insight in the probable change of variables when altering the parameters. The authors do however point out that system dynamics' disadvantage is that it is not parametric, i.e. a parameter change effect results in a behavioral direction but not in an exact quantification. This is why the authors applied discrete event simulation by means of the Simul8 software package to imitate a value stream for a specific time duration. Just like Dotoli et al. (2012), Agyapong-Kodua et al. confirm the ease of translation between the conceptual VSM and discrete event simulation. Furthermore, they also state the effectiveness of discrete event simulation in predicting and verifying bottlenecks of 'waste'.

Marzouk et al. (2012) use Extend V.6, based on ExtendSim, for discrete event simulation of engineering processes. On the one hand a simulation tool was used because of the need to model interdependencies that flowcharts cannot represent well, on the other hand to enable performance measurement. The authors choose Extend V.6 because they felt that it closely matched the building blocks used in engineering processes.

Gregg et al. (2011) also use ExtendSim as a basis to design a Lean Process Analysis Simulation (LPAS) tool for aircraft manufacturing. The main reason for Gregg et al to use ExtendSim was the open source nature of the programming language, a clear database structure and a messaging structure that matches the field of aircraft construction. Characteristic functionality of ExtendSim with LPAS includes quick data access, parent-child overview structures, incorporated distributions, process evaluation based on queuing theory, schedule evaluation, performance evaluation of changes in production set-up, cost structure analyses, extended time evaluation and statistical analyses. The authors did however find that the simulation model is not very capable to deal with continuous changes

in the value stream and scheduling. Furthermore, the cost analysis' accuracy leaves room for improvement.

The generic ExtendSim spinoff VSMSx enables the simulation of VSMs of various different systems (Shararah et al., 2011). The authors observed furthermore that VSMSx allows easy conversion of a VSM into the simulation tool since it contains the same building blocks in the GUI. Basically, an analyst draws a VSM, collects data, puts both in VSMSx, validates and verifies the model and collects the performance indicators. Besides, VSMSx has an accessible database structure and allows the use of stochastic variables. In addition, VSMSx enables animation of the simulation process by showing the current position of an entity. The accessibility of VSMSx contains some disadvantages as it is mainly focused at production processes using equivalent input data and resulting in a limited set of process options and performance indicators. Furthermore, verification and validation is mainly limited to face validation and expert opinions.

Cuatrecasas-Arbos et al. (2011) provide an alternative to discrete event simulation by the use of Operations-Time Charts (OTS). The authors investigated the graphical evaluation tool in the context of designing and optimizing production processes as it allows the behavioral study of what-if scenarios. An OTS model contains a chain of processes with a production schedule, graphically very similar to a GANT chart, to result in a set of parameters like inventory levels, performance measurement e.g. process times and also the evolution of those performance metrics through the system. The tool is not simulation which implies that all parameters are static and not provided dynamically over (simulation) time.

The authors compare OTS with project management software (PMS), production scheduling software (PSS) and discrete event simulation. PMS focuses on scheduling and OTS on production. Therefore, OTS is somewhat more complex but does provide the opportunity to calculate system performance indicators, allows for process loops and various process flow concepts like the lean supermarket. OTS, compared to PSS, enables the identification of 'waste' throughout the process since lead times are not a constant. PSS makes lead times constant because it focuses on balancing demand and available resources. Therefore, PSS does however provide the analyst with the option to model multiple products that use the same resources. Already hinted at the start of this paragraph, the main advantage of discrete event simulation is the evaluation of parameters over time. On the other hand, discrete event simulation does not provide a visual representation of the system's parameters at the end.

Hoogendoorn et al. (2007) studied the Tool for Object-oriented Modeling And Simulation (TOMAS) being discrete process simulation of complex logistic and production systems. Discrete process simulation differs from discrete event simulation by indicating the time that an element remains in same state. Whereas discrete event simulation processes are triggered by the occurrence of an event. For that purpose, TOMAS combines all event processes of an attribute in one. TOMAS has the flexibility to model unique

process structures and aligns with common business applications. Furthermore, TOMAS includes an animation module and the program allows for easy conversion to real-time support and application. Delphi is the underlying programming language due to reasons of simulation speed, accessible programming knowledge and possibilities for an object-oriented approach. The authors explain that TOMAS distinguishes active and passive processes. Both are called a TOMASelement because occasionally passive processes are part of another active process. Each TOMASelement uses or does not use one of two process approaches: altering the current process or the TOMASelement. Current processes can imply hold, standby, suspend and stop; methods of a TOMASelement start, resume, interrupt and cancel. These two methods enable parallel processing, as the Delphi notation covers only object, by continuously determining the system state and which process needs to progress. The authors state that the disadvantages of TOMAS include no presence of a GUI and a lack of proper combination of qualitative business models with the quantitative TOMAS approach.

Performance Metrics from Literature to Evaluate a Lean Transformation

Rother and Shook (2009) provide means to evaluate a value chain's performance on the basis of their Value Stream Mapping technique. Two performance metrics are directly related to process duration. First, the value-creating time (VCT) should match the cycle time (C/T), defined as the time it takes to process the subsequent unit, as close as possible. Second, the lead time (L/T), the time it takes for one unit to go through the entire value chain, should be shortened as much as possible by eliminating 'waste'.

Besides process duration, Rother and Shook define another set of performance metrics based on the most fundamental problem of mass production: overproduction. First, goal is to produce at takt time. The takt time in Equation 6 indicates how long duration might take to match the required sales rate.

$$Takt\ time = \frac{Available\ working\ time\ per\ day}{Customer\ demand\ rate\ per\ day}$$

Equation 6: The takt time equation.

Second, Rother and Shook point out that the amount of inventories should be minimized to achieve continuous flow. Therefore the amount of separate process steps or stages and the amount and size of inventories are a measure of the flow ability.

Hallam (2010) proposes to use a metric that measures the balance between productivity and planning risk by Equation 7.

$$t_{cpn} = takt\ time - EV(t_{dn})$$

Equation 7: The planned cycle time equation.

where:

t_{cpn} = planned cycle time [h]

takt time = the required time between consecutive products as function of available work time and customer demand [h]

$EV(t_{dn})$ = expected value of the delay time derived from a steady state distribution [h]

Jeong and Phillips (2011) developed the System Coupling Level Index (SCLI) as a way to measure complexity. The SCLI is part of the quantitative analysis in their CDP model and based on the notion that more interfaces between modules increases complexity. Based on the number of modules in each proposed concept and a N2 diagram, Equation 8 determines the total number of possible interfaces TPI:

$$Total\ Possible\ Interfaces\ (TPI) = \frac{m(m + 1)}{2}$$

Equation 8: Equation to determine the total number of relations in a process chain.

where:
m = number of modules

In the same way the existing number of interfaces TEI can be determined. The ratio between the TPI and TEI results in the SCLI as shown in Equation 9.

$$SCLI = \frac{TEI}{TPI}$$

Equation 9: The System Coupling Level Index equation to measure complexity as a ratio between total existing process relations and the maximum amount of process relations.

Dotoli et al. (2012) specifically focus on lead time and resource utilization in their simulation studies. Lead time being defined as the average time between the start of a completely new order until it is out of the considered system defined in the VSM. The authors defined resource utilization as the percentage of scarce resources busy, in the case of their study scarce resources were workers.

Lu et al. (2011) developed a Multi-Criteria Decision Making method (MCDM) based on a combination of Taguchi's quality loss function and The Order Preference by Similarity to Ideal Solution method (TOPSIS). The quality loss function of Taguchi (1986) measures the deviation of a process' performance towards a specified outcome using Equation 10.

$$L_{ij} = k (y_{ij} - m_{ij})^2$$

Equation 10: Quality loss function from Taguchi (1996) as item's squared deviation from a performance target expressed in a monetary value.

where:
L = the quality loss
k = the cost factor associated to not producing at specification
y = the simulation result on the performance metric
m = the specification value of performance
i = the nth scenario
j = the nth response

The TOPSIS approach accordingly normalizes and rates the outcome of Equation 10. Second, the derived rates need to be weighted before a best and most negative solution can be determined. The fourth step in TOPSIS then calculates the sum of the squared difference of each rate towards the best and most negative solution using a standard Euclidean distance formula. Fifth, a factor level of the worst solution compared to the ideal solution can be determined before a ranking of alternatives can be made.

Marzouk et al. (2012) focus on utilization rates of activities throughout an engineering process. Furthermore, they looked into the degree of projects initiated versus completed within the covered time-span of the simulation to indicate the leanness of the engineering chain.

Mahfouz et al. (2011) use mainly three performance indicators in their simulation study on packaging production that are coherent with lean thinking: cycle time, work in process and workforce utilization. The authors are convinced that those parameters provide a complete view by lean conclusions on demand management, maintenance, capacity and product flow.

W. Beelaerts van Blokland et al. (2009) develop, use and test the so-called 3C value leveraging model (Figure 114). The 3C model intends to balance the value drivers continuation, configuration and conception. Continuation concerns long-term growth of operations and therefore implies meeting customers' requirements. Continuation can be measured by means of (1) time to market, (2) break-even time, (3) market share and (4) profit per capita. The authors define conception as the stimulating factor for innovation as a derivative of customer demand. Beelaerts van Blokland et al. therefore indicate that the degree of conception can be measured by an innovation investment multiplier and R&D expenditure. The last C, configuration, encompasses the layout of the value chain. Here, the authors measure performance by determining the production multiplier and the turnover per capita.

Another paper of Beelaerts van Blokland (W. W. A. Beelaerts van Blokland et al., 2012) proves the leveraging relation between the 3C drivers for the aerospace industry. This conclusion has been based on a significant relation between turnover per employee, profit per employee and R&D expenditure per employee. Furthermore, the paper explains that a supply chain party should focus on balancing the 3C drivers as much as possible to achieve stable generation of value. This implies that the supply chain party is able to transform products' value of suppliers into the right product value as associated to customers' requirements. In the absence of a clear value balance, aircraft integrators in this case loose competitive power due to insufficient R&D policies that result in products with characteristics that customers do not value / want.

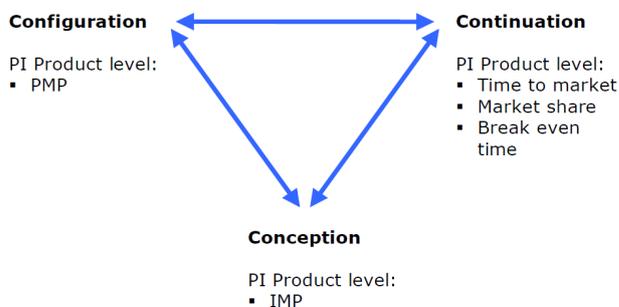


Figure 114: The 3C value leveraging model according to Beelaerts van Blokland (2009).

A Concurrent Value Assessment model (CVA) enables the quantification of a feasible target price level on the basis

of customer's (perceived) product value, a target cost level, control of engineering costs and control of suppliers' costs (Sprengers & Beelaerts van Blokland, 2013). The CVA bases itself on a matrix eigenvalue problem that reflects the relative importance of customers' criteria versus engineering measures, e.g. safety versus product weight. The result leads to customer perceived and technical competitive power value evaluation matrices. Sprengers and Beelaerts van Blokland first use those results to calculate the relative Degree Of Importance (DOI) to determine every used parts' importance in comparison to the specific engineering measure. Second, the authors compute the Degree Of quoted Cost (DOC) which reflects a parts' cost share of the total product cost. Third, a Degree Of customer Perception (DOP) follows from customers' perceived value and the DOI. These three indicators, DOI, DOC and DOP, form the main output of the CVA model and should ideally be balanced. If not, there is a clear solution direction, e.g. when DOP is larger than DOI but DOC is lower than DOI, this implies that costs can go down to lower customer perception to the importance level. The authors do however state that a big limitation arises from the fact that customer perception is measured from current products and markets rather than the eventual adapted product. Furthermore, customer perception is measured on the basis of satisfaction though more performance indicators like quality and reliability exist.

Della Bruna, Ensslin, and Ensslin (2011) researched a supply chain performance evaluation model for a production process of refrigerators in Brazil. The research contains two interesting outcomes: for one a comparison between assessment methodologies and for another the development and visualization of the assessment methodology that scored best. Three assessment methodologies have been studied: the balance score card methodology, multi-criteria decision analysis and the analytic hierarchical process methodology. The authors found that the multi-criteria decision analysis performed better than the other two methodologies because of the flexibility to adapt to the problem context, the exclusion of non interesting aspects, indexed performance measurement and an accessible application procedure. Della Bruna et al. accordingly developed an application strategy for multi-criteria decision analysis based on means-end trees. The operational goals in the lower parts of the means-end tree are quantified on an ordinal scale, e.g. delivery percentage from 0% to 100%. Each ordinal scale is then classified from bad to good performance and associated with a certain value level using MACBETH's judgment matrix. This can then be transformed to a cardinal scale value function which allows for easy comparison of different alternatives. Furthermore, a performance graph can be drawn for each alternative represented by a line that indicates the performance level for each performance indicator. A performance comparison results when each alternative is put into the same graph and/ or an overall score is calculated.

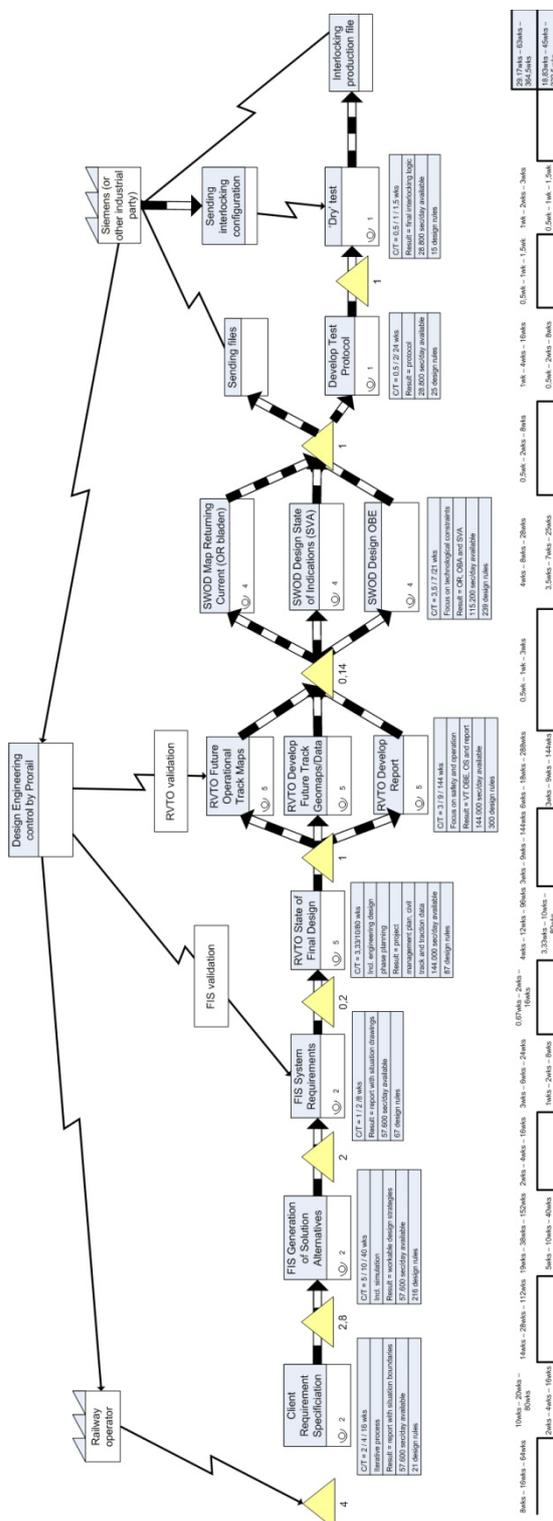
Cuatrecasas-Arbo et al. (2011) came up with a list of ten main performance metrics for their Operations-Time Chart analysis in production processes:

1. Total process lead time

2. Time till first unit
3. Process times with a focus on takt time and difference between value and non value added time
4. Total waiting time
5. The average waiting time
6. Size or duration of work in process inventory
7. The average of the size or duration of work in process inventory
8. The instant size or duration of work in process inventory
9. Productivity
10. Cycle time

Behrouzi et al. (2011) investigated the Iranian automotive industry to discover which performance metrics were used most often to accomplish a lean transformation. The authors did this on the basis of a means-end framework were they defined performance metrics regarding 'waste' in terms of quality and cost, delivery & reliability and flexibility. The research revealed that supply chain flexibility's metrics were not used much by producers, quality metrics however were generally used a lot. In terms of quality, the customer rejection rate, supplier rejection rate and defect rate were weighted highest. In terms of cost, energy costs and labor value added cost were weighted highest. In terms of delivery & reliability, perfect order fulfillment rate to suppliers and consumers, supplier lead time, on-time production and customer delivery lead time were weighted most important. In terms of flexibility, supplier volume flexibility, supplier delivery flexibility and manufacturer volume flexibility were weighted highest.

Appendix D - Prorail's Lean Design Performance



Experts provided estimations for the wait times and value add times shown in Table 30 and Table 31. The sum of the wait and value add times lead to the cycle times in Table 32.

Table 30: (non value added) wait times of the interlocking design processes

Wait times weeks	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	8	14	2	0.67	3	0.5	0.5	0.5
Mid complexity	16	28	4	2	9	1	2	1
High complexity	64	112	16	16	144	3	8	1.5

Table 31: Value added process times of the interlocking design processes.

VA times weeks	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	2	5	1	3.33	3	3.5	0.5	0.5
Mid complexity	4	10	2	10	9	7	2	1
High complexity	16	40	8	80	144	25	8	1.5

Table 32: Cycle times, i.e. wait times plus VA times, of the interlocking design process.

Cycle time weeks	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	10	19	3	4	6	4	1	1
Mid complexity	20	38	6	12	18	8	4	2
High complexity	80	152	24	96	288	28	16	3

The share of value added time per process follows from Equation 11:

$$VA\% \text{ of process } x = \frac{\text{value added time of process } x}{\text{cycle time of process } x}$$

Equation 11: The value added share equation.

Table 33: The shares of VA time of cycle time.

% VA	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	20.0%	26.3%	33.3%	83.3%	50.0%	87.5%	50.0%	50.0%
Mid complexity	20.0%	26.3%	33.3%	83.3%	50.0%	87.5%	50.0%	50.0%
High complexity	20.0%	26.3%	33.3%	83.3%	50.0%	89.3%	50.0%	50.0%

Table 34 presents the deviation in value added time share per process as a function of the mean:

$$VA\% \text{ deviation of process } x = (VA\% \text{ of process } x - \frac{1}{8} \sum_{x=1}^8 VA\%)$$

Equation 12: The value added time deviation equation.

Table 34: The deviation in VA time share compared to the mean.

Deviation	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	-30.0%	-23.7%	-16.7%	33.2%	0.0%	37.5%	0.0%	0.0%
Mid complexity	-30.0%	-23.7%	-16.7%	33.3%	0.0%	37.5%	0.0%	0.0%
High complex.	-30.0%	-23.7%	-16.7%	33.3%	0.0%	39.2%	0.0%	0.0%

A project's wait time is translated to NVAW by the following formula:

$$\text{Data inventory size of process } x = \frac{\text{Wait time of process } x}{\text{Value added time of process } x}$$

Equation 13: The equation to estimate the data NVAW size in number of projects

Table 35: Data NVAW sizes during the interlocking design processes.

NVAW	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test
Low complexity	4.00	2.80	2.00	0.20	1.00	0.14	1.00	1.00
Mid complexity	4.00	2.80	2.00	0.20	1.00	0.14	1.00	1.00
High complexity	4.00	2.80	2.00	0.20	1.00	0.12	1.00	1.00

The amount of design rules follow from various design documents (Prorail, 2010a, 2010b, 2010c, 2010d, 2012b; RIGD-Loxia, 2011) and expert opinions (Koelewijn et al.,

2013). Each process knows a fixed amount of rules to complete; comparable to a checklist. presents the amount of such rules per process.

Table 36: The amount of design rules measured in the amount of requirements per design process. The dry test and test protocol processes are not included because there was insufficient data and knowledge to provide a fair indication.

Processes	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD
# design rules	21	216	130	87	149	109
# design rules in possible second branch	-	-	-	-	45	120
# design rules in possible third branch	-	-	-	-	180	130
Total # design rules	21	216	130	87	370	359

A dimension to measure complexity is by means of the relations between processes. Table 36 shows an N2 diagram to present the relations per process. Please refer Appendix D for the value stream map which clarifies some choices.

A CRS is likely to change when the design engineering process is still in its FIS, maybe RVTO process. When an engineering agency develops the SWOD, it is unlikely that the CRS changes. The detailed elaboration of various alternatives depends on the state of the CRS; the results will be assessed in the model analysis process. This model analysis directly evaluates the possible strategies to present the RVTO with one solution direction. The RVTO designers (usually from another office) take that conclusion for granted, gather the relevant data and produce a planning. The result gets shared to the other RVTO processes. In the case RVTO work gets rejected, this could be traced back to the statement of final design process. Therefore, three to and three from the RVTO maps and report process, are pos

sible. The three sub processes in the RVTO maps and report process include verification with each other which results in three relations per process. Furthermore, Prorail provides feedback to the result of each sub process which increases the amount of relations to 12. When Prorail approves the result, the work needs to be distributed to the three sub processes of the SWOD. Three relations times three sub processes gives nine relations, and the same goes for SWOD's relation with the RVTO maps and report. In addition, the SWOD forms the basis for a test protocol as well. The test protocol derives data from the SWOD and provides the result to the dry test. Please refer Appendix E how the relations to and from Siemens are organized. The total for each process step is also called the number of total existing interfaces (TEI).

Jeong and Phillips (2011) define a complexity measure on the basis of TEI: the System Coupling Level Index (SCLI). Appendix C presents the related formulas.

Table 37: The complexity measured in the amount of interfaces. Appendix E discusses the amount of interfaces at the dry test.

Interfaces	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD	Test protocol	Dry test	TEI	TPI	SCLI
CRS	1	1	1	1	1				5	78	0.064
FIS alternatives	1	1	1						3		0.038
FIS system requirements	1	1	1	1					4		0.051
RVTO final design state			1		3				4		0.051
RVTO specification				3	12	9			24		0.308
SWOD					9	9	3		21		0.269
Test protocol						3	1	1	5		0.064

In order to assess the cost criterion, a productivity index has been developed. The productivity of a worker has been defined as the product of rail elements and design rules he/she can process per week. Equation 14 shows this relation:

Equation 14: A standard to measure the productivity of Prorail's interlocking design process.

$$Productivity = \frac{(\text{amount of rail elements at complexity level} * \text{number of design rules})}{\text{Value added time of process}} \Bigg/ \text{number of employees}$$

Table 38: Productivity of each interlocking design process according to Equation 14. The dry test and test protocol processes are not included since the amount of design rules could not be estimated in the previous.

Productivity	CRS	FIS alternatives	FIS system reqs.	RVTO final design state	RVTO specification	SWOD
Low complex	99.8	410.4	636.5	99.3	473.7	487.2
Mid complex	149.6	615.6	954.8	99.2	473.7	730.8
High complex	120.8	496.8	770.5	40.0	95.6	660.6

Appendix E - Siemens' Lean Engineering Performance

This appendix presents all results from the data analysis and statistical analysis that assess Siemens' lean performance of the current interlocking process. The analysis contains thirteen cases which represent four different interlocking projects of different interlocking project types.

Allocation of Project Accounts

Siemens registers contractual project data in a format imposed by the Dutch infrastructure provider Prorail. An allocation productmatrix divided the data over the four process steps. The next tables show the allocation productmatrices per project type. The process costs per account are then calculated by multiplying the specific process productmatrix column with the entire projectmatrix.

Contractual Account New Interlocking		Process	Productmatrix for each Interlocking Engineering Process				
Projectmanagement [Interlocking]			Engineering or production factor	Data Preparation	EVP Engineering	EVP Conversion	EVP Test
	Overall IXL-projectmanagement	all	0.333333	0.0825	0.0825	0.0825	0.0825
	Technisch projectmanagement	all	0.333333	0.0825	0.0825	0.0825	0.0825
	Financieel projectmanagement	all	0.333333	0.0825	0.0825	0.0825	0.0825
	Site projectmanagement	none					
	Interface (raakvlak) projectmanagement	all engineering		0.25	0.25	0.25	0.25
	Safetymanagement	all	0.333333	0.0825	0.0825	0.0825	0.0825
	Qualitymanagement	all	0.333333	0.0825	0.0825	0.0825	0.0825
	Logistiek medewerker	none					
	Documentmanagement, planning en support	all	0.333333	0.0825	0.0825	0.0825	0.0825
GENERIEK SYSTEEM							
Ontwerpen generiek systeem							
	Overall systeem ontwerp						
	ontwerpen van oplossingen	evp		0	1	0	0
	afleiden van eisen	translate		1	0	0	0
	verificatie ontwerpdocumenten	translate en evp		0.5	0.5	0	0
	requirement tracing	translate		1	0	0	0
	RAMS	test en translate		0.5	0	0	0.5
	Bouwen generieke software						
	Lab. Test						
	IXL	test		0	0	0	1
	Astris adapter	test		0	0	0	1
	Decentrale apparatuur	translate		1	0	0	0
	Operation Control	test		0	0	0	1
	Systeemmanagement	all engineering		0.25	0.25	0.25	0.25
	Validatie	evp		0	1	0	0
	Projectmanagement ontwerpen generiek systeem	all engineering		0.25	0.25	0.25	0.25
	Aanpassing Operation Control	projection		0	0	1	0
	Aanpassing Power Supply	projection		0	0	1	0
	GASC	all engineering		0.25	0.25	0.25	0.25
	ISA	projection		0	0	1	0
Ontwerp TESTSYSTEEM							
	Overall systeem design	evp		0	1	0	0
	Ontwerp t.b.v. Installatie en bedradingstekeningen	evp		0	1	0	0
Hardware TESTSYSTEEM							
	IXL Centraal	projection		0	0	1	0
	Interface KEV/KBV adapter	projection		0	0	1	0
	Lokale bediening	projection		0	0	1	0
	Managementsysteem	projection		0	0	1	0
	Telecomnetwerk	projection		0	0	1	0
	IXL Decentraal	projection		0	0	1	0
	Input boards	projection		0	0	1	0
	Output Boards	projection		0	0	1	0
	Onderhoudstools	projection		0	0	1	0
	Subsysteem Voedingen	projection		0	0	1	0
	Service and Diagnostic	projection		0	0	1	0
	Cable Distribution	projection		0	0	1	0
	Overige testcomponenten	projection		0	0	1	0

	Overige hardware testsysteem		projection		0	0	1	0
Ontwikkelen simulatoren								
	ontwikkelen (onderposten, elementen)		test		0	0	0	1
	ontwerpen/bouwen		test		0	0	0	1
	labtesten (onderposten, elementen)		test		0	0	0	1
	labtest		test		0	0	0	1
Assemblage TESTSYSTEEM								
	Centrale interlockingsysteem		test		0	0	1	0
	Decentrale interlockingdelen		test		0	0	1	0
Installatie en Montage TESTSYTEEM on testsite								
	Plaatsen, bekabelen en IBS		test		0	0	1	0
Documenten en instructiehandleidingen TESTSYSTEEM								
	As Built documentatie		test		0	0	0	1
	Handleidingen		test		0	0	0	1
Reservdelen TESTSYSTEEM								
	Reservdelen							
LOKATIESPECIFIEKE TOEPASSING								
Ontwerpen lokatiespecifieke toepassing								
	Overall systeem ontwerp							
	ontwerpen van oplossingen	evp			0	1	0	0
	afleiden van eisen	translate			1	0	0	0
	verificatie ontwerpdocumenten	translate en evp			0.5	0.5	0	0
	RAMS	test en translate			0.5	0	0	0.5
	Data prep.							
	IXL	evp			0	1	0	0
	Astris - adapter	evp			0	1	0	0
	Decentrale apparatuur	evp			0	1	0	0
	Operation Control	evp			0	1	0	0
	Lab. Test							
	IXL	test			0	0	0	1
	Astris - adapter	test			0	0	0	1
	Decentrale apparatuur	projection			0	0	1	0
	Application Specific Integration	test			0	0	0	1
	Data Test (ATZ)	test			0	0	0	1
	Ontwerp t.b.v. Installatie en bedradingstekeningen							
	IXL	evp			0	1	0	0
	Lokatiespecifieke validatie	test			0	0	0	1
Ontwerp subsystemen								
	Gebouwen / relaishuizen							
	Conditionering							
	Civiele bouw							
	Toegangswegen							
	Verlichting							
	Klimaatregeling							
	Algemeen							
	Voedingen							
	Telecommunicatiesystemen							
Hardware ON SITE								
	IXL Centraal							
	KEV/KBV - adapter							
	Lokale bediening							
	Managementsysteem							
	IXL Decentraal							
	Input boards							
	Output Boards							
	Onderhoudstools							
	Subsysteem Voedingen							
	B-Relais en B-Relaiskasten							
	Cable Distribution							
	Network							
	Juridical Recorder							
	Others							
Diensten								
	Vertalen documentatie	all		0.333	0.0825	0.0825	0.0825	0.0825
Assemblage								
	Centrale interlockingsysteem							
	Decentrale interlockingdelen							
Installatie en Montage on site								
	ASTRIS - adapter							
	Plaatsen B-Relais							
	Werkvoorbereiding							
	Montage hoofdpst							

	Software laden								
Testen, verificatie en validatie									
	FAT		test		0	0		0	1
	SIT1		test		0	0		0	1
	SIT2		test		0	0		0	1
		Voedingen	test		0	0		0	1
		Telecommunicatie	test		0	0		0	1
		Astris adapter	test		0	0		0	1
		KEV/KBV - adapter	test		0	0		0	1
		Interface aangrenzende beveiligingssytemen	test		0	0		0	1
		SIT2 testen met ingenieursbureau	test		0	0		0	1
	Integratietest en inbedrijfstellen, SIT3		test		0	0		0	1
VEILIGHEID & KWALITEIT									
	V&G		test		0	0		0	1
	GASC		test		0	0		0	1
	SASC		test		0	0		0	1
	ISA		test		0	0		0	1
	NOBO		test		0	0		0	1
	Kwaliteitsborging		test		0	0		0	1
Nazorg									
	Opleidingen en Trainingen								
Documenten en instructiehandleidingen									
	As Built documentatie		evp		0	1		0	0
	Voorschriften								
	Handleidingen								
Reserveden									
	Reserveden								
Onderhandelingsresultaat									
	Projectkorting								

Contractual Account Interlocking Changes		Process	Productmatrix for each Interlocking Engineering Process						
			Engineering vs production factor	Data Preparation	EVP Engineering	EVP Conversion	EVP Test		
Projectmanagement [Interlocking]									
	Overall IXL-projectmanagement	all	0.333333333	0.0825	0.0825		0.0825	0.0825	
	Technisch projectmanagement	all	0.333333333	0.0825	0.0825		0.0825	0.0825	
	Site projectmanagement								
	Interface (raakvlak) projectmanagement	all engineering		0.25	0.25		0.25	0.25	
	Safetymanagement	all	0.333333333	0.0825	0.0825		0.0825	0.0825	
	Qualitymanagement	all	0.333333333	0.0825	0.0825		0.0825	0.0825	
	Financieel projectmanagement	all	0.333333333	0.0825	0.0825		0.0825	0.0825	
GENERIEK SYSTEEM									
Ontwerpen generiek systeem									
	Overall systeem ontwerp								
		ontwerpen van oplossingen	evp		0	1		0	0
		afleiden van eisen	translate		1	0		0	0
		verificatie ontwerpdocumenten	translate en evp		0.5	0.5		0	0
	RAMS	test en translate		0.5	0		0	0.5	
	Bouwen generieke software								
	Lab. Test								
		IXL	test		0	0		0	1
		KEV/KBV - adapter	test		0	0		0	1
		Decentrale apparatuur	test		0	0		0	1
LOKATIESPECIFIEKE TOEPASSING									
Ontwerpen lokatiespecifieke toepassing									
	Overall systeem ontwerp								
		ontwerpen van oplossingen	evp		0	1		0	0
		afleiden van eisen	translate		1	0		0	0
		verificatie ontwerpdocumenten	translate en evp		0.5	0.5		0	0
	RAMS	test en translate		0.5	0		0	0.5	
	Data prep.								
		IXL	evp		0	1		0	0
		KEV/KBV - adapter	evp		0	1		0	0
		Decentrale apparatuur	evp		0	1		0	0
		Overige subsystemen	evp		0	1		0	0
	Lab. Test								
		IXL	test		0	0		0	1
		KEV/KBV - adapter	test		0	0		0	1
		Decentrale apparatuur	translate		1	0		0	0

		Applicatiespecifieke integratie	test		0	0	0	1
		Data test (ATZ)	test		0	0	0	1
	Ontwerp t.b.v. installatie en bedradingstekeningen		evp		0	1	0	0
	Lokatiespecifieke validatie		test		0	0	0	1
Ontwerp subsystemen								
	Gebouwen / relaishuizen							
		Conditionering						
		Civiele bouw						
		Toegangswegen						
		Verlichting						
		Klimaatregeling						
	Voedingen							
	Telecommunicatiesystemen							
Hardware ON SITE								
	IXL Centraal							
	KEV/KBV - adapter							
	Lokale bediening							
	Managementsysteem							
	IXL Decentraal							
	Input boards							
	Output Boards							
	Onderhoudstools							
	Subsysteem Voedingen							
Diensten								
Assemblage								
	Centrale interlockingsysteem							
	Decentrale interlockingdelen							
Installatie en Montage on site								
	Montage							
	Software laden							
Testen, verificatie en validatie								
	FAT		test		0	0	0	1
	SIT1		test		0	0	0	1
	SIT2		test		0	0	0	1
		Voedingen	test		0	0	0	1
		Telecommunicatie	test		0	0	0	1
		VPT	test		0	0	0	1
		Interface aangrenzende beveiligingsystemen	test		0	0	0	1
		SIT2 testen ingenieursbureau	test		0	0	0	1
	Integratietest en inbedrijfstellen, SIT3		test		0	0	0	1
VEILIGHEID & KWALITEIT								
	V&G		test		0	0	0	1
	GASC		test		0	0	0	1
	SASC		test		0	0	0	1
	ISA		test		0	0	0	1
	NOBO		test		0	0	0	1
	Kwaliteitsborging		test		0	0	0	1
Nazorg								
	Opleidingen en Trainingen							
Documenten en instructiehandleidingen								
	As Built documentatie		evp		0	1	0	0
Reservdelen								
	Reservdelen							

Contractual Account Interlocking Changes		Process	Productmatrix for each Interlocking Engineering Process				
			Data Preparation	EVP Engineering	EVP Conversion	EVP Test	
Projectmanagement [Interlocking]							
Overall IXL-projectmanagement		all	0.333333	0.0825	0.0825	0.0825	0.0825
Technisch projectmanagement		all	0.333333	0.0825	0.0825	0.0825	0.0825
Site projectmanagement							
Interface (raakvlak) projectmanagement		all engineering		0.25	0.25	0.25	0.25
Safetymanagement		all	0.333333	0.0825	0.0825	0.0825	0.0825
Qualitymanagement		all	0.333333	0.0825	0.0825	0.0825	0.0825
Financieel projectmanagement		all	0.333333	0.0825	0.0825	0.0825	0.0825
GENERIEK SYSTEEM							
Ontwerpen generiek systeem							
Overall systeem ontwerp IXL							
	ontwerpen van oplossingen	evp		0	1	0	0
	afleiden van eisen	translate		1	0	0	0
	verificatie ontwerpdocumenten	translate en evp		0.5	0.5	0	0

Overall systeem ontwerp RBC en overig							
	ontwerpen van oplossingen	evp		0	1	0	0
	afleiden van eisen	translate		1	0	0	0
	verificatie ontwerpdocumenten	translate en evp		0.5	0.5	0	0
RAMS		test en translate		0.5	0	0	0.5
Bouwen generieke software							
Lab. Test							
	IXL	test		0	0	0	1
	RBC	test		0	0	0	1
	KEV/KBV adapter	test		0	0	0	1
	GSM-R interface	test		0	0	0	1
	Decentrale systeemdelen	translate		1	0	0	0
	LEU's	test		0	0	0	1
	Balises	test		0	0	0	1
LOKATIESPECIFIEKE TOEPASSING							
Ontwerpen lokatiespecifieke toepassing							
Overall systeem ontwerp							
	ontwerpen van oplossingen	evp		0	1	0	0
	afleiden van eisen	translate		1	0	0	0
	verificatie ontwerpdocumenten	translate en evp		0.5	0.5	0	0
RAMS		test en translate		0.5	0	0	0.5
Data prep.							
	IXL	evp		0	1	0	0
	RBC	evp		0	1	0	0
	Interface GSM-R	evp		0	1	0	0
	Data prep balises	evp		0	1	0	0
	Switchable balises	evp		0	1	0	0
	fixed balises	evp		0	1	0	0
	LEU	evp		0	1	0	0
	Interface KEV/KBV - adapter	evp		0	1	0	0
	Decentrale apparatuur	evp		0	1	0	0
	Overige subsystemen	evp		0	1	0	0
Lab. Test							
	IXL	test		0	0	0	1
	RBC	test		0	0	0	1
	KEV/KBV adapter	test		0	0	0	1
	GSM-R interface	test		0	0	0	1
	Decentrale systeemdelen	test		0	0	0	1
	LEU's	test		0	0	0	1
	Balises	test		0	0	0	1
	Applicatiespecifieke integratie	test		0	0	0	1
	Data test (ATZ)	test		0	0	0	1
Ontwerp t.b.v. Installatie en bedradingstekeningen							
	Inmeten balises	translate		1	0	0	0
	S&OA bladen aanpassen	translate		1	0	0	0
	IXL - L2	projection		0	0	1	0
Lokatiespecifieke validatie		test		0	0	0	1
Ontwerp subsystemen							
Gebouwen / relaihuizen							
	Conditionering						
	Civiele bouw						
	Toegangswegen						
	Verlichting						
	Klimaatregeling						
Voedingen							
Telecommunicatiesystemen							
Hardware ON SITE							
IXL Centraal							
RBC							
Interface KEV/KBV							
Interface RBC naar GSM-R							
Telecomnetwerk							
IXL Decentraal							
Input boards							
Output Boards							
Balises							
	Fixed balise						
	Switchable balise						
LEU							
Onderhoudstools							
Subsysteem Voedingen							
Benodigde kabel-, leg- en geulkosten tbv SW balises etc.							

Stop Merk Boards							
Balise Assemblage kit							
MMI							
Juridical Recorder							
Assemblage							
Centraal interlockingsysteem							
RBC							
Decentrale interlockingdelen							
Installatie en Montage on site							
Balises en LEU's							
Plaatsen, bekabelen en IBS RBC							
Software laden							
Testen, verificatie en validatie							
FAT		test		0	0	0	1
SIT1		test		0	0	0	1
SIT2		test		0	0	0	1
	Voedingen	test		0	0	0	1
	Telecommunicatie	test		0	0	0	1
	VPT	test		0	0	0	1
	Interface aangrenzende beveiligingssystemen	test		0	0	0	1
	trein-baan intergratie (L2 testruns)	test		0	0	0	1
	IXL	test		0	0	0	1
Integratietest en inbedrijfstellen, SIT3		test		0	0	0	1
VEILIGHEID & KWALITEIT							
V&G		test		0	0	0	1
GASC		test		0	0	0	1
	GASC IXL	test		0	0	0	1
	GASC RBC	test		0	0	0	1
	GASC overige subsystemen	test		0	0	0	1
SASC		test		0	0	0	1
ISA		test		0	0	0	1
NOBO		test		0	0	0	1
Kwaliteitsborging		test		0	0	0	1
Nazorg							
Opleidingen en Trainingen							
Documenten en instructiehandleidingen							
As Built documentatie		evp		0	1	0	0
OBE bladen update ETCS		translate		1	0	0	0
Reservedelen							
Reservedelen							

Data Analysis

The result of the previous section contains the costs and times per account (project management, project control operation etc.) for each of the four processes. The next shows the descriptive time and cost results and the results of computations to derive productivity and utilization rates.

The analysis includes:

- Descriptive analysis of monetary data
 - Proportions of costs per process step
 - In general
 - Split per project type
 - Split per corridor
- Statistical analysis of monetary data
 - Activity cost allocation
 - Differences between project types
 - Differences between corridors
 - Differences between process steps
 - Relation between track elements and costs
- Descriptive analysis of time data
 - Proportions of time per process step
 - In general
 - Split per project type
 - Split per corridor
- Statistical analysis of time data
 - Activity time allocation

- Differences between project types
- Differences between corridors
- Differences between process steps
 - Relation between track elements and time
- Productivity
 - Split per project type
 - Split per corridor
- Utilization rate
 - Split per project type
 - Split per corridor
- Productivity versus utilization

Cost Results

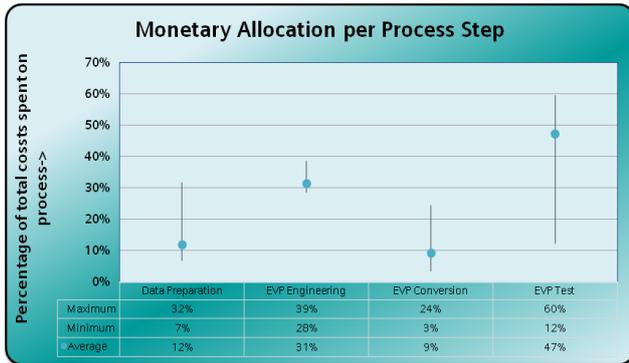


Figure 115: The distribution of total costs for each of the four processes. Each distribution contains 13 cases of which 4 new projects, 3 small interlocking modification projects, 3 big interlocking modification projects and 3 ETCS related interlocking projects. The percentages reflect each process' cost share of the total process cost per project. The dots present averages, the line ends minimum and maximum values.

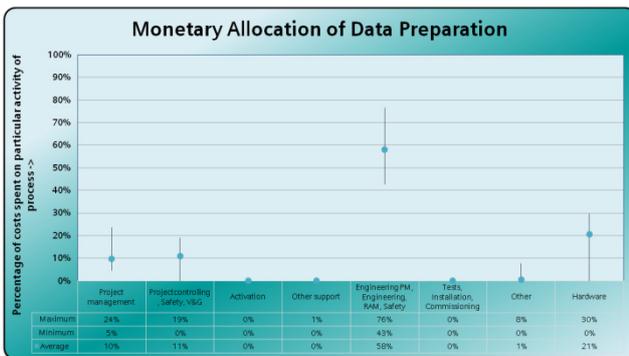


Figure 116: Shows the relative cost structure of the data preparation process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

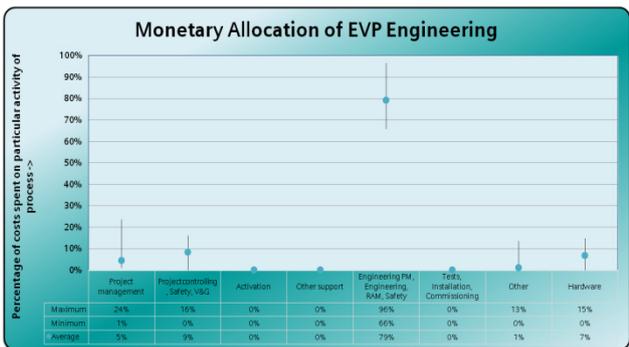


Figure 117: Shows the relative cost structure of the EVP engineering process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

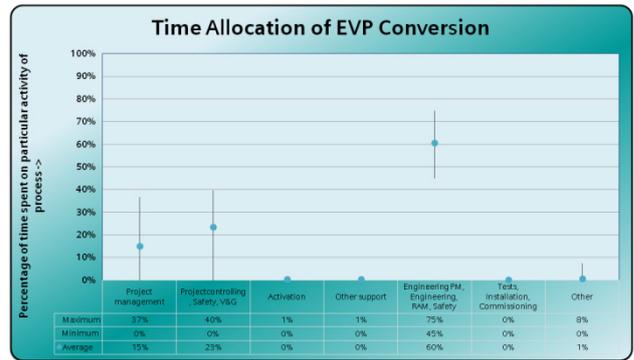


Figure 118: Shows the relative cost structure of the EVP conversion for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

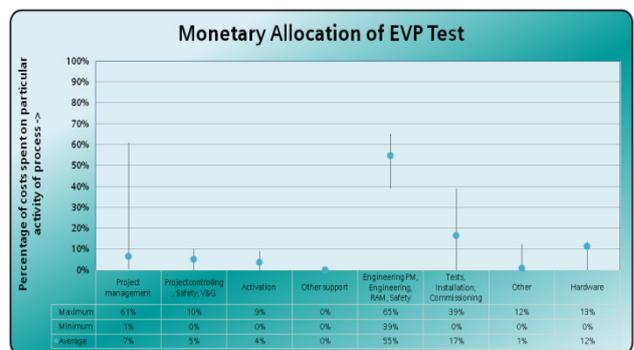


Figure 119: Shows the relative cost structure of the EVP testing process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

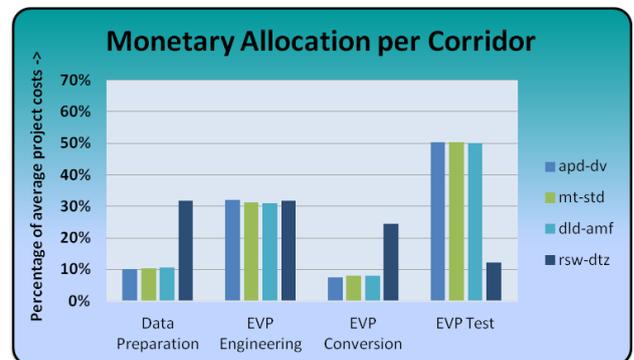


Figure 120: Shows the average relative cost contributions to each of the four interlocking engineering processes per project corridor investigated. The Delft tunnel project clearly deviates from the Mistral corridors. This is mainly caused because the Delft project corridor investigated. The Delft tunnel project clearly deviates from the Mistral corridors. This is mainly caused because the Delft project outsources a lot of testing activities to engineering agencies.

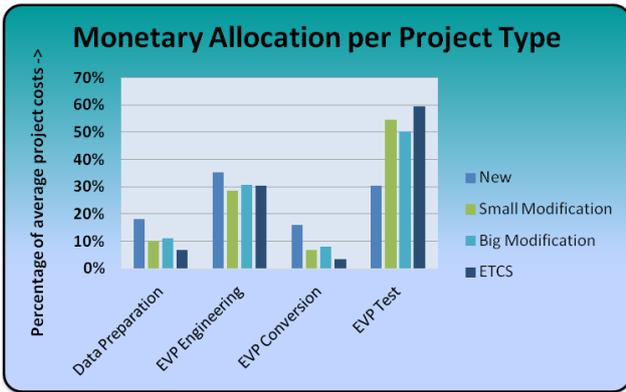


Figure 121: Shows the average relative cost contributions to each of the four interlocking engineering processes per project type investigated. New interlocking projects clearly spent more time on collecting all required data. The other three project types can probably rely on existing data

Time Results

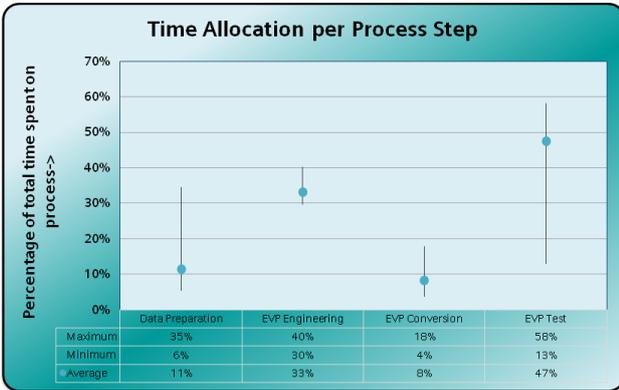


Figure 122: The distribution of total time for each of the four processes. Each distribution contains 13 cases of which 4 new projects, 3 small interlocking modification projects, 3 big interlocking change projects and 3 ETCS related interlocking projects. The percentages reflect each process' time share of the total process time per project. The dots present averages, the line ends minimum and maximum values.

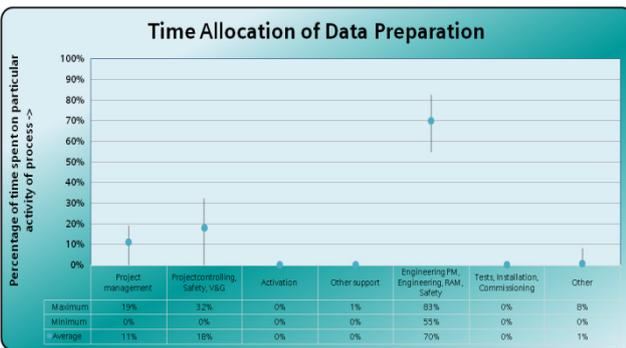


Figure 123: Shows the relative time structure of the data preparation process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

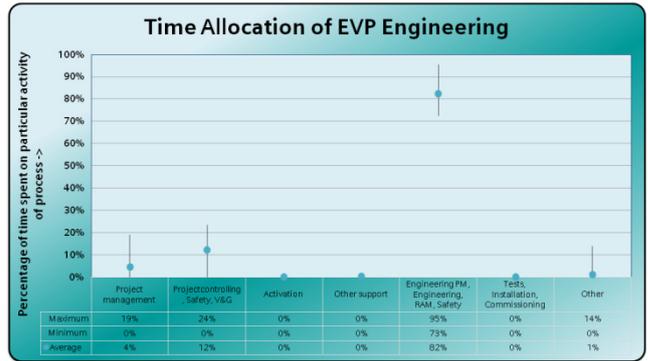


Figure 124: Shows the relative time structure of the EVP engineering process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

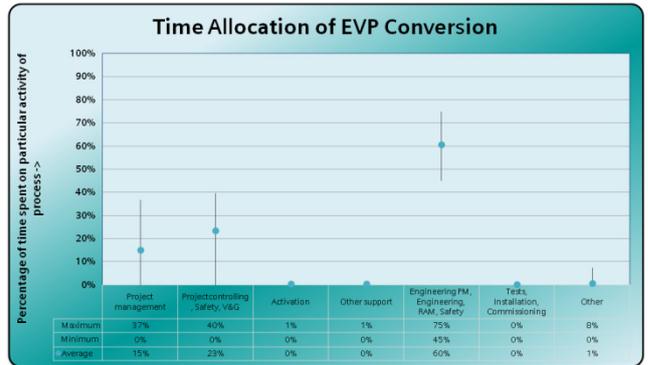


Figure 125: Shows the relative time structure of the EVP conversion for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

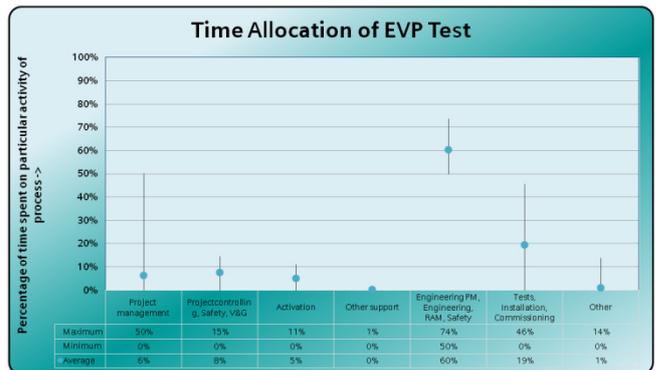


Figure 126: Shows the relative time structure of the EVP testing process for eight accounts: project management, project control, operation, other support, engineering, testing, other and hardware. The dots present averages, the line ends minimum and maximum values.

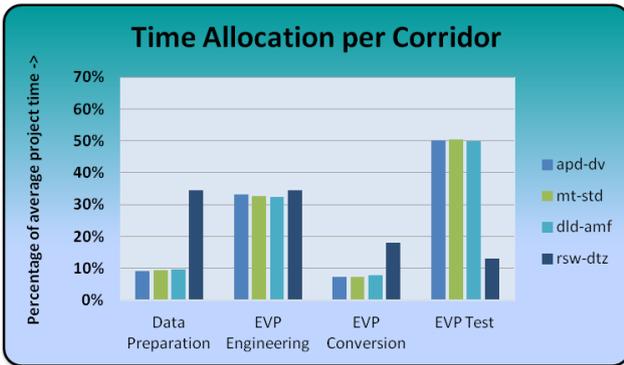


Figure 127: Shows the average relative time contributions to each of the four interlocking engineering processes per project corridor investigated. Again, the Delft tunnel project clearly deviates from the Mistral corridors. This is mainly caused because the Delft tunnel project clearly deviates from the Mistral corridors. This is mainly caused because the Delft project outsources a lot of testing activities to engineering agencies.

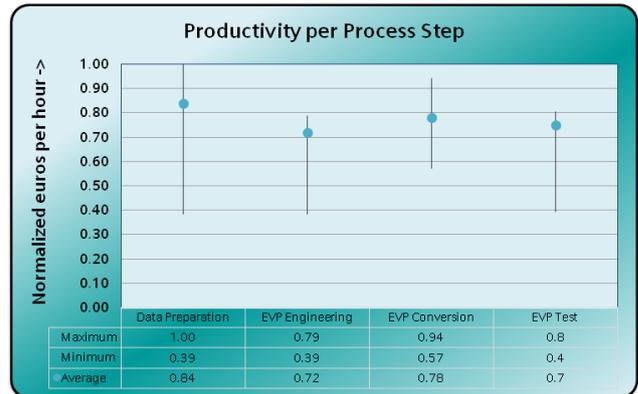


Figure 129: The distribution of productivity rates of different projects for each of the four interlocking engineering processes at Siemens. The lower productivity rates correspond to the Delft tunnel project. The higher productivity rates mostly correspond to the Delft tunnel project. The higher productivity rates mostly correspond to the interlocking modification projects of the Mistral corridors. The dots present averages, the line ends minimum and maximum values.

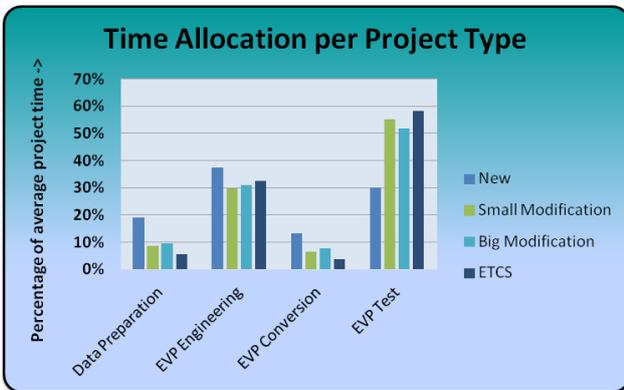


Figure 128: Shows the average relative time contributions to each of the four interlocking engineering processes per project type investigated. New interlocking projects clearly spent more time on collecting all required data. The other three project types can probably rely on existing data.

Productivity

The amount of value created per time step assesses the productivity of the interlocking engineering processes at Siemens. For that purpose, the amount of euros per activity are divided by the required process hours. For reasons of confidentiality, the actual figures are normalized. The actual productivity values are divided by the maximum productivity value of all activities or of all process steps in case of a comparison in totals. The next presents the general results and the results per corridor and project type.

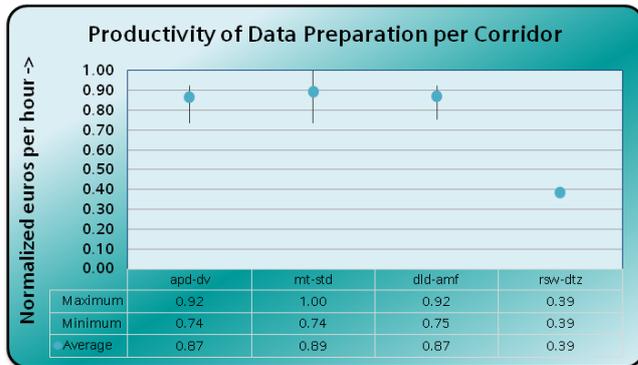


Figure 130: The data preparation productivity rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

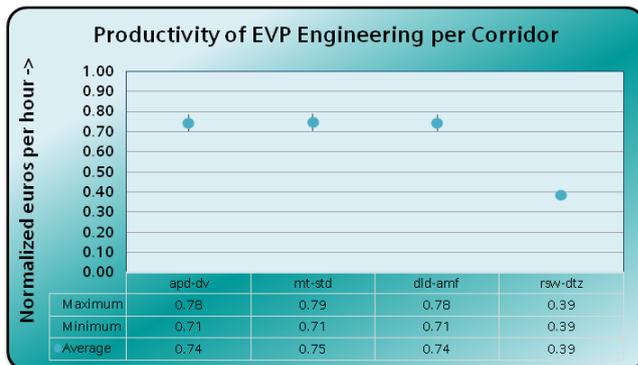


Figure 131: The EVP engineering productivity rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

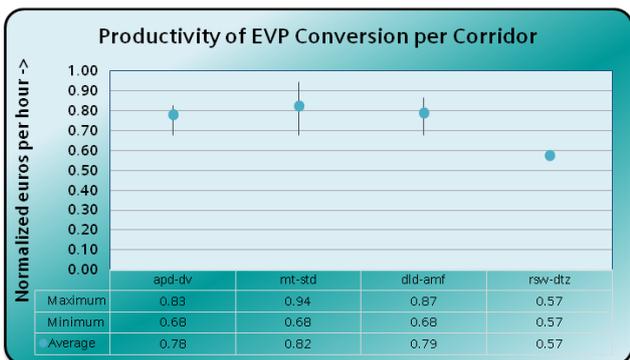


Figure 132: The EVP conversion productivity rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

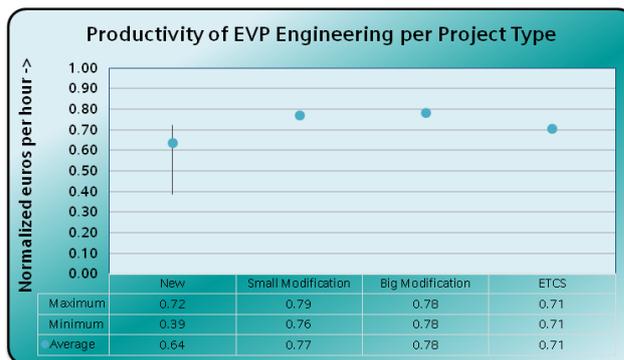


Figure 135: The EVP engineering productivity rate distribution decomposed per project type. Remarkable is the considerable variety at new interlocking projects caused by the low productivity rate of the Delft project. The dots present averages, the line ends minimum and maximum values.

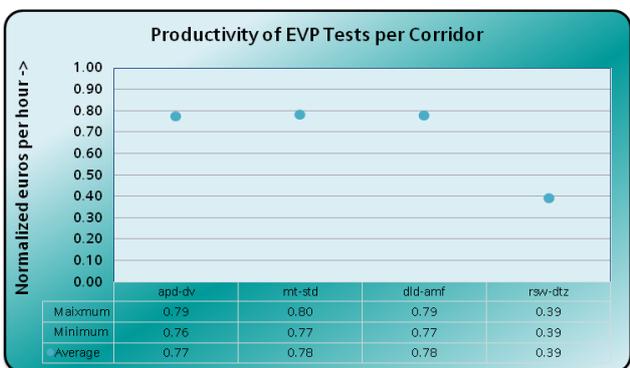


Figure 133: The EVP test productivity rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

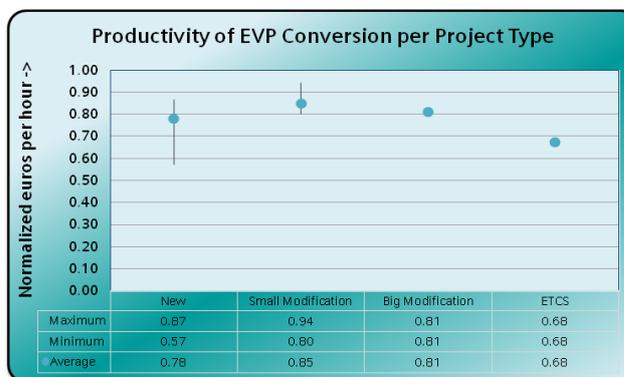


Figure 136: The EVP conversion productivity rate distribution decomposed per project type. Remarkable is the considerable variety at new interlocking projects caused by the low productivity rate of the Delft project. The dots present averages, the line ends minimum and maximum values.

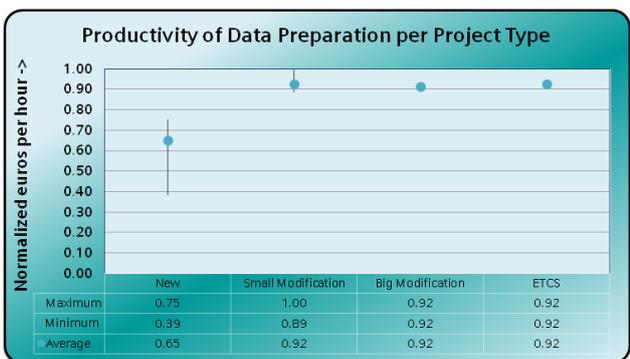


Figure 134: The data preparation productivity rate distribution decomposed per project type. Remarkable is the considerable variety at new interlocking projects caused by the low productivity rate of the Delft project. The dots present averages, the line ends minimum and maximum values.

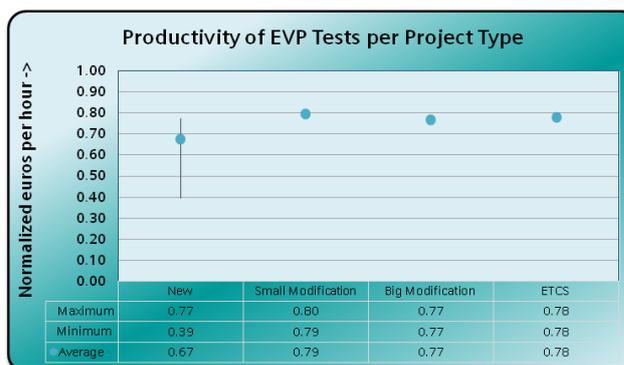


Figure 137: The EVP testing productivity rate distribution decomposed per project type. Remarkable is the considerable variety at new interlocking projects caused by the low productivity rate of the Delft project. The dots present averages, the line ends minimum and maximum values.

Resource Utilization

The ability to transform input into output per time step defines resource utilization. Manhours are the most important resource in this engineering process. Track elements indicates the transformation size. Therefore, the amount of track elements processes per manhour defines the resource utilization. The next presents the general results and the results per corridor and project type.

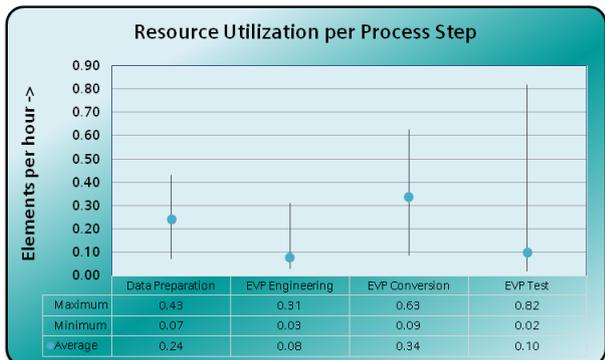


Figure 138: The distribution of resource utilization rates per engineering process. Most high utilization rates belong to the big change interlocking projects, ETCS interlocking projects and/or the Delft tunnel project. Lower utilization rates mostly cohere to small interlocking modification projects. The dots present averages, the line ends minimum and maximum values.

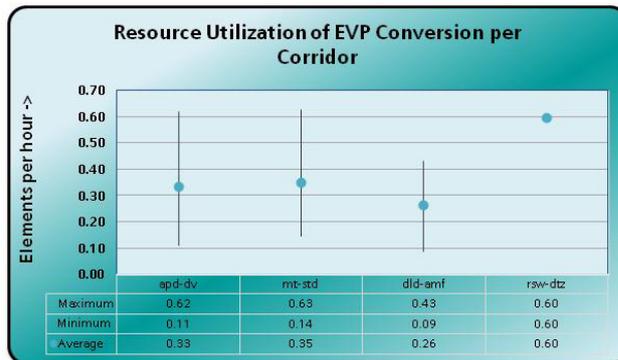


Figure 141: The EVP conversion resource utilization rate distribution decomposed per corridor. Big change and ETCS projects cause upward utilization variety; small change and new projects cause downward utilization variety. The dots present averages, the line ends minimum and maximum values.

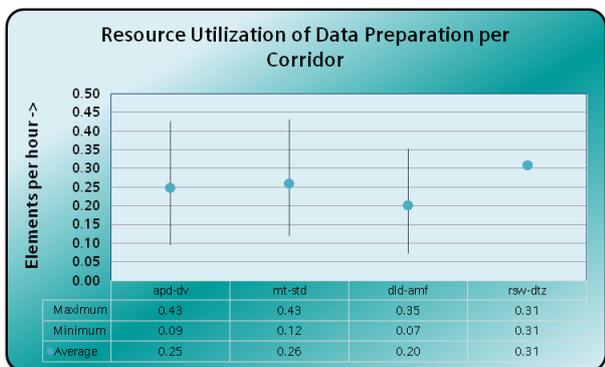


Figure 139: The data preparation resource utilization rate distribution decomposed per corridor. Big change and ETCS projects cause upward utilization variety; small change and new projects cause downward utilization variety. The dots present averages, the line ends minimum and maximum values.

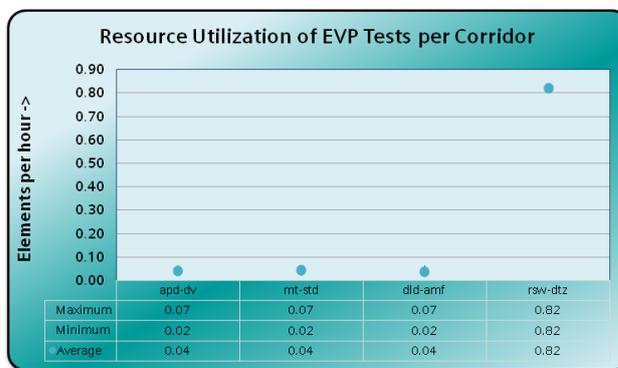


Figure 142: The EVP testing resource utilization rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

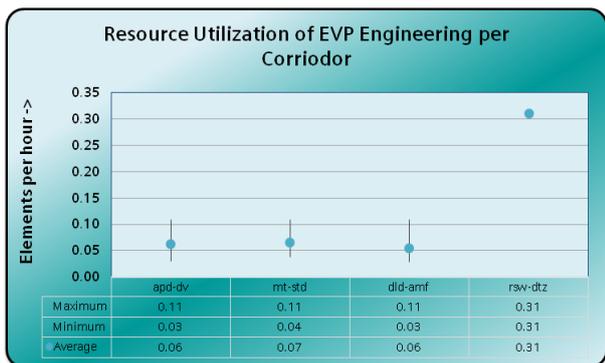


Figure 140: The EVP engineering resource utilization rate distribution decomposed per corridor. The dots present averages, the line ends minimum and maximum values.

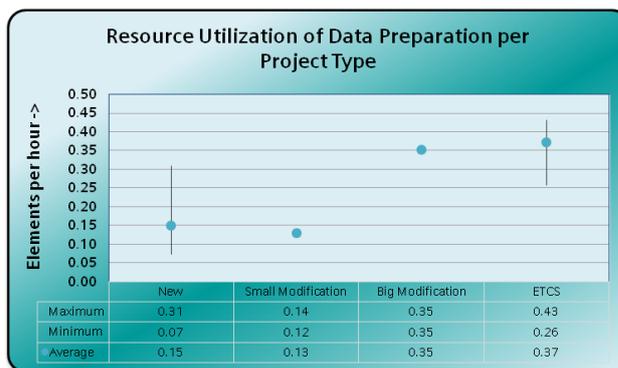


Figure 143: The data preparation resource utilization rate distribution decomposed per project type. The dots present averages, the line ends minimum and maximum values.

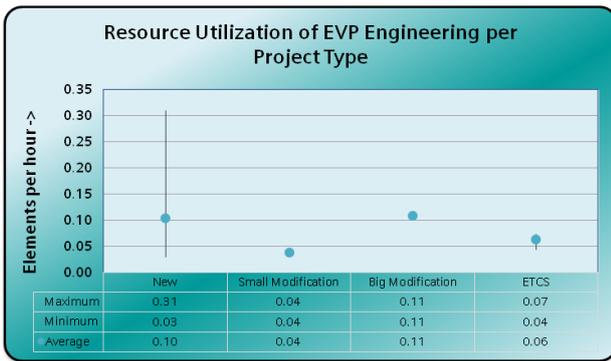


Figure 144: The EVP engineering resource utilization rate distribution decomposed per project type. The dots present averages, the line ends minimum and maximum values.

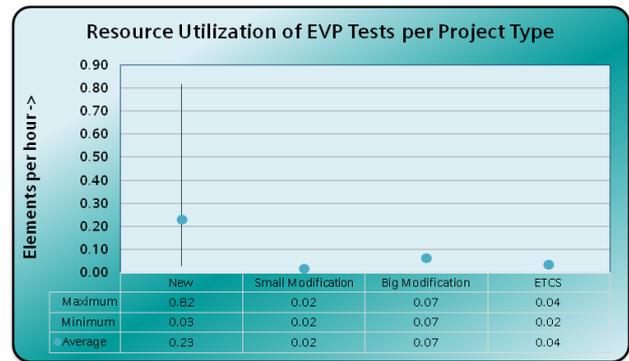


Figure 146: The EVP testing resource utilization rate distribution decomposed per project type. The dots present averages, the line ends minimum and maximum values.

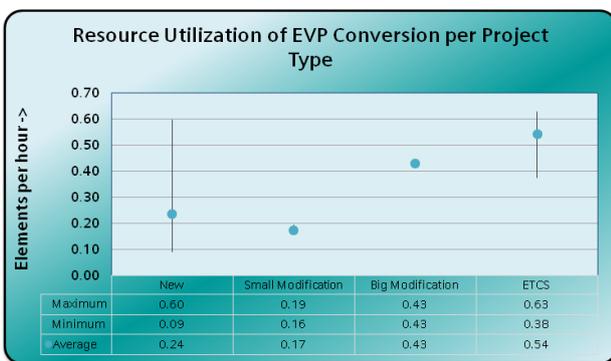


Figure 145: The EVP conversion resource utilization rate distribution decomposed per project type. The dots present averages, the line ends minimum and maximum values.

Statistical Analyses

Various statistical tests compare the time and money distributions to find how variables relate to each other. First, a Kolmogorov-Smirnov test is performed. Then, a Kruskal-Wallis test compares various distributions. Last, a correlation test between cost accounts, total costs and the amount of track elements assesses direct effects of variable changes. Each test uses a double-tail significance level of 5%.

Kolmogorov-Smirnov test

A vital issue concerns the use of a parametric or non-parametric test approach. Since the amount of data is limited, a serious concern exists whether the data approximates a normal distribution: a prerequisite of a parametric test approach. The Kolmogorov-Smirnov test provides certainty.

The Kolmogorov-Smirnov tests the next hypothesis:

The random sample distribution of a cost account equals the normal distribution

One-Sample Kolmogorov-Smirnov Test for Monetary values

		Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware	Total
N		52	52	52	52	52	52	52	52	52
	Positive	.402	.347	.427	.315	.498	.514	.321	.309	
	Negative	-.258	-.209	-.285	-.239	-.369	-.409	-.257	-.247	
	Kolmogorov-Smirnov Z	2.896	2.502	2.741	3.078	2.270	3.591	3.708	2.313	2.225
	Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000	.000

One-Sample Kolmogorov-Smirnov Test for Time values

		Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Total
N		52	52	52	52	52	52	52	52
Most Extreme	Positive	.358	.345	.374	.317	.498	.514		
Differences	Negative	-.241	-.206	-.337	-.237	-.369	-.409		
	Kolmogorov-Smirnov Z	2.585	2.487	2.699	3.070	2.283	3.591	3.708	2.401
	Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.000

For reasons of confidentiality, the absolute distribution characteristics cannot be presented. The above tables provide sufficient data to reject the hypothesis since each significance level is lower than 5%. In other words, the cost and time account distribution does not match the normal distribution and further statistical analysis is limited to non-parametric tests. The main disadvantage of non-parametric tests is their lower reliability.

Kruskal-Wallis Test

The Kruskal-Wallis test can assess whether the distribution of a cost account is the same for various subgroups. For example: is the engineering cost distribution the same for new project types as for ETCS interlocking projects.

The Kruskal-Wallis test investigates the following hypothesis:

The x random samples belong to the same population.

The four random samples of the project types (new, small change, big change, ETCS), process steps (data preparation, EVP engineering, EVP conversion and EVP test) and corridors (dld-amf, dv-apd, mt-std and rsw-dtz).

The Kruskal-Wallis test provides the next results when investigating the random samples of **project types**:

Test Statistics for Monetary values^{a,b}

	Total	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware
Chi-Square	24.481	49.759	20.609	.889	22.518	27.344	.357	9.543	6.140
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.000	.000	.000	.828	.000	.000	.949	.023	.105

a. Kruskal Wallis Test
b. Grouping Variable: Project Type

Test Statistics for Time values^{a,b}

	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware	Total
Chi-Square	49.337	20.609	.889	22.518	33.136	.357	9.543	.000	29.794
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.000	.000	.828	.000	.000	.949	.023	1.000	.000

a. Kruskal Wallis Test
b. Grouping Variable: Project Type

The accounts that have a significance value <0.05 reject the test's hypothesis and do not have equal distributions for each of the project types. This implies that the operation account, test account and hardware account (only in case of monetary distributions) only have equal cost distributions when compared between two or more project

types. The total accounts equals the sum of all accounts per process step, per corridor, per project type.

The Kruskal-Wallis test provides the next results when investigating the random samples of **process steps**:

Test Statistics for Monetary values^{a,b}

	Total	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware
Chi-Square	12.848	.085	3.334	39.024	3.689	9.390	45.009	.008	12.114
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.005	.994	.343	.000	.297	.025	.000	1.000	.007

a. Kruskal Wallis Test
b. Grouping Variable: Process steps

Test Statistics for Time values^{a,b}

	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware	Total
Chi-Square	.084	3.334	39.024	3.689	8.892	45.009	.008	.000	10.763
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.994	.343	.000	.297	.031	.000	1.000	1.000	.013

a. Kruskal Wallis Test
b. Grouping Variable: Process steps

The accounts that have a significance value <0.05 reject the test's hypothesis and do not have equal distributions for each of the four process steps. This implies that the management account, project control account, other support account, other account (although it actually contains no costs/hours) and hardware account (only in case of

monetary distributions) have equal cost distributions when compared between two or more process steps.

The Kruskal-Wallis test provides the next results when investigating the random samples of **corridors**:

Test Statistics Monetary values^{a,b}

	Total	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware
Chi-Square	3.097	2.837	10.991	1.661	1.691	3.473	1.251	9.543	8.401
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.377	.417	.012	.646	.639	.324	.741	.023	.038

a. Kruskal Wallis Test
b. Grouping Variable: Corridor

Test Statistics Time values^{a,b}

	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware	Total
Chi-Square	2.921	10.991	1.661	1.691	.707	1.251	9.543	.000	1.454
df	3	3	3	3	3	3	3	3	3
Asymp. Sig.	.404	.012	.646	.639	.871	.741	.023	1.000	.693

a. Kruskal Wallis Test
b. Grouping Variable: Corridor

The accounts that have a significance value <0,05 reject the test's hypothesis and do not have equal distributions for each of the four corridors. This implies that the total account, management account, operation control account, other support account, engineering account and testing account have equal cost distributions when compared between two or more corridors.

Spearman's Correlation Test

The Spearman test is the non-parametric variant of a standard correlation test. Before a Spearman test may be executed, there need to be arguments that the distributions do have a linear relation. For that purpose, a scatterplot investigates the linearity of the cost and time data points.

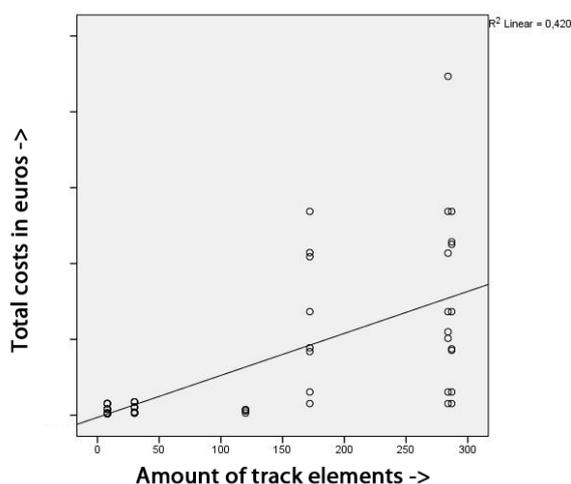


Figure 147: Shows the relation between the amount of track elements and the total costs for each of the four engineering processes, per project type and per corridor. Costs are in euros.

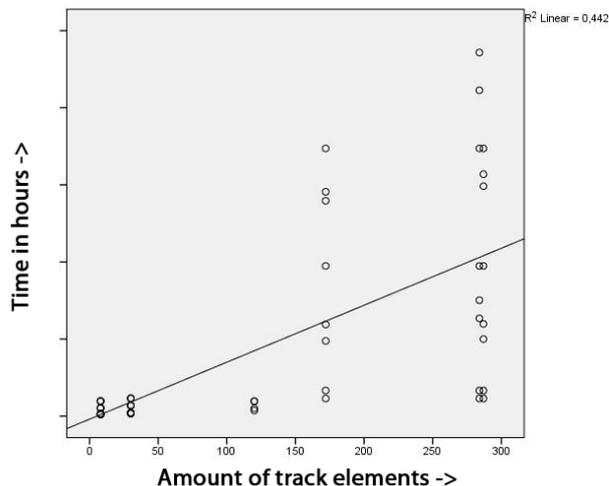


Figure 148: Shows the relation between the amount of track elements and the total time for each of the four engineering processes, per project type and per corridor. Time is in minutes.

Both scatterplots provide sufficient reason to believe that the relation between costs / hours and the amount of track elements (the process transformation elements as described earlier) is linear: the data points on each side of the line keep each other in balance. Furthermore, there is no strong reason to assume another distribution. An exponential distribution for example would have too many data points at high track element amounts with low total costs.

The next two tables present the Spearman correlation test results for monetary and time values respectively.

Spearman Correlations for Monetary values

		#Elements	Total	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Hardware
Spearman's rho	#Elements	1.000	.743	.853**	.743	.153**	.599**	.777	.060**	.274**	.362
	Total	.737**	1.000**	.675**	.895**	.520	.743**	.978**	.471	.341**	.686**
	Managements	.853**	.675**	1.000	.689**	.197**	.690	.700**	.054**	.393	.412**
	PCSVG	.743**	.895**	.689**	1.000**	.355**	.705**	.928**	.236**	.389**	.648**
Operation	.153	.520	.197	.355	1.000**	.496	.391	.918**	.135	.651	
OtherSupport	.599**	.743**	.690**	.705**	.496**	1.000**	.686**	.382**	.412**	.777**	

ERS	Correlation Coefficient	.777**	.978**	.700**	.928**	.391**	.686**	1.000**	.337**	.351**	.585**
	Sig. (2-tailed)	.000	.000	.000	.000	.004	.000	.	.015	.011	.000
	N	52	52	52	52	52	52	52	52	52	52
TIC	Correlation Coefficient	.060	.476	.054	.236	.918**	.382	.337	1.000**	.033	.547
	Sig. (2-tailed)	.671	.000	.704	.093	.000	.005	.015	.	.817	.000
	N	52	52	52	52	52	52	52	52	52	52
Other	Correlation Coefficient	.274*	.341*	.393**	.389*	.135*	.412**	.351*	.033*	1.000**	.339*
	Sig. (2-tailed)	.049	.013	.004	.004	.340	.002	.011	.817	.	.014
	N	52	52	52	52	52	52	52	52	52	52
Hardware	Correlation Coefficient	.362**	.686**	.412**	.648**	.651**	.777**	.585**	.547**	.339**	1.000**
	Sig. (2-tailed)	.008	.000	.002	.000	.000	.000	.000	.000	.014	.
	N	52	52	52	52	52	52	52	52	52	52

** Correlation is significant at the 0.01 level (2-tailed). * Correlation is significant at the 0.05 level (2-tailed).

Correlations

		#elements	Managements	PCSVG	Operation	OtherSupport	ERS	TIC	Other	Total
#elements	Correlation Coefficient	1.000	.153	.599**	.813**	.599**	.813**	.060	.274**	.793
	Sig. (2-tailed)	.	.279	.000	.000	.000	.000	.671	.049	.000
	N	52	52	52	52	52	52	52	52	52
Managements	Correlation Coefficient	.849**	.196**	.687	.765**	.687	.765**	.054**	.392	.746**
	Sig. (2-tailed)	.000	.163	.000	.000	.000	.000	.706	.004	.000
	N	52	52	52	52	52	52	52	52	52
PCSVG	Correlation Coefficient	.743**	.355**	.705**	.883	.705**	.883	.236**	.394**	.888**
	Sig. (2-tailed)	.000	.010	.000	.000	.000	.000	.093	.004	.000
	N	52	52	52	52	52	52	52	52	52
Operation	Correlation Coefficient	.153	1.000	.496	.359**	.496	.359**	.918	.135	.491
	Sig. (2-tailed)	.279	.	.000	.009	.000	.009	.000	.340	.000
	N	52	52	52	52	52	52	52	52	52
OtherSupport	Correlation Coefficient	.599**	.496**	1.000**	.686**	1.000**	.686**	.382**	.412**	.740**
	Sig. (2-tailed)	.000	.000	.	.000	.	.000	.005	.002	.000
	N	52	52	52	52	52	52	52	52	52
ERS	Correlation Coefficient	.813**	.359**	.686**	1.000**	.686**	1.000**	.305**	.351**	.980**
	Sig. (2-tailed)	.000	.009	.000	.	.000	.	.028	.011	.000
	N	52	52	52	52	52	52	52	52	52
TIC	Correlation Coefficient	.060	.918	.382	.305	.382	.305	1.000	.033	.438
	Sig. (2-tailed)	.671	.000	.005	.028	.005	.028	.	.817	.001
	N	52	52	52	52	52	52	52	52	52
Other	Correlation Coefficient	.274*	.135*	.412**	.351**	.412**	.351**	.033*	1.000**	.370*
	Sig. (2-tailed)	.049	.340	.002	.011	.002	.011	.817	.	.007
	N	52	52	52	52	52	52	52	52	52
Hardware	Correlation Coefficient
	Sig. (2-tailed)
	N	52	52	52	52	52	52	52	52	52
Total	Correlation Coefficient	.793**	.491**	.740**	.980**	.740**	.980**	.438**	.370**	1.000**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.001	.007	.
	N	52	52	52	52	52	52	52	52	52

** Correlation is significant at the 0.01 level (2-tailed). * Correlation is significant at the 0.05 level (2-tailed).

The correlation value indicates what happens with the column value when the row variable increases by 1 unit. This leads to insight in a positive or negative relation between two variables. It does however not provide insight into causality (which variable changes first). Furthermore, when a significance value is below 0.05; the correlation may be assumed.

Complexity Analysis

The amount of interfaces is a way to measure the complexity of Siemens’ interlocking engineering. A process’ amount of interfaces defined by the amount of relations it has with previous, succeeding and its own processes. An N2 diagram reveals the amount of relations in a transparent way; the next table shows the N2 diagram for the four interlocking engineering processes at Siemens. The last process before Siemens starts engineering, SWOD, and the first process after Siemens finishes the design, the dry test, are included to express inter party relations in the value chain.

Figure 39 shows that the data preparation, EVP engineering and EVP test processes each have a feedback loop to the preceding process to validate the result. The EVP conversion does not have such a feedback loop to, in this case, the EVP engineering process. Developing the test interlocking machine does not include a mean to validate the EVP, the subsequent testing process takes care of that. The dry test at the infrastructure manager could in extreme cases conclude that certain elements for example miss or are wrongly positioned. In such case, the engineering process at Siemens needs to adjust the design from the data preparation process.

Appendix C describes how the total of existing interfaces can indicate the system coupling level index: a measure of process complexity.

Table 39: The N2 diagram related to the interlocking engineering stage.

Value Chain Process	SWOD	Data preparation	EVP engineering	EVP conversion	EVP test	Dry test	Production	TEI	TPI	SCLI
Data preparation	1	1	1					3	21	0.143
EVP engineering		1	1	1				3		0.143
EVP conversion				1	1			2		0.095
EVP test			1	1	1	1		4		0.190
Dry test		1	1	1	1	1	1	6		0.285

Appendix F - Interview Summaries

Titel

Interview at Siemens Braunschweig

Date

April 23th 2013

Location

Siemens, Ackerstr. 22, Braunschweig, Germany

Duration

09:00 – 12:00

Interviewee

Mathias Muehlhause –

Rail IT expert at Siemens Braunschweig, Germany BU Rail Automation

Bernd Wisotzki –

Rail expert at Siemens Braunschweig, Germany BU Rail Automation

Interviewer

Mark Bosschaart

Bob Janssen – Siemens Netherlands BU Rail Automation

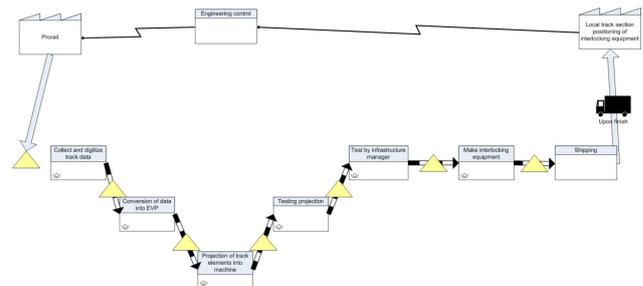
Mathias supports the development of RailML by enriching the RailML wiki, which is currently considered the manual of RailML. Furthermore, he is a member of the RailML interlocking development team. This team also contains specialists from parties like SBB, Thales, Alstom, Infrabel and so on. Mathias again notices the fact that RailML currently only exchanges infrastructure, timetable and rolling stock data. Interlocking, or consequent routes are not yet present in the RailML architecture.

Mathias continues by explaining that an agreement has been reached on the interlocking object relations (UML) at the last RailML congress. The interlocking team continues in June at Siemens Berlin.

Bernd discusses some of the rationales behind RailML. He mentions the INESS report and stresses that although he believes RailML is a good way forward, INESS is merely based on UMLs and not the real interlocking process. In addition, Deutsche Bahn currently uses an XML alternative with BahnPro. The RailML support of DB can be a catalyst for RailML. BahnPro is however not a competitor of RailML.

Bob provides a concise explanation on the RailML assignment of Prorail. He mentions that a lot of documents are transferred by non machine readable files instead of machine readable files. The consequence is manual translation with a large chance of errors. In addition, OS maps, again not machine readable, need to be checked for all driving possibilities. Which on its turn is a quite cumbersome procedure. Especially because InfraAtlas is not designed for signal technology.

Bob asks how the interlocking process actually works out. Bernd gives an elaborated talk that results in the following value chain:

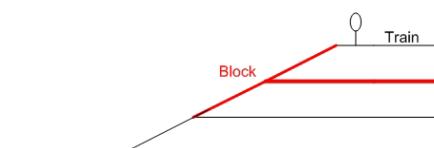


An EVP means Element VerbindungsPlan, which is the logic how Siemens connects rail elements on a given interlocking area / topology.

Bernd continues by explaining that not just the translation of non machine readable to machine readable data takes a lot of time, also the testing and validation processes do. Just 30% of all costs is related to interlocking hardware. In addition, interlocking systems of Siemens are completely made by hand which makes it more expensive over competitors. This is why Siemens strongly desires to reduce the costs related to engineering.

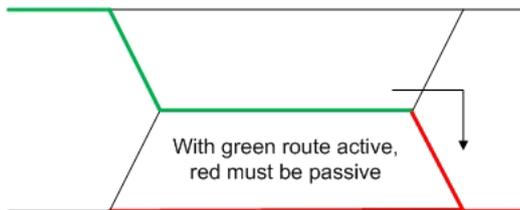
Bernd then continues a way in which interlocking projections could be made more concise. This procedure is called the spurplanstelle. A spurplanstelle does not require all routes in the interlocking area. When a route that is not defined is requested, combinations of routes lead to the requested route.

Bernd and Mathias start a discussion on ‘automation’ of the interlocking engineering process by RailML. First of all, there needs to be a decent tool to check the consistency of all rail elements in the XML. That includes whether or not all data from the infrastructure provider is currently correct. Then, there is the question of special interlocking situations. In Germany for example, having to merging parallel tracks, you would typically block that switch when a train needs to stop close to it at the double track section. Bernd and Mathias thought these situations are too specific for RailML to be modeled and would therefore need alternative logic.



Another more common German interlocking situation --- cannot be described algebraically. This e.g. occurs when a

double track section gets a middle track for some distance thus having three tracks there. When in Germany a train wants to enter that middle track, a parallel movement in the same direction may not occur. This situation is therefore even impossible to describe in RailML.



Mathias and Bernd therefore conclude on this part that about 80% can be done by IT and 20% of the interlocking engineering still requires people. The biggest advantage comes from a complete test of the entire interlocking area. Bernd thus stresses that one should not have the illusion that RailML can process an interlocking just with the push of a button.

Then Bernd continues to explain the development history of Siemens' interlocking system for the Netherlands: SIMIS W. The SIMIS W for the Netherlands is experienced about equal to the Belgium version. The projection of interlocking systems differs in both countries. In Belgium, Infrabel manages the projection with some help from Siemens and Alstom to map interlocking areas. Prorail does not have the knowledge at all and outsources the mapping process completely. Engineering agencies like Movares take care of that process. More interesting is the fact that they do not have the knowledge of ETCS which is why they try to prevent or slowdown the implementation of ETCS in the Netherlands. Parties like Siemens, Alstom and Bombardier do have the knowledge of ETCS.

At the end a little discussion is started how ETCS level 3 could be implemented as Prorail desires this option in longterm. Since real moving blocks have the disadvantage that all types of rolling stock must be identifiable, cargo trains become too much of a challenge. This is why Siemens currently thinks of alternative strategies.

Titel

Interview at Prorail Safety Systems

Date

April 22th 2013

Location

Prorail, De Inktpot, Utrecht

Duration

13:00 – 15:30

Interviewee

Jan Storck – Interlocking expert at Prorail Train Safety.
Sander Dragt – Train safety system expert at Prorail Train Safety.

Interviewer

Mark Bosschaart

Mark explains that his main goal for this meeting is to identify what processes take place if Prorail wants to develop a new or replace a current interlocking system.

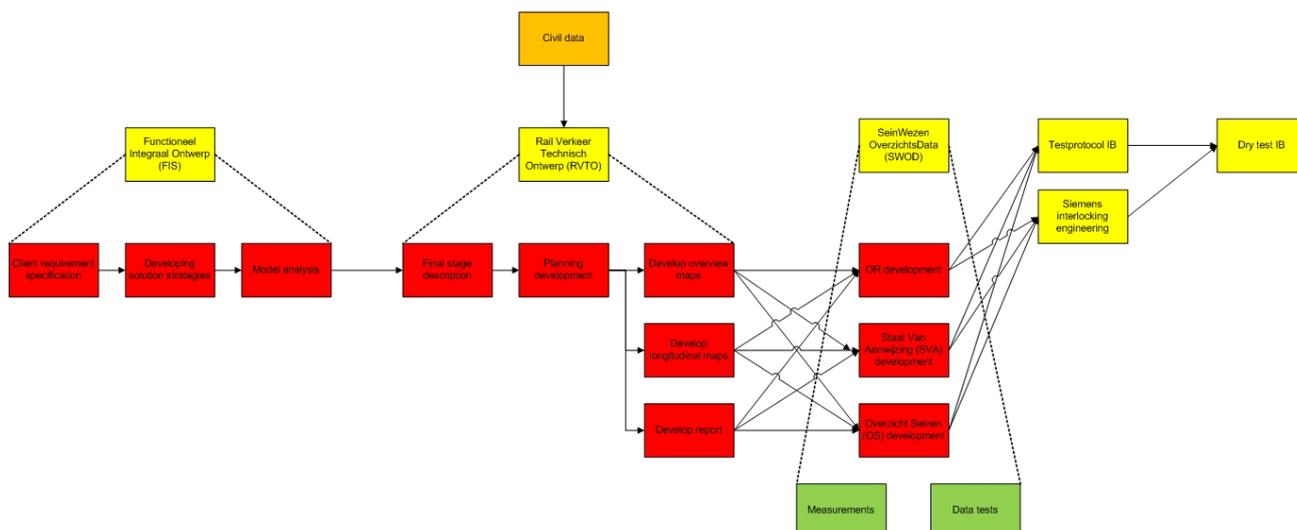
We use post-its to define all processes and put them together. After some structuring, the chain becomes like the next:

Mark asks for the main description of the processes:

FIS describes which requirements an interlocking area must stick to and which alternative is chosen to comply to those requirements. The CRS formulates those requirements, the solution alternatives develop and test those alternatives and the model analysis elaborates on the most sufficient design.

The RVTO concerns the first detailed design process focused on operational requirements. This stage requires civil data as input to start. The first step elaborates further on the model analysis of FIS having the civil data. Then a planning is developed to execute the remainder of the interlocking engineering including production and installation. The third stage in RVTO is a simultaneous process of developing overview maps, longitudinal maps and a report to make sure that every important aspect of rail operation is included in this traffic operational design. An example is for instance traction and the amount of tracks.

The SWOD is meant to enrich the maps of RVTO such that an interlocking be built. This involves a simultaneous process of developing overview return maps, maps that contain all rail elements in the interlocking area (OBE) and maps that contain all possible routes with consequent signal positions. In order to finish a SWOD, much data is derived and validated by on sight measurements and tests. The result of the SWOD goes to industry, a.o. Siemens, to develop an interlocking. When the interlocking is finished on paper, Prorail performs a dry tests to confirm the quality. Then, industry will actually produce the interlocking.

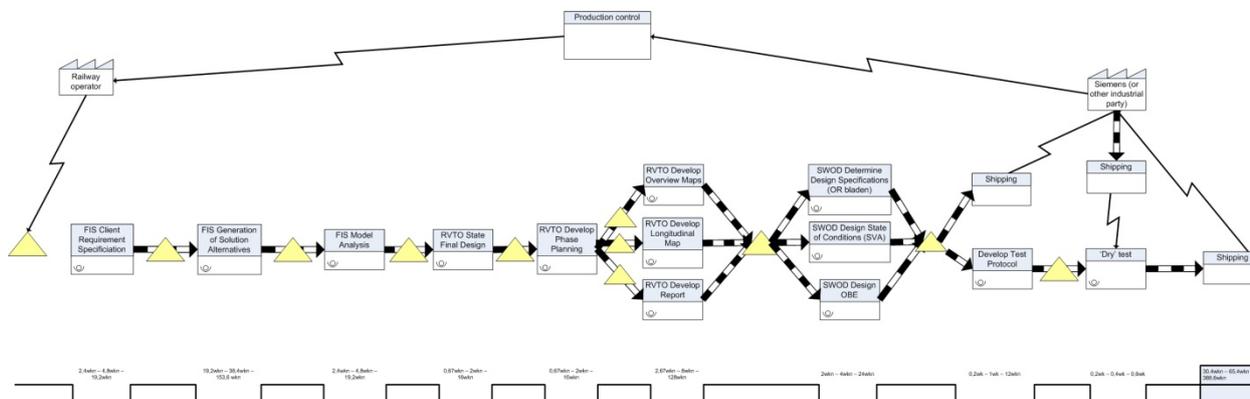


Mark asks if the gentlemen could provide some indications of lead times.

This question leads, with the previous question, to a first value chain map:

Additional explanations of Sander and Jan:

- Almost every process results in non machine readable files instead of machine readable ones which decreases precision and data reliability
- Jan believes that most files are non machine readable because engineering agencies are afraid to lose orders
- Employees at Prorail work on multiple projects at the same time: implies that projects need to wait before being processed



Titel**Interview at Prorail Rail Technology Projects****Date**May 23th 2013**Location**

Prorail, Arthur van Schendelstraat 670, Utrecht

Duration

14:00 – 15:30

Interviewee

Henrik Koelewijn – Rail expert at Prorail Projects, department of rail technology. Work area focuses on RVTO, SWOD and final tests.

Wim de Rijk – Rail expert at Prorail Projects, department of rail technology. Work area focuses on validation of FIS and RVTO.

Sander Dragt – Train safety system expert at Prorail Train Safety.

Interviewer

Mark Bosschaart

First, a consistency check is done on the basis of the developed value chain in earlier interviews.

Henrik and Wim comment that FIS and the CRS are two different things. Right now CRS looks like a part of FIS; that is not the case. Furthermore, both indicate that the RVTO process planning stage is usually incorporated in the RVTO state of final design (in the case such a planning is required).

Henrik and Wim also provide some detail in the process descriptions. First, they argue that CRS is the only activity that Prorail completely executes themselves. Second, the result of the FIS then is the operational solution to achieve the CRS.

Mark asked which institutions are involved in the process.

Henrik explains that the ideal situation contains one engineering agency for FIS, RVTO and SWOD. Prorail prefers this setting to improve data reliability. In reality, FIS and RVTO are done by one engineering agency; SWOD by another. This makes the total of engineering agencies two instead of one. In large projects it might occur that three engineering agencies are involved.

Mark asks how Prorail controls this design process.

Henrik explains that a validation check is performed at the end of the FIS and at the end of RVTO. A validation check will NOT be performed at the end of SWOD.

Wim says that he has never experienced the situation in which he accepted a FIS or RVTO at once. Most of the time, the second check is correct.

Wim and Henrik continue by explaining that this implies feedback loops throughout the FIS and RVTO processes.

Especially the FIS, which is fed by the CRS, might substantially change over time due to different interest of actors that is reflected in the CRS.

Mark asks about process times, both value adding and non value adding.

Wim, Henrik, Sander and Mark map all times relevant for the value chain. The result is the value chain map in Appendix D.

Wim further estimates that the time he needs to check FIS and RVTO are both about 2 to 3 weeks, while a disapproved FIS/RVTO easily leads to 40 hours of additional work. The times provided include those validation check times.

** as a site note from Mark: the times of the earlier interview with Sander and Jan correspond quite well to the times given by Henrik and Wim. Also, the total process times correspond quite well. This raises the reliability of the figures.

Mark asks the gentlemen about the number of people involved.

Wim and Henrik indicate that CRS and FIS have about 2, RVTO (for all stages) a pool of 5, SWOD a pool of 4 and the protocol development and dry test about 1.

Mark also is interested in the resulting data of each stage.

CRS leads to a set of requirements

FIS results in a report with some sketches of the proposed track layout ideas.

RVTO results in a report, applied traffic OBE maps and OS maps. The RVTO however requires external civil track data and traction data.

SWOD leads in three files: the final OBE, OS and SVA

Mark asks about the feedback loops: what does this imply for process orders?

Wim explains that theoretically each process needs to be finished before the next starts. This is necessary because each subsequent step desires the preceding data. In reality however, processes start earlier than desired. This follows from engineering agencies that do not want to wait too long as that would raise their costs.

Mark asks whether there is a target specification for interlocking projects?

Henrik explains that each project has a specific target. Though there is no strict deadline.

Mark asks both whether there exist a project database concerning interlocking projects?

Henrik says that there is no such thing.

Mark, in reply to the previous question, asks the gentlemen about their experiences and what they believe proves most inefficient?

Wim and Henrik give six answers:

1. The overlap in processes discussed earlier, leads to errors that later need to be restored
2. The translation from VTOBE in RVTO to OBE in SWOD does not work like a charm. Occasionally, the SWOD

responsible engineering agency finds errors in the VTOBE that they or the previous engineering agency needs to restore

3. Control check at Loxia often indicates that the design requirements are rarely completely according to the norms
4. The fundamental RVTO which forms the basis for an OBE has to be made from scratch for each project of the same interlocking area
5. It occurs that engineering agencies accept deliverables checked by Prorail (which receive a so-called protocol) while Prorail explicitly meant different files
6. Many files and maps are still exchanged and interpreted by non machine readable files. This increases the chance on mistakes significantly

Others:

Please try to make some conclusions on rail operation safety as well.

Titel

Interview Siemens IT department

Date

June 18th 2013

Location

Siemens, Prinses Beatrixlaan 800, Den Haag

Duration

14:00 – 15:30

Interviewee

Elsa Gonzalez Palacios – IT tool developer at Siemens' Engineering and Industrial Automation department

Interviewer

Mark Bosschaart

Mark explains the research subject and the goal of IT system RailML.

Elsa starts by explaining the basic framework for introducing a new framework like RailML. She says that, simply put, the success of an automation tool used across the chain with multiple parties depends on three aspects:

1. Data arrival – i.e. do we need to check the data, how do we check the data and who is responsible
2. Data transformation – i.e. how do we use the data and which arrangements/protocols do we make across the chain
3. Export data – i.e. in which way or format will the data be stored
 - In order to achieve an overview of the above, extensive interviews need to be kept.

Elsa goes on by explaining that although the above might seem simple, a lot goes wrong with the management of various data versions. Elsa explains that it is inevitable that conflicting or outdated versions will circulate. The question therefore is not how to prevent it, but how to limit the effects. The solution goes beyond just agreements on processes or protocols. The process and data needs to be transparent. Thus, if you assume that the data is invalid, just do it wrong if you cannot solve it. The next developer will directly recognize the mistake and maybe is able to solve it. This prevents an endless feedback cycle between two processes.

Mark explains which advantages RailML might potentially deliver.

Elsa directly answers that it will never be as simple as pushing a button and downloading your interlocking files. There will always be new protocol rules or project adjustments that require (specific) changes to the outcome. As a result, you would want to know whether a party may change the data or not. Furthermore, upfront data validation of each process becomes vital.

In general, Elsa believes that the RailML approach has two major disadvantages:

1. You barely see what you are working on
2. Agreements / formats / protocols will not completely be applied due to traditional approaches.

To mitigate the second issue, a very tight cooperation between all parties is required. The question remains how can you keep all parties aligned. The challenges increase because IT systems nowadays have a very short life cycle. RailML will thus contain new aspects, formats, functionality etc every few years or even months.

Mark specifically discusses the structure of a RailML file and the relations with a subsidiary as interlocking. When something changes, the relations need to change too. How would that best work?

Elsa says that ideally all related elements change automatically. The problem though is that you need to be sure that ALL related elements changed. Otherwise you will get errors and it will definitely take a lot of time to restore them. Furthermore, the way the elements change automatically depends on the specific project. This makes things interesting because if you would start a new interlocking project, the approach is mostly top-down (start with infra then interlocking relations). Although if you would undergo an interlocking modification project your approach would typically be bottom-up (and you might start with a specific interlocking relation that needs to be changed in infrastructure). A bottom-up approach is harder because you are not aware if the static data may / can change. How does your automatic approach reckon that this? Elsa then says that the only way is a lot of verification and validation. Verification will work but validation is hard. So you would best define and prove a clear process structure / protocol and check whether that has been done correctly. You thus check on process instead of result.

Mark touches the topic of standardization.

Elsa questions whether you need to include various lingual translations of the format. Engineering terms are very specific and this might be the key issue of concern: if the people that work with the system do not understand it, you will definitely get problems. Furthermore, you need to make a trade-off: will you make a very clear standardization or a more flexible standardization. The first has the disadvantage that party specific habits return and result in interpretation errors. A flexible approach might however not achieve the standardizing approach across multiple countries and railway infrastructure managers.

An important start when deciding about this trade-off is to find out which data a party's process need from outside. That is the critical data you need to standardize. A clear distinction between internal and external data should be made. This should lead to a consensus in a situation that every party wants to standardize, though with their own standards. Its about finding a common ground and shared benefits.

To conclude she says, two things should be managed very well:

1. New format (RailML) versions
 2. Data adjustments (e.g. new signals, extra switches etc)
- You will change the system: account for that situation!!

Last: the system should never become a black box. Then you will lose the overview and make mistakes. Therefore, visualization and proper instructions are required.

Titel**Interview at Loxia****Date**

June 19th 2013

Location

Loxia, Godebaldkwartier 385, Utrecht

Duration

14:30 – 16:00

Interviewee

Gerben Schut – Functional manager of InfraAtlas at Loxia

Sander Dragt – Train safety system expert at Prorail Train Safety

InterviewerMark Bosschaart

Mark first would like to what the main activities of Loxia are and what the common ground is with rail safety systems.

Gerben explains that they focus on InfraAtlas which he describes as a complete description of the rail topology in the Netherlands. InfraAtlas is a planning system meant to support the development or modification projects related to rail infrastructure. The only common ground Loxia has with rail safety systems are the elements required for OBE maps.

Mark explains the engineering design process and asks what the role of Loxia is within that chain.

Gerben tells that Loxia develops and maintains all aspects of the InfraAtlas. Loxia itself does not interact with the interlocking process. However, engineering agencies involved in the interlocking process retrieve data from the InfraAtlas to develop OBE maps for example.

How does Loxia keep IT architecture definitions unique?

This is something that Loxia is aware of but does not manage. Especially since there are other related databases and systems too.

How does Loxia tackle version management?

Loxia tackles version management by means of cut outs in the network. An engineering agency then gets the task to verify and propose the adjustment to such a section of data. Furthermore, for instance with an interlocking project, new OBE maps get communicated directly to Loxia. Loxia then aims to adjust InfraAtlas before the section gets operational again. With small interlocking projects, this can be quite a challenge. Gerben stresses that it is very important that all engineering agencies and Loxia work with the same format / protocols to prevent errors.

What does Loxia want to do with RailML? What is Gerben's vision on RailML?

Loxia currently does not have plans regarding RailML. He believes that RailML development will not go very quickly. The main issue is that a lot of infrastructure managers do not have such extended databases with rail topology as here in the Netherlands. We are, together with some other countries like Switzerland, frontrunners. The underlying reason arises from the high Dutch rail performance demands which require extensive calculation models. Furthermore, RailML is a nice tool to describe rail infrastructure based on nodes and branches, but a lot of elements can still not be represented. Examples are various types of signals. In addition, he believes that RailML will not become a database that for instance replaces InfraAtlas. An important gap in RailML is the time factor related to elements. Gerben sees RailML as a data exchange tool; compare it to a travel plug. On the other side of the spectrum is ETCS however that requires IT systems for development due to its complexity. Therefore, the implementation of ETCS could be a catalyst of RailML.

Gerben mentions the tool InfraAtlas to RailML v1

Gerben explains that the tool was originally developed for simulation purposes at DHV. The tool aims to simplify and quicken the process of converting track data into Opentrack. The tool has never been completely finished and is currently not used (that often) anymore.

OBE maps and related results present rail infrastructure in a straight way. For the purpose of modeling and simulation, curvature is required. How could you include that in your data?

Gerben does not have an idea how this could be included. He stresses that the advantage of a schematic overview like an OBE is the fact that there are no (actual) curves represented. Therefore, InfraAtlas data is actually not meant to represent curves and so on.

Gerben continues by mentioning that some parties including Prorail and Loxia are measuring the RD-coordinates of the Dutch rail network with special trains. These trains measure by (3d) cameras, gps, odometers and so on.

How do you include RD-coordinates with InfraAtlas? Could this be a solution for the curvature issue (of the previous question)?

The measuring trains' route stores the sequence of rail infrastructure elements it passes (e.g. switches and signals) and can therefore easily couple the RD data to the current InfraAtlas data with the exception of some harder cases like bufferstops. Gerben explains that those coordinates will not solve the curvature issue because by using the coordinates, your network will not be schematic anymore. A coordinate approach will deliver a more GIS style network which does not have the transparency and conciseness of a schematic approach. Gerben believes that manual work remains required to implement curvature in a schematic network for modeling purposes.

How do you visualize the data process and prevent a black box of data?

Generally, Railmaps is designed for that purpose. Railmaps represents and visualizes the Dutch rail network with a wide variety of characteristics. Railmaps not just include track topology but also aspects like pve, longitudinal data, timetable nodes, axle loads and so on. Engineering agencies currently use Railmaps to require the necessary data for interlocking projects. Railmaps currently is quite slow which limits its use. InfraMonitor is another tool to visualize data from InfraAtlas but one that only Loxia uses. It is a very powerful tool to draw track sections and is much faster than the complete drawing of an OBE. Furthermore, InfraMonitor can be used for the purpose of capacity simulation.

Titel

Interview at the Prorail Simulation Department

Date

July 4th 2013

Location

Prorail, De Inktpot, Utrecht

Duration

13:00 – 14:00

Interviewee

Dick Middelkoop – Program manager model development at Prorail

Sander Dragt – Train safety system expert at Prorail Train Safety

Interviewers

Mark Bosschaart

Bob Janssen

Mark explains the reason for the meeting by elaborating on the research project. It is ought that RailML might provide added benefit for the interlocking design process when combined with simulation.

Dick splits his reply into two parts: the tool (RailML) decision and the method of simulation. Dick explains that they develop a tool called INCA with the same purpose as RailML, but for use during the interlocking design processes at Prorail only. The biggest advantage of INCA over RailML is that it is directly aligned to InfraAtlas and does not need a conversion process. Furthermore, INCA is closer to a new database than a data exchange tool since it can record the mutation process of data through the chain. INCA aims to provide each process with the necessary data, to store the intermediate enriched data and to pass the data to the next process until the end of the chain results in an OBE, OR and SVA. Parties' tools can be aligned to INCA for a smooth process. As long as the result can be imported in INCA according to the desired format, the party may do whatever he desires, e.g. simulation, to derive the results

The second topic that Dick mentions with regard to the statement is whether simulation is really needed. Opentrack is a nice tool but you don't always need it. There are generally four reasons to perform a simulation test:

1. Multi train run for the purpose of stability tests
2. Visualizations
3. Education
4. Gaming (people interaction)

How does Prorail currently make simulation models and for what purposes?

Simulation with Opentrack is mostly used for capacity and timetable stability projects. A tool has been developed to

directly import data from InfraAtlas into Opentrack. This is actually as simple as pushing a button.

Mark asks why they chose to develop INCA instead of RailML and if RailML has been used in the past?

There has been a pilot with RailML several years ago. A conversion tool from InfraAtlas to RailML has been made for that purpose. The big issue was that, especially at that time, a lot of options in RailML were missing. Also, as just explained, IA to RailML requires an additional conversion process. Last, RailML has no further advantage for the use at the interlocking design processes that Prorail controls from CRS to SWOD. Dick stresses that he agrees on the idea of RailML. Dick believes that tools like RailML and/or INCA decrease engineering times and improve the final product quality because the workload decreases and IT systems prevent a lot of data errors. He also understands that a standardized data exchange protocol will be very effective for the connection between infrastructure managers and industry. Dick however experienced that within the interlocking design part of the chain, RailML would result in more work due to the extra conversion process. Standardization only works when a lot of parties cooperate. That makes sense from the industry point of view (Siemens) as they develop interlocking systems for multiple railway companies. It does however not lead to sufficient gains when only used at Prorail and a few Dutch engineering agencies. Furthermore, Dick does not believe that RailML can eventually align every database, program and output in the chain. In other words: one tool is not the solution but we should focus on limiting the total amount of tools.

What are the challenges in making the chain leaner with RailML or INCA?

The main challenges cope with a so-called data gap and the location of source data. The data gap challenge arises in the cooperation process with engineering agencies. When you provide them with machine readable data, instead of them measuring it, they will lose work. Therefore, they would try to earn more money with the same data. One way to do this is to use the data in other, non Prorail projects, as well. This leads to two problems: one, the revenues do not flow back into development of the system (free rider problem), two, the data does not get updated at both locations. This leads to the second challenge because the more tools you use and the more data is applied in different processes, the more difficult it is to find the source location with the most up to date data.

How could you mitigate those problems?

This is something we are working on with the engineering agencies at this point. You should think of an open data source that all parties can access, provide updates to and finance development and maintenance. It is different from open source because we do not want parties to change to tool in itself.

Dick explains that Prorail tries to develop a system which covers all the vital elements of railway operation like traction, infrastructure, timetable, rolling stock and so on. On top of that, they want to develop a simulation layer/tool that includes the relevant items of those elements for

simulation. In this way, they want to align the various disciplines in Prorail. Currently, the simulation results remain in the specific domain while they might enrich other studies too. Also, they want to align parties in the chain by including them in this system. The idea is to enable easy data exchange without the need for conversion.

Mark asks whether simulation like Opentrack could be useful in the context of RailML and the interlocking chain?

This depends on your analysis goal. Usually we define three layers within this respect. Do we want results on a micro, meso or macro level of detail; but actually there is one more: the nano level. The nano level reflects a highly detailed operation level of technology like the interlocking or safety system. Opentrack is a nice tool to do stability analyses for capacity estimations and timetable development on a micro, meso or macro level. Opentrack does not make sense on a nano level. When there are no stochastic variables and interactions of multiple replications do not seem to play an important role, simulation and Opentrack do not need to be used. In that case, on a micro level you could use FRISO; at meso you could use SIMONE; at macro level you could use SITA; at nano level BITS. BITS is a kind of emulator for the signaling system in place.

Experts believe that RailML might shorten the testing process, for example by simulation. Are they incorrect then?

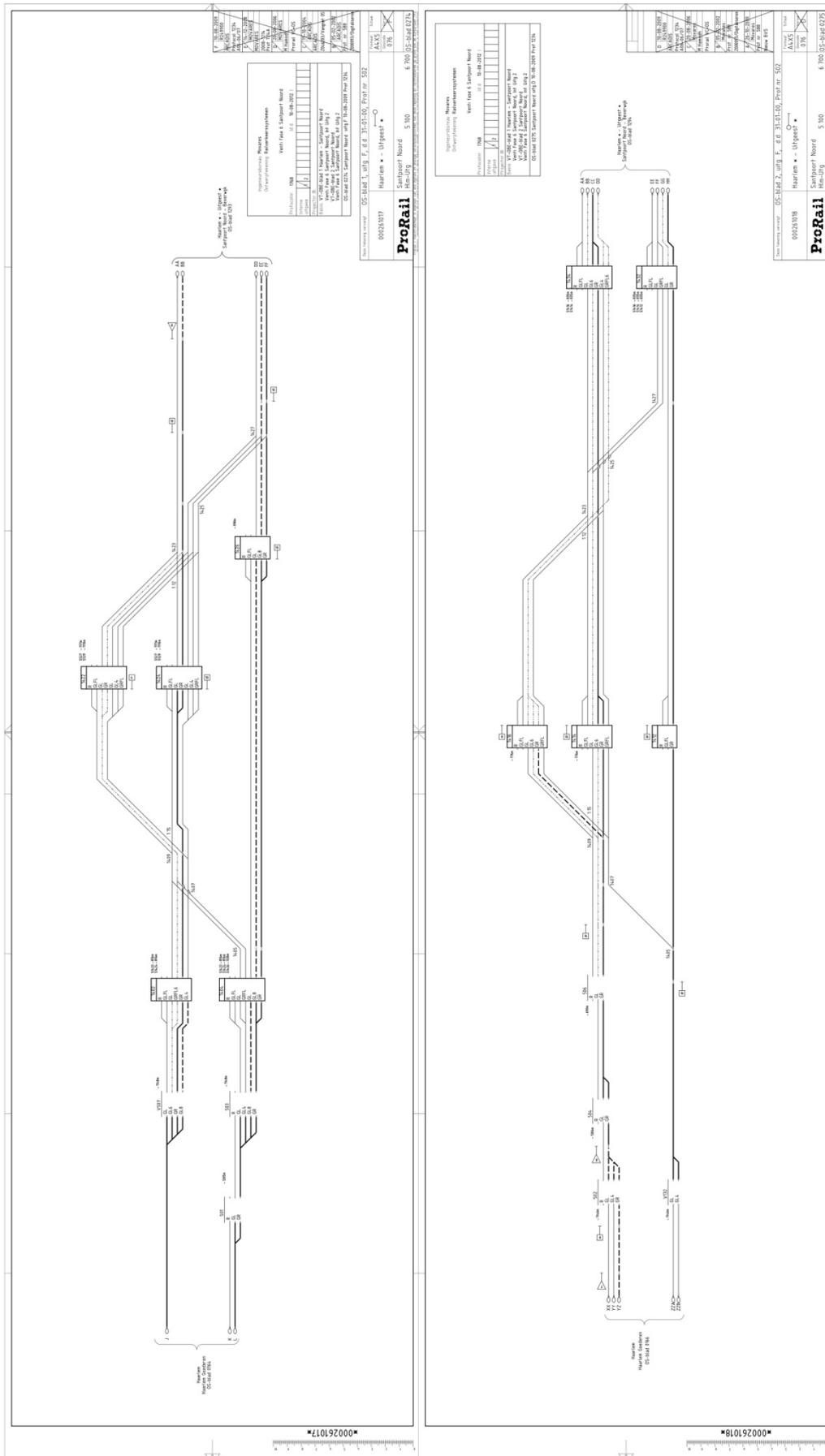
RailML or a comparable tool might definitely shorten the testing process because of potential for simpler validation for example. But simulation is not required in this context.

Experts also state that an Opentrack model will be enriched with the interlocking formalization in RailML. To which extent will the results improve?

I do not agree that they will. More detail is not always better. At some point, you lose overview when having a very detailed system. In addition, you also need to maintain the data to keep it up to date. That becomes a lot harder with more elements.

Some simulation practitioners however believe that RailML with interlocking might enable simulation like Opentrack to test the interlocking aspect dependencies or to compare track capacity between different signaling systems?

Dick explains that he doubts the added benefit. He experiences that capacity estimations are already very accurate and he thinks that the interlocking formalization in RailML will not significantly improve capacity tests. Furthermore, to measure the difference in capacity when implementing ETCS to replace ATB, simulation is not required. In addition, he again states that simulation is also not required to test interlocking systems. BITS would be a good tool for Prorail to check industry's interlocking product by emulation. Industry, like Siemens, probably have tools themselves to test the specific interlocking system EBS as well.



First, the analysis investigates the rail infrastructure from the Santpoort area in the direction of Haarlem. A total of 9 inconsistencies between VT-OBE/OS and reality are counted. In reality:

1. no static speed sign in the direction of Hlm at track 1 at signal 514;
2. no static speed sign in the direction of Hlm at track 2 at signal 516;
3. no static speed sign in the direction of Hlm at track 3 at signal 518;
4. a wrong signal number at track 1: 514 should be 1412;
5. a wrong signal number at track 2: 516 should be 1414;
6. a wrong signal number at track 3: 518 should be 1416;
7. a dwarf signal (514) which should be a normal signal;
8. no progressive speed indicator at signal 514;
9. a wrong speed indication sign at track 2, after the switch.

Especially the dwarf signal is a matter of concern for an interlocking system since the possible signal aspects change.

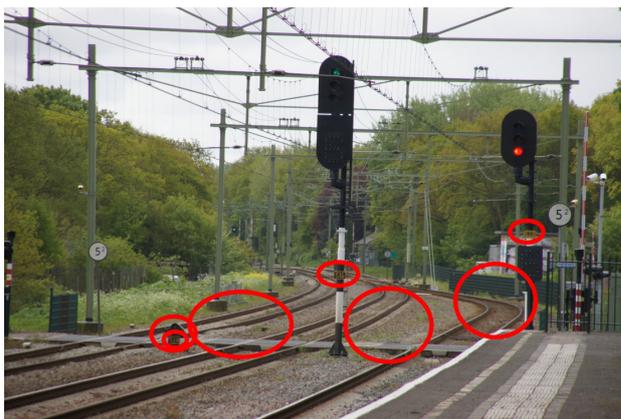


Figure 151: The findings of figure 1 from yet another viewpoint. Again the inconsistencies are confirmed.



Figure 149: The red circles mark wrong signal numbers, missing static speed signs and a dwarf signal which should actually be a normal signal.



Figure 152: Figure 1 from a further perspective and at different angle. It can be clearly noticed that there is not a single static speed sign.

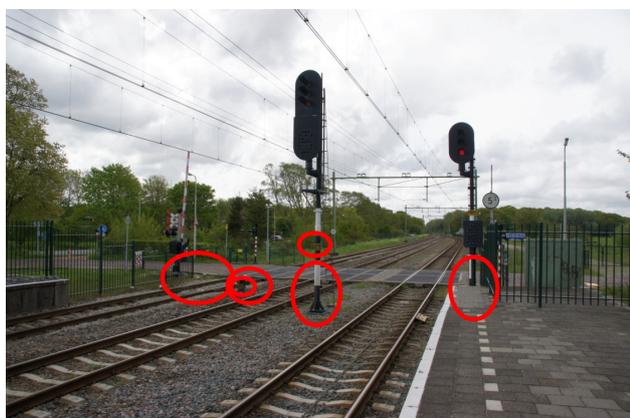


Figure 150: The findings of figure 1 confirmed from another angle.

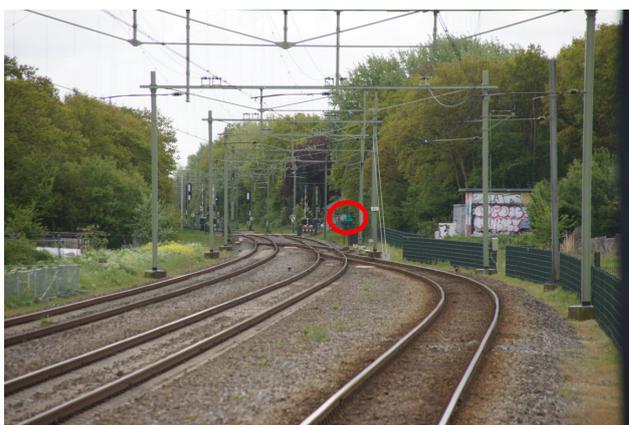


Figure 153: Downstream towards Hlm the speed sign is incorrectly drawn on the OBE. The OBE visualizes a standard 10 speed sign, while it should be a triangle accelerate to 10 speed sign.

Second, the Bv direction is analyzed. 15 inconsistencies are found in this direction. In reality:

1. signal number 519 should be 1422;
2. signal number 521 should be 1424;
3. signal number 523 should be 1426;
4. dwarf signal 519 should be a normal signal;
5. dwarf signal 521 should be normal signal;

6. a progressive speed sign should be positioned next to signal 519;
7. a progressive speed sign should be positioned next to signal 521;
8. a static speed sign 6 misses next to signal 519;
9. a static speed sign 10 misses next to signal 521;
10. a static speed sign 10 misses next to signal 523;
11. switch number 19 should be 1423;
12. switch number 20a should be 1425;
13. switch number 20b should be 1427;
14. a static speed sign 10 misses after switch 20a;
15. static speed profile 8 after signal 20a should be triangle shaped with its point downwards.



Figure 154: The red circles indicate two dwarf signals that should be two normal signals with progressive speed signs on paper. Furthermore, two static speed signs are missing and the signal numbers are incorrect.



Figure 155: Confirms the findings of the previous figure and shows that the static speed sign also misses at track 1.



Figure 156: Again confirms the findings of the previous two figures and also misses a static 10 speed profile after the switch that merges tracks 2 and 3.



Figure 157: This figure also finds that the switch numbers are incorrect and the signal number at track 1 is incorrect.



Figure 158: The standard static speed sign 8 in the back should be a triangle facing downwards with an 8. In addition, all three switch numbers are incorrect. Again the static speed sign 10 towards Bv on the left track misses as well.

Then, three more inconsistencies are found in the Sptn area. In reality:

1. switch number 7a should be 1405;
2. switch number 7b should be 1407;
3. switch number 8 should be 1409.



Figure 159: Proves that three switches are characterized by a wrong number.

The above inconsistencies count for the VT-OBE and OS that result from the RVTO process. The SWOD adjusts the VT-OBE to result in an OBE. The latest Sptn OBE does not contain inconsistencies. A recent change of infrastructure numbers, communicated by a separate report in the RVTO, however shows that both the OBE and real life infrastructure contain 12 inconsistencies with that report. These inconsistencies include 6 conflicting signal numbers (Hlm direction bullits 4, 5 and 6; Bv direction bullits 1, 2 and 3) and 6 conflicting switch numbers (Bv direction bullits 11, 12 and 13; next to platforms bullits 1, 2 and 3).

Appendix H - Element Complexity Analysis

This appendix relates various rail elements with each other for every station interlocking area in the Netherlands. The analysis aims to support conclusions on the level of engineering complexity expressed in the size of the interlocking area. InfraAtlas provides the data for this analysis. The analysis investigates six relations of rail elements:

- the amount of track circuit sections versus the amount of track welds;
- the amount of track circuit sections versus the amount of switches;
- the amount of track circuit sections versus the amount of main signals;
- the amount of track circuit sections versus the amount of tracks;
- the amount of main signals versus the amount of switches;
- the amount of main signals versus the amount of tracks.

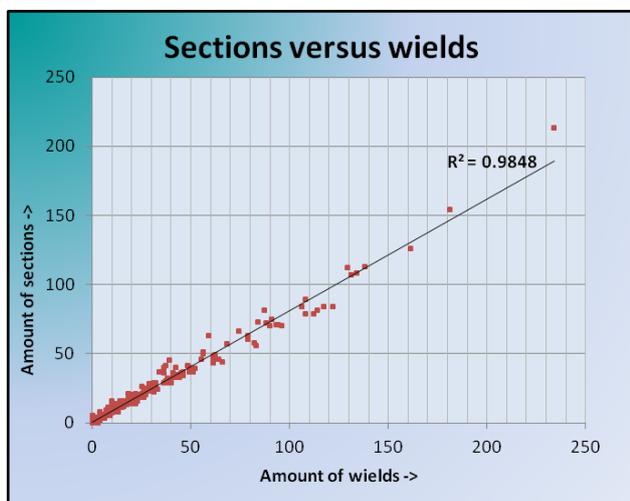


Figure 160: The amount of track circuit sections per amount of welds for every station interlocking area in the Netherlands. The relation is surprisingly linear with no clear deviations. The amount of welds appears to be somewhat larger over the amount of sections.

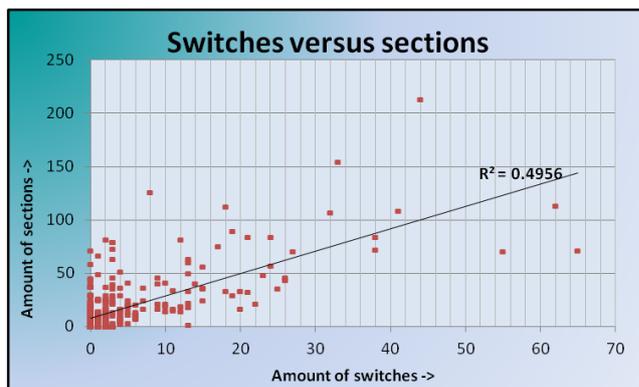


Figure 161: The amount of track circuit sections per amount of switches for every station interlocking area in the Netherlands. The linear relation between the interlocking areas is not very strong because of the variety.

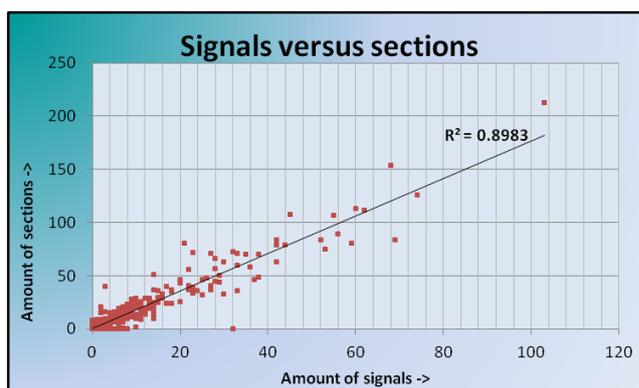


Figure 162: The amount of track circuit sections versus the amount of signals per station interlocking area in the Netherlands. The data points indicate a linear relation between both rail elements although there exists some deviation.

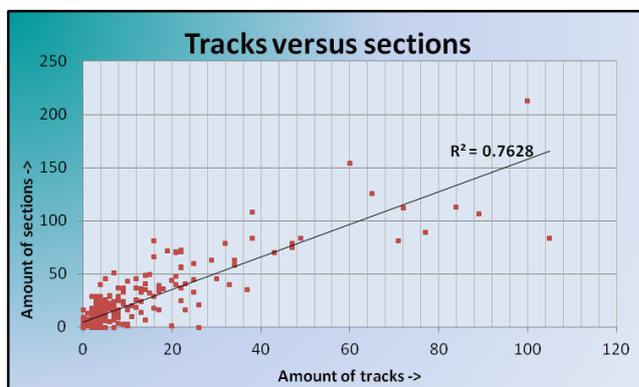


Figure 163: The amount of track circuit sections versus the amount of tracks per station interlocking area in the Netherlands. Tracks being defined as the amount of tracks from node to node; where a node can be a switch or bufferstop. The data points somewhat indicate a linear relation between both rail elements although there exists some deviation.

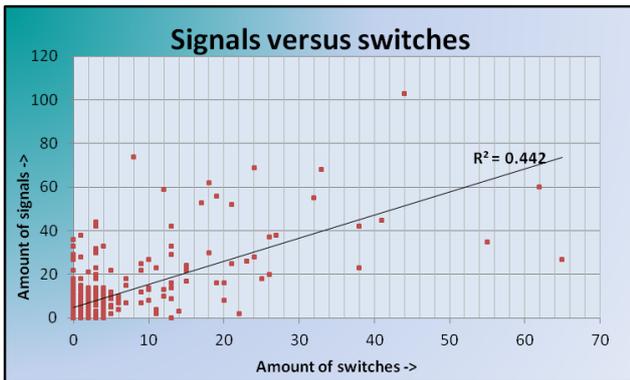


Figure 164: The amount of signals versus the amount of switches per station interlocking area in the Netherlands. The linear relation between the interlocking areas is not very strong because of the large variety.

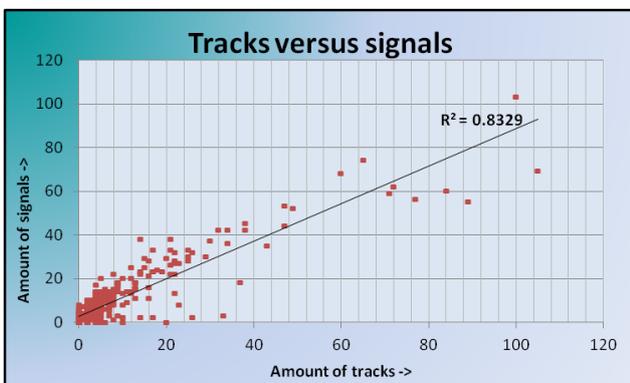


Figure 165: The amount of tracks versus the amount of signals per interlocking area in the Netherlands. Tracks being defined as the amount of tracks from node to node; where a node can be a switch or bufferstop. The data points appear to show a linear relation although with quite some downward deviation.

Appendix I - Sptn in RailML

The RailML is based on the VT-OBE in Appendix G.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Created with Liquid XML 2013 Designer Edition (Trial) 11.0.10.4574
(http://www.liquid-technologies.com) -->
<!-- $Id$ -->
<railml xmlns="http://www.railml.org/schemas/2013"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.railml.org/schemas/2013 ../schema/railML.xsd"
  version="2.2">
  <infrastructure id="IS">
    <infraAttrGroups>
      <infraAttributes id="Nederland">
        <owner ownernName="Prorail" />
        <trainProtection medium="rail"
          monitoring="continuous" />
        <epsgCode default="coord"
          extraHeight="nap" />
      </infraAttributes>
    </infraAttrGroups>
    <tracks>
      <track id="Sptn_1425R_1427R"
        description="Sptn overloopwissel Noord">
        <trackTopology>
          <trackBegin id="Sptn_1425R"
            pos="0"
            absPos="6.065">
            <connection id="Sptn_1425RV"
              ref="Sptn_1425VR" />
          </trackBegin>
          <trackEnd id="Sptn_1427R"
            pos="0.048"
            absPos="6.113">
            <connection id="Sptn_1427RV"
              ref="Sptn_1427VR" />
          </trackEnd>
        </trackTopology>
        <trackElements>
          <gradientChanges>
            <gradientChange id="Sptn_1425R_10000"
              pos="0"
              absPos="5.41"
              dir="up"
              slope="0.061" />
          </gradientChanges>
          </trackElements>
          <ocsElements>
            <trainDetectionElements>
              <trackCircuitBorder id="Sptn_1425R_100"
                pos="0.015"
                name="1423T_1427T"
                absPos="5.080"/>
            </trainDetectionElements>
          </ocsElements>
        </track>
      <track id="Sptn_1405L_1407L"
        description="Sptn overloopwissel Zuid">
        <trackTopology>
          <trackBegin id="Sptn_1405L"
            pos="0"
            absPos="5.41">
            <connection id="Sptn_1405LV"
              ref="Sptn_1405VL" />
          </trackBegin>
          <trackEnd id="Sptn_1407L"
            pos="0.08"
            absPos="5.49">
            <connection id="Sptn_1407LV"
              ref="Sptn_1407VL" />
          </trackEnd>
        </trackTopology>
        <trackElements>
          <gradientChanges>
            <gradientChange id="Sptn_1405L_10000"
              pos="0"
              absPos="6.065"
              dir="up"
              slope="0.29" />
          </gradientChanges>
          </trackElements>
          <ocsElements>
            <trainDetectionElements>
              <trackCircuitBorder id="Sptn_1405L_100"
                pos="0.046"
                name="1405T_1407T"
                absPos="5.456" />
            </trainDetectionElements>
          </ocsElements>
        </track>
      <track id="Hlm_137aL_Sptn_1407R"
        description="Sptn richting Hlm">
        <trackTopology>
          <trackBegin id="Hlm_137aL"
            pos="0"
            absPos="X">
            <connection id="Hlm_137aLV"
              ref="Hlm_137aVL" />
          </trackBegin>
          <trackEnd id="Sptn_1407R"
            pos="X"
            absPos="5.49">
            <connection id="Sptn_1407RV"
              ref="Sptn_1407VR" />
          </trackEnd>
        </trackTopology>
        <trackElements>
          <gradientChanges>
            <gradientChange id="Sptn_1407R_10000"
              pos="0"
              name="1407R10000"
              absPos="X"
              dir="up"
              slope="X"
              description="gradient changes outside Santpoort area" />
            <gradientChange id="Sptn_1407R_1200"
              pos="X"
              name="1407R1200"
              absPos="5.135"
              dir="up"
              slope="-0.190" />
          </gradientChanges>
          <speedchanges>
            <speedchange id="Sptn_1407R_200"
              pos="x"
              absPos="5.416"
              vmax="100"
              dir="down" />
          </speedchanges>
          <levelCrossings>
            <levelCrossing id="Hlm_137aL_Sptn_1407R_5.290"
              pos="x"
              absPos="5.290" />
          </levelCrossings>
        </trackElements>
        <ocsElements>
          <signals>
            <signal id="Sptn_1402"
              pos="X"
              absPos="5.135"
              name="1402"
              dir="up" />
            <signal id="Sptn_1407R_1600"
              pos="x"
              absPos="5.100"
              dir="down"
              virtual="true"
              description="Open track to station area transition"/>
            <signal id="Sptn_1407R_1601"
              pos="x"
              absPos="5.100"
              dir="up"
              virtual="true"
              description="Open track to station area transition"/>
            <signal id="X"
              pos="X"
              description="more signals in this track but outside Sptn area" />
          </signals>
          <trainDetectionElements>
            <trackCircuitBorder id="Sptn_1407R_300"
              pos="X"
              name="1407T_1414CT"
              absPos="5.416"/>
            <trackCircuitBorder id="Sptn_1407R_500"
              pos="X"
              name="1414CT_1414DT"
              absPos="5.315"/>
            <trackCircuitBorder id="Sptn_1407R_700"
              pos="X"
              name="1414DT_1414ET"
              absPos="5.261"/>
            <trackCircuitBorder id="Sptn_1407R_1000"
              pos="X"
              name="1407T_1414CT"
              absPos="5.416"/>
          </trainDetectionElements>
        </ocsElements>
      </track>
    </tracks>
  </infrastructure>
</railml>

```

```

        pos="X"
        name="1414ET_141FT"
        absPos="5.149"/>
    <trackCircuitBorder id="X"
        description="more traindetectors in this track but outside
Sptn area" />
    </trainDetectionElements>
</locsElements>
</track>
<track id="Hlm_137bV_Sptn_1405V"
    description="Hlm naar Sptn">
<trackTopology>
    <trackBegin id="Hlm_137bV"
        pos="0"
        absPos="X">
        <connection id="Hlm_137bVL"
            ref="Hlm_137bVL" />
    </trackBegin>
    <trackEnd id="Sptn_1405V"
        pos="X"
        absPos="5.41">
        <connection id="Sptn_1405VR"
            ref="Sptn_1405RV" />
    </trackEnd>
</connections>
    <switch id="Sptn_1405"
        pos="X">
        <connection id="Sptn_1405VL"
            ref="Sptn_1405LV"
            orientation="outgoing"
            course="left" />
    </switch>
    <switch id="X"
        pos="X"
        description="switch in Hlm area">
    </switch>
</connections>
</trackTopology>
<trackElements>
<gradientChanges>
    <gradientChange id="Sptn_1405V_10000"
        pos="0"
        absPos="X"
        dir="up"
        slope="X"
        description="gradient changes outside Santpoort area" />
    <gradientChange id="Sptn_1405V_800"
        pos="X"
        absPos="5.135"
        dir="up"
        slope="-0.221" />
</gradientChanges>
<speedchanges>
    <speedchange id="Sptn_1405V_400"
        pos="x"
        absPos="5.261"
        vmax="100"
        dir="down" />
</speedchanges>
<levelCrossings>
    <levelCrossing id="Hlm_137bV_Sptn_1405V_5.290"
        pos="x"
        absPos="5.290" />
</levelCrossings>
</trackElements>
<ocsElements>
<signals>
    <signal id="Sptn_1404"
        pos="X"
        absPos="5.135"
        name="1404"
        dir="up" />
    <signal id="Sptn_1405V_1200"
        pos="x"
        absPos="5.100"
        dir="down"
        virtual="true"
        description="Open track to station area transition"/>
    <signal id="Sptn_1405V_1201"
        pos="x"
        absPos="5.100"
        dir="up"
        virtual="true"
        description="Open track to station area transition"/>
    <signal id="X"
        description="more signals in this track but outside Sptn" />
</signals>
</trainDetectionElements>
<trackCircuitBorder id="Sptn_1405V_100"
    pos="X"
    name="1404BT_1405T"
    absPos="5.315" />
<trackCircuitBorder id="Sptn_1405V_300"

```

```

        pos="X"
        name="1404AT_1404BT"
        absPos="5.261" />
    <trackCircuitBorder id="Sptn_1405V_600"
        pos="X"
        name="503FT_1404AT"
        absPos="5.170" />
    <trackCircuitBorder id="X"
        description="more traindetectors in this track but outside
Sptn area" />
    </trainDetectionElements>
</locsElements>
</track>
<track id="Sptn_1405R_1427L"
    description="Sptn station spoor 1 met perron">
<trackTopology>
    <trackBegin id="Sptn_1405R"
        pos="0"
        absPos="5.41">
        <connection id="Sptn_1405RV"
            ref="Sptn_1405VR" />
    </trackBegin>
    <trackEnd id="Sptn_1427L"
        pos="0.703"
        absPos="6.113">
        <connection id="Sptn_1427LV"
            ref="Sptn_1427VL" />
    </trackEnd>
</trackTopology>
<trackElements>
<gradientChanges>
    <gradientChange id="Sptn_1405RX"
        pos="0"
        absPos="5.41"
        dir="up"
        slope="-0.221" />
    <gradientChange id="Sptn_1405R_800"
        pos="0.358"
        absPos="5.768"
        dir="up"
        slope="0.678" />
    <gradientChange id="Sptn_1405R_1300"
        pos="0.653"
        absPos="6.063"
        dir="up"
        slope="0.056" />
</gradientChanges>
<speedchanges>
    <speedchange id="Sptn_1405R_X1"
        pos="0.358"
        absPos="5.768"
        vmax="100"
        dir="down" />
    <speedchange id="Sptn_1405R_X2"
        pos="0.653"
        absPos="6.063"
        vmax="100"
        dir="up" />
</speedchanges>
<levelCrossings>
    <levelCrossing id="Sptn_1405R_1427L_5.76"
        pos="0.35"
        absPos="5.760" />
</levelCrossings>
</trackElements>
<ocsElements>
<signals>
    <signal id="Sptn_1412"
        pos="0.358"
        absPos="5.768"
        name="1412"
        dir="down" />
    <speed>
    <speedChangeRef ref="Sptn_1405R_X1"/>
    </speed>
</signal>
    <signal id="Sptn_1426"
        pos="0.653"
        absPos="6.063"
        name="1426"
        dir="up" />
    <speed>
    <speedChangeRef ref="Sptn_1405R_X2"/>
    </speed>
</signal>
</signals>
</trainDetectionElements>
<trackCircuitBorder id="Sptn_1405R_200"
    pos="0.094"
    name="1404CT_1405T"
    absPos="5.504" />
<trackCircuitBorder id="Sptn_1405R_400"
    pos="0.326"

```

```

        name="1404CT_1404DT"
        absPos="5.736" />
<trackCircuitBorder id="Sptn_1405R_600"
  pos="0.356"
  name="1404DT_1404ET"
  absPos="5.766" />
<trackCircuitBorder id="Sptn_1405R_1400"
  pos="0.662"
  name="1404ET_1427T"
  absPos="6.072" />
</trainDetectionElements>
</locsElements>
</track>
<track id="Sptn_1409R_1423L"
  description="Sptn station spoor 2">
<trackTopology>
<trackBegin id="Sptn_1409R"
  pos="0"
  absPos="5.52">
<connection id="Sptn_1409RV"
  ref="Sptn_1409VR" />
</trackBegin>
<trackEnd id="Sptn_1423L"
  pos="0.52"
  absPos="6.04">
<connection id="Sptn_1423LV"
  ref="Sptn_1423VL" />
</trackEnd>
</trackTopology>
<trackElements>
<gradientChanges>
<gradientChange id="Sptn_1409R_10000"
  pos="0"
  absPos="5.52"
  dir="up"
  slope="-0.190" />
<gradientChange id="Sptn_1409R_700"
  pos="0.248"
  absPos="5.768"
  dir="up"
  slope="0.765" />
<gradientChange id="Sptn_1409R_1200"
  pos="0.431"
  absPos="5.951"
  dir="up"
  slope="0.077" />
</gradientChanges>
<speedchanges>
<speedchange id="Sptn_1409R_X1"
  pos="0.248"
  absPos="5.768"
  vmax="100"
  dir="down" />
<speedchange id="Sptn_1409R_X2"
  pos="0.431"
  absPos="5.951"
  vmax="100"
  dir="up" />
</speedchanges>
<levelCrossings>
<levelCrossing id="Sptn_1409R_1423L_5.76"
  pos="0.24"
  absPos="5.760" />
</levelCrossings>
</trackElements>
<ocsElements>
<signals>
<signal id="Sptn_1414"
  pos="0.248"
  absPos="5.768"
  name="1414"
  dir="down">
<speed>
<speedChangeRef ref="Sptn_1409R_X1" />
</speed>
</signal>
<signal id="Sptn_1424"
  pos="0.431"
  absPos="5.951"
  name="1424"
  dir="up">
<speed>
<speedChangeRef ref="Sptn_1409R_X2" />
</speed>
</signal>
</signals>
<trainDetectionElements>
<trackCircuitBorder id="Sptn_1409R_100"
  pos="0.048"
  name="1407T_1414BT"
  absPos="5.568" />
<trackCircuitBorder id="Sptn_1409R_300"
  pos="0.216"
        name="1414AT_1414BT"
        absPos="5.736" />
<trackCircuitBorder id="Sptn_1409R_500"
  pos="0.245"
  name="1414AT_1434CT"
  absPos="5.765" />
<trackCircuitBorder id="Sptn_1409R_1000"
  pos="0.453"
  name="1423T_1434CT"
  absPos="5.973" />
</trainDetectionElements>
</locsElements>
</track>
<track id="Sptn_1409L_1423R"
  description="Stpn station spoor 3 met perron">
<trackTopology>
<trackBegin id="Sptn_1409L"
  pos="0"
  absPos="5.52">
<connection id="Sptn_1409LV"
  ref="Sptn_1409VL" />
</trackBegin>
<trackEnd id="Sptn_1423R"
  pos="0.52"
  absPos="6.04">
<connection id="Sptn_1423RV"
  ref="Sptn_1423VR" />
</trackEnd>
</trackTopology>
<trackElements>
<gradientChanges>
<gradientChange id="Sptn_1409LR_10000"
  pos="0"
  absPos="5.52"
  dir="up"
  slope="-0.190" />
<gradientChange id="Sptn_1409L_700"
  pos="0.248"
  absPos="5.768"
  dir="up"
  slope="0.765" />
<gradientChange id="Sptn_1409L_1200"
  pos="0.431"
  absPos="5.951"
  dir="up"
  slope="0.077" />
</gradientChanges>
<speedchanges>
<speedchange id="Sptn_1409L_X1"
  pos="0.248"
  absPos="5.768"
  vmax="80"
  dir="down" />
<speedchange id="Sptn_1409L_X2"
  pos="0.431"
  absPos="5.951"
  vmax="60"
  dir="up" />
</speedchanges>
<levelCrossings>
<levelCrossing id="Sptn_1409L_1423R_5.76"
  pos="0.24"
  absPos="5.760" />
</levelCrossings>
</trackElements>
<ocsElements>
<signals>
<signal id="Sptn_1416"
  pos="0.248"
  absPos="5.768"
  name="1416"
  dir="down">
<speed>
<speedChangeRef ref="Sptn_1409L_X1" />
</speed>
</signal>
<signal id="Sptn_1422"
  pos="0.471"
  absPos="5.951"
  name="1422"
  dir="up">
<speed>
<speedChangeRef ref="Sptn_1409L_X2" />
</speed>
</signal>
</signals>
<trainDetectionElements>
<trackCircuitBorder id="Sptn_1409L_100"
  pos="0.048"
  name="1407T_1416BT"
  absPos="5.568" />
<trackCircuitBorder id="Sptn_1409L_400"
  pos="0.216"

```

```

        name="1416AT_1416BT"
        absPos="5.736" />
<trackCircuitBorder id="Sptn_1409L_600"
  pos="0.247"
  name="1416AT_1416T"
  absPos="5.767" />
<trackCircuitBorder id="Sptn_1409L_2000"
  pos="0.453"
  name="1416T_1423T"
  absPos="5.973"/>
</trainDetectionElements>
</locsElements>
</track>
<track id="Sptn_1407V_1409V"
  description="Sptn verbindingsspoor Zuid">
<trackTopology>
  <trackBegin id="Sptn_1407V"
    pos="0"
    absPos="5.49">
    <connection id="Sptn_1407VR"
      ref="Sptn_1407RV" />
  </trackBegin>
  <trackEnd id="Sptn_1409V"
    pos="0.03"
    absPos="5.52">
    <connection id="Sptn_1409VR"
      ref="Sptn_1409RV" />
  </trackEnd>
  <connections>
  <switch id="Sptn_1407"
    pos="0">
    <connection id="Sptn_1407VL"
      ref="Sptn_1407LV"
      orientation="incoming"
      course="right" />
  </switch>
  <switch id="Sptn_1409"
    pos="0,03">
    <connection id="Sptn_1409VL"
      ref="Sptn_1409LV"
      orientation="outgoing"
      course="left" />
  </switch>
  </connections>
</trackTopology>
<trackElements>
  <gradientChanges>
  <gradientChange id="Sptn_1407V_10000"
    pos="0"
    absPos="5.52"
    dir="up"
    slope="-0.190" />
  </gradientChanges>
</trackElements>
</track>
<track id="Sptn_1423V_1425V"
  description="Sptn verbindingsspoor Noord">
<trackTopology>
  <trackBegin id="Sptn_1423V"
    pos="0"
    absPos="6.04">
    <connection id="Sptn_1423VL"
      ref="Sptn_1423LV" />
  </trackBegin>
  <trackEnd id="Sptn_1425V"
    pos="0.025"
    absPos="6.065">
    <connection id="Sptn_1425VL"
      ref="Sptn_1425LV" />
  </trackEnd>
  <connections>
  <switch id="Sptn_1423"
    pos="0">
    <connection id="Sptn_1423VR"
      ref="Sptn_1407RV"
      orientation="incoming"
      course="left" />
  </switch>
  <switch id="Sptn_1425"
    pos="0.025">
    <connection id="Sptn_1425VR"
      ref="Sptn_1409RV"
      orientation="outgoing"
      course="right" />
  </switch>
  </connections>
</trackTopology>
<trackElements>
  <gradientChanges>
  <gradientChange id="Sptn_1423V_10000"
    pos="0"
    absPos="6.04"
    dir="up" />
  </gradientChanges>
  <speedchanges>
  <speedchange id="Sptn_1425L_X"
    pos="X"
    absPos="6.103"
    vmax="100"
    dir="down" />
  <speedchange id="Sptn_1425L_500"
    pos="X"
    absPos="6.300"
    vmax="80"
    dir="down" />
  </speedchanges>
</trackElements>
<ocsElements>
  <signals>
  <signal id="Sptn_1434"
    pos="X"
    absPos="6.603"
    name="1434"
    dir="down" />
  <signal id="X"
    description="more signals in this track but outside Sptn" />
  </signals>
</trainDetectionElements>
<trackCircuitBorder id="Sptn_1425L_200"
  pos="X"
  name="1423T_1434BT"
  absPos="6.103" />
<trackCircuitBorder id="Sptn_1425L_400"
  pos="X"
  name="1434AT_1434BT"
  absPos="6.300" />
<trackCircuitBorder id="Sptn_1425L_900"
  pos="X"
  name="530BT_1434AT"
  absPos="6.592" />
<trackCircuitBorder id="X"
  description="more traindetectors in this track but outside
  Sptn area" />
</trainDetectionElements>
</locsElements>
</track>
<track id="Bv_36V_Sptn_1427V"
  description="Sptn richting Utg">
<trackTopology>
  <trackBegin id="Bv_36V"
    pos="0"
    slope="0.077" />
  </gradientChanges>
</trackElements>
</track>
<track id="Bv_38aV_Sptn_1425L"
  description="Utg naar Sptn">
<trackTopology>
  <trackBegin id="Bv_38aV"
    pos="0"
    absPos="X">
    <connection id="Bv_38aVR"
      ref="Bv_38aRV" />
  </trackBegin>
  <trackEnd id="Sptn_1425L"
    pos="X"
    absPos="6.065">
    <connection id="Sptn_1425LV"
      ref="Sptn_1425VL" />
  </trackEnd>
  <connections>
  <switch description="switch in Bv area">
  </switch>
  </connections>
</trackTopology>
<trackElements>
  <gradientChanges>
  <gradientChange id="Sptn_1425L_10000"
    pos="X"
    absPos="6.065"
    dir="down"
    slope="0.077" />
  <gradientChange id="Sptn_1425L_1200"
    pos="X"
    absPos="6.603"
    dir="down"
    slope="X"
    description="gradient change cannot be computed since the
    end lies outside Santpoort area" />
  <gradientChange id="Sptn_1425L_11000"
    pos="X"
    absPos="X"
    dir="down"
    slope="X"
    description="perhaps more gradient changes outside Santpoort
    area" />
  </gradientChanges>
  <speedchanges>
  <speedchange id="Sptn_1425L_X"
    pos="X"
    absPos="6.103"
    vmax="100"
    dir="down" />
  <speedchange id="Sptn_1425L_500"
    pos="X"
    absPos="6.300"
    vmax="80"
    dir="down" />
  </speedchanges>
</trackElements>
<ocsElements>
  <signals>
  <signal id="Sptn_1434"
    pos="X"
    absPos="6.603"
    name="1434"
    dir="down" />
  <signal id="X"
    description="more signals in this track but outside Sptn" />
  </signals>
</trainDetectionElements>
<trackCircuitBorder id="Sptn_1425L_200"
  pos="X"
  name="1423T_1434BT"
  absPos="6.103" />
<trackCircuitBorder id="Sptn_1425L_400"
  pos="X"
  name="1434AT_1434BT"
  absPos="6.300" />
<trackCircuitBorder id="Sptn_1425L_900"
  pos="X"
  name="530BT_1434AT"
  absPos="6.592" />
<trackCircuitBorder id="X"
  description="more traindetectors in this track but outside
  Sptn area" />
</trainDetectionElements>
</locsElements>
</track>
<track id="Bv_36V_Sptn_1427V"
  description="Sptn richting Utg">
<trackTopology>
  <trackBegin id="Bv_36V"
    pos="0"

```

```

        absPos="X">
        <connection id="Bv_36VR"
            ref="Bv_36RV" />
    </trackBegin>
    <trackEnd id="Sptn_1427V"
        pos="X"
        absPos="6.113">
        <connection id="Sptn_1427VL"
            ref="Sptn_1427LV" />
    </trackEnd>
    <connections>
        <switch id="Sptn_1427"
            pos="X">
            <connection id="Sptn_1427VR"
                ref="Sptn_1427RV"
                orientation="outgoing"
                course="right" />
        </switch>
        <switch description="switch in Bv area">
    </switch>
    </connections>
</trackTopology>
<trackElements>
<gradientChanges>
<gradientChange id="Sptn_1427V_10000"
    pos="X"
    absPos="6.113"
    dir="down"
    slope="0.056" />
<gradientChange id="Sptn_1427V_900"
    pos="X"
    absPos="6.603"
    dir="down"
    slope="X"
    description="gradient change cannot be computed since the
end lies outside Santpoort area" />
<gradientChange id="Sptn_1427V_11000"
    pos="X"
    absPos="X"
    dir="down"
    slope="X"
    description="perhaps more gradient changes outside Santpoort
area" />
</gradientChanges>
<speedchanges>
<speedchange id="Sptn_1427V_300"
    pos="X"
    absPos="6.223"
    vmax="100"
    dir="down" />
</speedchanges>
</trackElements>
<ocsElements>
<signals>
<signal id="Sptn_1432"
    pos="X"
    absPos="6.603"
    name="1432"
    dir="up" />
<signal id="Sptn_1427L_1300"
    pos="X"
    absPos="6.700"
    dir="down"
    virtual="true"
    description="Open track to station area transition"/>
<signal id="Sptn_1427L_1301"
    pos="X"
    absPos="6.700"
    dir="up"
    virtual="true"
    description="Open track to station area transition"/>
<signal id="X"
    description="more signals in this track but outside Sptn" />
</signals>
<trainDetectionElements>
<trackCircuitBorder id="Sptn_1427V_200"
    pos="X"
    name="1426AT_1427T"
    absPos="6.223" />
<trackCircuitBorder id="Sptn_1427V_700"
    pos="X"
    name="1426AT_1426BT"
    absPos="6.592" />
<trackCircuitBorder id="X"
    description="more traintetectors in this track but outside
Sptn area" />
</trainDetectionElements>
</ocsElements>
</track>
</tracks>
<trackGroups>
<line id="Hlm_137aL_137bV_Utg_207bV_225bV"
    code="Hlm-Utg"
        description="Haarlem - Uitgeest">
        <trackRef ref="Sptn_1425R_1427R" />
        <trackRef ref="Sptn_1405L_1407L" />
        <trackRef ref="Hlm_137aL_Sptn_1407R" />
        <trackRef ref="Hlm_137bV_Sptn_1405V" />
        <trackRef ref="Sptn_1405R_1427L" />
        <trackRef ref="Sptn_1409R_1423L" />
        <trackRef ref="Sptn_1409L_1423R" />
        <trackRef ref="Sptn_1407V_1409V" />
        <trackRef ref="Sptn_1423V_1425V" />
        <trackRef ref="Bv_38aV_Sptn_1425L" />
        <trackRef ref="Bv_36V_Sptn_1427V" />
    </line>
</trackGroups>
</infrastructure>
</railml>

```

Appendix J - Sptn Interlocking in RailML

This appendix shows the main RailML iterations that led to the final interlocking formalization in RailML for Sptn. The interlocking formalization depicts the possible routes starting from signals 1402 and 1404 in the Sptn area (consider Figure 39). First, the initial route based approach RailML is shown. Second, the interlocking formalization based on signal aspect dependencies is presented. The last RailML merges both approaches into a final interlocking RailML proposal of Sptn.

The formalization bases itself on the RailML v2.2 of Appendix I and the OS in Appendix G.

The Route based Interlocking RailML

```
<interlocking>
<signals>
<signal refid="Sptn_1402" Cvps="130" Cvns="130">
<routeActivationRequests>
<routeActivationRequest refid="detector at HIm_505"/>
</routeActivationRequests>
<signalAspectDependencies>
<signalAspectDependency code="R" Vp="130" Vg="0">
<targetSignalTypeRef refid="1422"/>
<targetSignalTypeRef refid="1424"/>
</signalAspectDependency>
<signalAspectDependency code="YFL" Vp="130" Vg="0">
<targetSignalTypeRef refid="1422"/>
<targetSignalTypeRef refid="1424"/>
</signalAspectDependency>
<signalAspectDependency code="YL" Vp="130" Vg="60">
<targetSignalTypeRef refid="Sptn_1422">
<segment refid="Sptn_1402_1422"/>
<routePriorities>
<routePriority1 id="Sptn_1407R_1409L"/>
</routePriorities>
</targetSignalTypeRef>
<targetSignalTypeRef refid="Sptn_1424">
<segment refid="Sptn_1402_1424"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="GRFL6" Vp="130" Vg="60">
<targetSignalTypeRef refid="Sptn_1422">
<segment refid="Sptn_1402_1422"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="YL4" Vp="130" Vg="80">
<targetSignalTypeRef refid="Sptn_1422">
<segment refid="Sptn_1402_1422"/>
<routePriorities/>
</targetSignalTypeRef>
<targetSignalTypeRef refid="Sptn_1424">
<segment refid="Sptn_1402_1424"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="GR" Vp="130" Vg="130">
<targetSignalTypeRef refid="Sptn_1424">
<segment refid="Sptn_1402_1424"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
</signalAspectDependencies>
</signal>
<signal refid="Sptn_1404" Cvps="130" Cvns="130">
<routeActivationRequests>
<routeActivationRequest refid="detector at HIm_501"/>
</routeActivationRequests>
<signalAspectDependencies>
<signalAspectDependency code="R" Vp="130" Vg="0">
<targetSignalTypeRef refid="1422"/>
<targetSignalTypeRef refid="1424"/>
<targetSignalTypeRef refid="1426"/>
</signalAspectDependency>
```

```
</signalAspectDependency>
<signalAspectDependency code="YFL" Vp="130" Vg="0">
<targetSignalTypeRef refid="1422"/>
<targetSignalTypeRef refid="1424"/>
<targetSignalTypeRef refid="1424"/>
</signalAspectDependency>
<signalAspectDependency code="YL" Vp="130" Vg="40">
<targetSignalTypeRef refid="Sptn_1422">
<segment refid="Sptn_1404_1422"/>
<routePriorities/>
</targetSignalTypeRef>
<targetSignalTypeRef refid="Sptn_1424">
<segment refid="Sptn_1404_1424"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="YL" Vp="130" Vg="80">
<targetSignalTypeRef refid="Sptn_1426">
<segment refid="Sptn_1404_1426"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="GRFL" Vp="130" Vg="40">
<targetSignalTypeRef refid="Sptn_1422">
<segment refid="Sptn_1404_1422"/>
<routePriorities/>
</targetSignalTypeRef>
<targetSignalTypeRef refid="Sptn_1424">
<segment refid="Sptn_1404_1424"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="GR" Vp="130" Vg="130">
<targetSignalTypeRef refid="Sptn_1426">
<segment refid="Sptn_1404_1426"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
<signalAspectDependency code="YL8" Vp="130" Vg="130">
<targetSignalTypeRef refid="Sptn_1426">
<segment refid="Sptn_1404_1426"/>
<routePriorities/>
</targetSignalTypeRef>
</signalAspectDependency>
</signalAspectDependencies>
</signal>
</signals>
<segments>
<segment refid="Sptn_1402_1422">
<elements>
<signalRef/>
<trackSection/>
<switchRef/>
<crossingRef/>
<derailerRef/>
<trainDetectorRef/>
<levelCrossingRef/>
</elements>
<flankProtection/>
</segment>
<segment refid="Sptn_1402_1424">
<elements>
<signalRef/>
<trackSection/>
<switchRef/>
<crossingRef/>
<derailerRef/>
<trainDetectorRef/>
<levelCrossingRef/>
</elements>
<flankProtection/>
</segment>
<segment refid="Sptn_1404_1422">
<elements>
<signalRef/>
<trackSection/>
<switchRef/>
<crossingRef/>
<derailerRef/>
<trainDetectorRef/>
<levelCrossingRef/>
</elements>
```

```

</flankProtection/>
</segment>
<segment refid="Sptn_1404_1424">
<elements>
<signalRef/>
<trackSection/>
<switchRef/>
<crossingRef/>
<derailerRef/>
<trainDetectorRef/>
<levelCrossingRef/>
</elements>
</flankProtection/>
</segment>
<segment refid="Sptn_1404_1426">
<elements>
<signalRef/>
<trackSection/>
<switchRef>
<switch refID="Sptn_1405" position="normal"/>
</switchRef>
<crossingRef>
<levelcrossing refID="Hlm_137bV_Sptn_1405V_5.290" beam="down"/>
<levelcrossing refID="Sptn_1405R_1427L_5.76" beam="down"/>
</crossingRef>
<derailerRef/>
<trainDetectorRef>
<trackCircuitBorder refid="Sptn_1405V_100" detection="false">
<trackCircuitBorder refid="Sptn_1405V_300" detection="false">
<trackCircuitBorder refid="Sptn_1405V_600" detection="false">
<trackCircuitBorder refid="Sptn_1405R_200" detection="false">
<trackCircuitBorder refid="Sptn_1405R_400" detection="false">
<trackCircuitBorder refid="Sptn_1405R_600" detection="false">
</trainDetectorRef>
<levelCrossingRef/>
</elements>
</flankProtection/>
</segment>
</segments>
</interlocking>

```

The Signal Aspect Dependency based Interlocking RailML

```

<Signals>
<Signal refID="Sptn_1402" Kv="13" Ka="13" Kann="Kann">
<aspects>
<aspect code="R" target="Sptn_1422" Vp="13" Vz="0" />
<aspect code="R" target="Sptn_1424" Vp="13" Vz="0" />
<aspect code="GLFL" target="Sptn_1422" Vp="13" Vz="0" />
<aspect code="GLFL" target="Sptn_1424" Vp="13" Vz="0" />
<aspect code="GL" target="Sptn_1422" Vp="13" Vz="6" />
<aspect code="GL" target="Sptn_1424" Vp="13" Vz="6" />
<aspect code="GRFL6" target="Sptn_1422" Vp="13" Vz="6" />
<aspect code="GR" target="Sptn_1424" Vp="13" Vz="13" />
<aspect code="GL4" target="Sptn_1422" Vp="13" Vz="8" />
<aspect code="GL4" target="Sptn_1424" Vp="13" Vz="8" />
</aspects>
</Signal>
<Signal refID="Sptn_1404" Kv="13" Ka="13" Kann="Kann">
<aspects>
<aspect code="R" target="Sptn_1422" Vp="13" Vz="0" />
<aspect code="R" target="Sptn_1424" Vp="13" Vz="0" />
<aspect code="R" target="Sptn_1426" Vp="13" Vz="0" />
<aspect code="GLFL" target="Sptn_1422" Vp="13" Vz="0" />
<aspect code="GLFL" target="Sptn_1424" Vp="13" Vz="0" />
<aspect code="GLFL" target="Sptn_1426" Vp="13" Vz="0" />
<aspect code="GL" target="Sptn_1422" Vp="13" Vz="4" />
<aspect code="GL" target="Sptn_1424" Vp="13" Vz="4" />
<aspect code="GL" target="Sptn_1426" Vp="13" Vz="8" />
<aspect code="GRFL" target="Sptn_1422" Vp="13" Vz="4" />
<aspect code="GRFL" target="Sptn_1424" Vp="13" Vz="4" />
<aspect code="GR" target="Sptn_1426" Vp="13" Vz="13" />
<aspect code="GL8" target="Sptn_1426" Vp="13" Vz="13" />
</aspects>
</Signal>
<Signal refID="Sptn_1422" Kv="8" Ka="6" Kann="Kann">
<aspects>
<aspect code="R" target="Bv_527" Vp="6" Vz="0" />
<aspect code="R" target="Bv_529" Vp="6" Vz="0" />
<aspect code="R" target="Bv_527" Vp="4" Vz="0" />
<aspect code="R" target="Bv_529" Vp="4" Vz="0" />
<aspect code="GLFL" target="Bv_527" Vp="6" Vz="0" />
<aspect code="GLFL" target="Bv_529" Vp="6" Vz="0" />
<aspect code="GLFL" target="Bv_527" Vp="4" Vz="0" />
<aspect code="GLFL" target="Bv_529" Vp="4" Vz="0" />
<aspect code="GL" target="Bv_527" Vp="6" Vz="6" />
<aspect code="GL" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GL" target="Bv_527" Vp="4" Vz="4" />
<aspect code="GL" target="Bv_529" Vp="4" Vz="4" />
<aspect code="GL" target="Bv_529" Vp="6" Vz="6" />
<aspect code="GR" target="Bv_527" Vp="6" Vz="6" />
<aspect code="GR" target="Bv_527" Vp="4" Vz="4" />
<aspect code="GL4" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GL4" target="Bv_529" Vp="4" Vz="4" />
<aspect code="GRFL" target="Bv_529" Vp="8" Vz="4" />

```

```

<aspect code="GRFL" target="Bv_529" Vp="4" Vz="4" />
</aspects>
</Signal>
<Signal refID="Sptn_1424" Kv="13" Ka="13" Kann="Kann">
<aspects>
<aspect code="R" target="Bv_527" Vp="6" Vz="0" />
<aspect code="R" target="Bv_529" Vp="6" Vz="0" />
<aspect code="R" target="Bv_527" Vp="4" Vz="0" />
<aspect code="R" target="Bv_529" Vp="4" Vz="0" />
<aspect code="GLFL" target="Bv_527" Vp="6" Vz="0" />
<aspect code="GLFL" target="Bv_529" Vp="6" Vz="0" />
<aspect code="GLFL" target="Bv_527" Vp="4" Vz="0" />
<aspect code="GLFL" target="Bv_529" Vp="4" Vz="0" />
<aspect code="GL" target="Bv_527" Vp="13" Vz="13" />
<aspect code="GL" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GL" target="Bv_527" Vp="4" Vz="4" />
<aspect code="GL" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GL" target="Bv_529" Vp="4" Vz="4" />
<aspect code="GL" target="Bv_529" Vp="4" Vz="4" />
<aspect code="GR" target="Bv_527" Vp="13" Vz="13" />
<aspect code="GR" target="Bv_527" Vp="4" Vz="4" />
<aspect code="GL4" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GL4" target="Bv_529" Vp="4" Vz="4" />
<aspect code="GRFL" target="Bv_529" Vp="8" Vz="4" />
<aspect code="GRFL" target="Bv_529" Vp="4" Vz="4" />
</aspects>
</Signal>
<Signal refID="Sptn_1426" Kv="13" Ka="13" Kann="Kann">
<aspects>
<aspect code="R" target="Bv_529" Vp="8" Vz="0" />
<aspect code="GLFL" target="Bv_529" Vp="8" Vz="0" />
<aspect code="GL" target="Bv_529" Vp="13" Vz="8" />
<aspect code="GR" target="Bv_529" Vp="13" Vz="13" />
<aspect code="GL8" target="Bv_529" Vp="13" Vz="13" />
</aspects>
</Signal>
</Signals>

```

The RailML Group's Approach (signal elements only)

```

<signals>
<signalref="1402" description="special IXL functions of signal"
signalAspectGroupRef="1"/>
<signalref="1404" description="special IXL functions of signal"
signalAspectGroupRef="2"/>
<signalref="1422" description="special IXL functions of signal"
signalAspectGroupRef="3"/>
<signalref="1424" description="special IXL functions of signal"
signalAspectGroupRef="3"/>
<signalref="1426" description="special IXL functions of signal"
signalAspectGroupRef="4"/>
</signals>
<signalTypes>
<signalType="MS:Entry"/>
<signalType="MS:Exit"/>
</signalTypes>
<signalAspects>
<signalAspect="R" name="red" signalspeed="0" targetspeed="0"/>
<signalAspect="YLFL" name="yellow flashing"
signalspeed="0" targetspeed="0"/>
<signalAspect="YL" name="yellow" targetspeed="0"/>
<signalAspect="YL8" name="yellow 80km/h" targetspeed="40"/>
<signalAspect="GR" name="green" targetspeed="max"/>
<signalAspect="GRFL" name="green flashing" targetspeed="40"/>
</signalAspects>
<signalAspectGroups>
<signalAspectGroup="1">
<signalAspectRef="R"/>
<signalAspectRef="YLFL"/>
<signalAspectRef="YL"/>
<signalAspectRef="GRFL"/>
<signalAspectRef="YL8"/>
<signalAspectRef="GR"/>
</signalAspectGroup>
<signalAspectGroup="2">
<signalAspectRef="R"/>
<signalAspectRef="YLFL"/>
<signalAspectRef="YL"/>
<signalAspectRef="GRFL"/>
</signalAspectGroup>
<signalAspectGroup="3">
<signalAspectRef="R"/>
<signalAspectRef="YLFL"/>
<signalAspectRef="YL"/>
<signalAspectRef="GR"/>
</signalAspectGroup>
<signalAspectGroup="4">
<signalAspectRef="R"/>
<signalAspectRef="YLFL"/>
<signalAspectRef="YL"/>
<signalAspectRef="YL8"/>
<signalAspectRef="GR"/>
</signalAspectGroup>
</signalAspectGroups>
<signalAspectDependencies>

```



```

</signalAspectDependency>
<signalAspectDependency code="GRFL" sVpo="80" pVo="40">
  <targetRef refid="Bv_529"/>
</signalAspectDependency>
<signalAspectDependency code="GRFL" sVpo="40" pVo="40">
  <targetRef refid="Bv_529"/>
</signalAspectDependency>
<signalAspectDependency code="GR" sVpo="60" pVo="60">
  <targetRef refid="Bv_527"/>
</signalAspectDependency>
<signalAspectDependency code="GR" sVpo="40" pVo="40">
  <targetRef refid="Bv_527"/>
</signalAspectDependency>
<signalAspectDependency code="YL4" sVpo="80" pVo="40">
  <targetRef refid="Bv_529"/>
</signalAspectDependency>
<signalAspectDependency code="YL4" sVpo="40" pVo="40">
  <targetRef refid="Bv_529"/>
</signalAspectDependency>
</signalAspectDependencies>
</signal>
<signal refid="Sptn_1424">
  <routeActivationRequests>
    <routeActivationRequest refid="detector at Hlm_501"/>
  </routeActivationRequests>
  <signalAspectDependencies>
    <signalAspectDependency code="R" sVpo="60" pVo="0">
      <targetRef refid="Bv_527"/>
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="R" sVpo="40" pVo="0">
      <targetRef refid="Bv_527"/>
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YLFL" sVpo="60" pVo="0">
      <targetRef refid="Bv_527"/>
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YLFL" sVpo="40" pVo="0">
      <targetRef refid="Bv_527"/>
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL" sVpo="130" pVo="130">
      <targetRef refid="Bv_527"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL" sVpo="80" pVo="40">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL" sVpo="40" pVo="40">
      <targetRef refid="Bv_527"/>
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="GRFL" sVpo="80" pVo="40">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="GRFL" sVpo="40" pVo="40">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="GR" sVpo="130" pVo="130">
      <targetRef refid="Bv_527"/>
    </signalAspectDependency>
    <signalAspectDependency code="GR" sVpo="40" pVo="40">
      <targetRef refid="Bv_527"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL4" sVpo="80" pVo="40">
      <targetRef refid="Sptn_1427V_300"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL4" sVpo="40" pVo="40">
      <targetRef refid="Sptn_1427V_300"/>
    </signalAspectDependency>
  </signalAspectDependencies>
</signal>
<signal refid="Sptn_1426">
  <routeActivationRequests>
    <routeActivationRequest refid="detector at Hlm_501"/>
  </routeActivationRequests>
  <signalAspectDependencies>
    <signalAspectDependency code="R" sVpo="80" pVo="0">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YLFL" sVpo="80" pVo="0">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL" sVpo="130" pVo="80">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="GR" sVpo="130" pVo="130">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
    <signalAspectDependency code="YL8" sVpo="130" pVo="130">
      <targetRef refid="Bv_529"/>
    </signalAspectDependency>
  </signalAspectDependencies>
</signal>
</signal>
</signals>
<speedChangeSigns>
  <speedChangeSign refid="Sptn_1427V_300">
    <speedChangeProfile sVpo="40" pVo="0" eVo="40" eVno="80">
      <targetRef="Bv_529"/>
    </speedChangeProfile>
  </speedChangeSign>
</speedChangeSigns>
<segments>
  <segment id="Sptn_1402_1422">
    <elements>
      <signalRef/>
      <trackSection/>
      <switchRef/>
      <crossingRef/>
      <derailerRef/>
      <trainDetectorRef/>
      <levelCrossingRef/>
    </elements>
    <flankProtection/>
    <routePriorities/>
  </segment>
  <segment id="Sptn_1402_1424">
    <elements>
      <signalRef/>
      <trackSection/>
      <switchRef/>
      <crossingRef/>
      <derailerRef/>
      <trainDetectorRef/>
      <levelCrossingRef/>
    </elements>
    <flankProtection/>
    <routePriorities/>
  </segment>
  <segment id="Sptn_1404_1422">
    <elements>
      <signalRef/>
      <trackSection/>
      <switchRef/>
      <crossingRef/>
      <derailerRef/>
      <trainDetectorRef/>
      <levelCrossingRef/>
    </elements>
    <flankProtection/>
    <routePriorities/>
  </segment>
  <segment id="Sptn_1404_1424">
    <elements>
      <signalRef/>
      <trackSection/>
      <switchRef/>
      <crossingRef/>
      <derailerRef/>
      <trainDetectorRef/>
      <levelCrossingRef/>
    </elements>
    <flankProtection/>
    <routePriorities/>
  </segment>
  <segment id="Sptn_1404_1426">
    <elements>
      <signalRef/>
      <trackSection/>
      <switchRef>
        <switch refid="Sptn_1405" position="normal"/>
      </switchRef>
      <crossingRef>
        <levelcrossing refid="Hlm_137bV_Sptn_1405V_5.290" beam="down"/>
        <levelcrossing refid="Sptn_1405R_1427L_5.76" beam="down"/>
      </crossingRef>
      <derailerRef/>
      <trainDetectorRef>
        <trackCircuitBorder refid="Sptn_1405V_100" detection="false"/>
        <trackCircuitBorder refid="Sptn_1405V_300" detection="false"/>
        <trackCircuitBorder refid="Sptn_1405V_600" detection="false"/>
        <trackCircuitBorder refid="Sptn_1405R_200" detection="false"/>
        <trackCircuitBorder refid="Sptn_1405R_400" detection="false"/>
        <trackCircuitBorder refid="Sptn_1405R_600" detection="false"/>
      </trainDetectorRef>
      <levelCrossingRef/>
    </elements>
    <flankProtection/>
    <routePriorities>
      <routePriority id="Sptn_1405R"/>
    </routePriorities>
  </segment>
</segments>
</interlocking>

```

Appendix K - RailML's Data Exchange Coverage

This appendix shows the input and output (data) products for each process in the interlocking engineering design process. Furthermore, the tables indicate to which extent RailML could exchange the data.

Koelewijn et al. (2013), Storck and Dragt (2013) and Middelkoop (2013). RailML can currently exchange the data marked in green. RailML could potentially exchange the data in orange via a calculation tool. RailML does not have any means yet to exchange data in red. The grey products can never be exchange by RailML since it does not concern data.

The main interlocking related input and output (data) products on the basis of Appendix D, Prorail (2010d),

Aggregated design process	Input product	Output product
Client Requirement Specification	Topology of current interlocking area	Situation boundaries
	Timetable / Planned infra usage	Solution requirements
	Problem Statement	Stakeholder analysis
FIS Generation of Solution Alternatives	Situation boundaries	Concept track topologies
	Solution requirements	
	Stakeholder analysis	Trade-off matrix
	Decision making report	
FIS System Requirements	Concept track topologies FIS	System decisions like assumptions and requirements
		Updated CRS
		Legal aspects
		Desired track topology map
		GIS map
	Prorail topology preference	Longitudinal maps
		Braking curve calculations
		Travel time calculations
		Consecutive train time calculations
		Axle load calculations
RVTO Statement of Final Design	System decisions like assumptions and requirements	Project management plan
	Updated CRS	Electric circuit maps
	Legal aspects	Overhead wire characteristics maps
	Desired track topology map	
	GIS map	Power supply maps
	Longitudinal maps	
	Braking curve calculations	Updated braking curve calculations
	Travel time calculations	
	Consecutive train time calculations	Level crossing maps
	Axle load calculations	
RVTO Specification of the Interlocking Area	Project management plan	"Verkeerstechnisch Overzicht Baan en Emplacement" (VT-OBE) map
	Desired track topology map	"Overzichtstekeningen Seinbeelden" (OS) map
	GIS map	Updated GIS maps
	Longitudinal maps	
	Updated Braking curve calculations	Updated longitudinal maps
	Travel time calculations	
	Consecutive train time calculations	Cross section maps
	Axle load calculations	
	Electric circuit maps	Overview of speed restrictions
	Overhead wire characteristics maps	
	Power supply maps	RVTO testing procedures
	Level crossing maps	RVTO report
	SWOD Design OR, OBE and SVA maps	"Verkeerstechnisch Overzicht Baan en Emplacement" (VT-OBE) map
"Overzichtstekeningen Seinbeelden" (OS) map		
Updated GIS maps		
Updated longitudinal maps		
Updated braking curve calculations		"Overzicht Baan en Emplacement" (OBE) map
Overview of speed restrictions		
RVTO report		
Cross section maps		

	Electric circuit maps Overhead wire characteristics maps Power supply maps Level crossing maps	"Staat Van Aanwijzingen" (SVA) report
Develop Test Protocol	OS OBE SVA	Interlocking test protocol for BITS
Dry test	Interlocking test protocol for BITS Interlocking system test report from industry Updated EVP	Interlocking test results Prorail Final EVP

The input and output data products for each process in the engineering design process (Kanoun, 2003, 2011). RailML can currently exchange the data marked in green. RailML could potentially exchange the data in orange via a calculation tool. RailML does not have any means yet to ex-

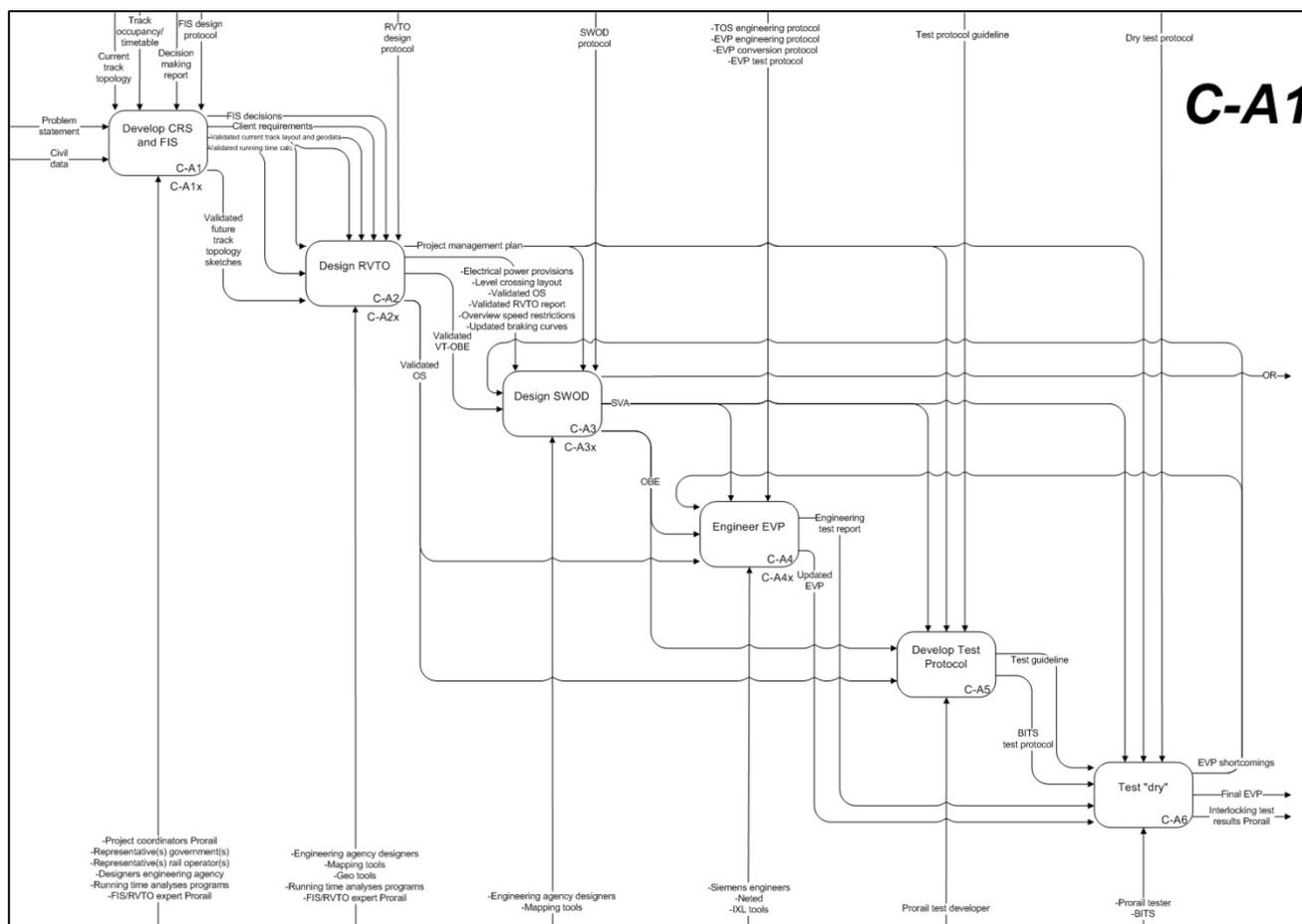
change data in red. The grey products can never be exchanged by RailML since it does not concern data.

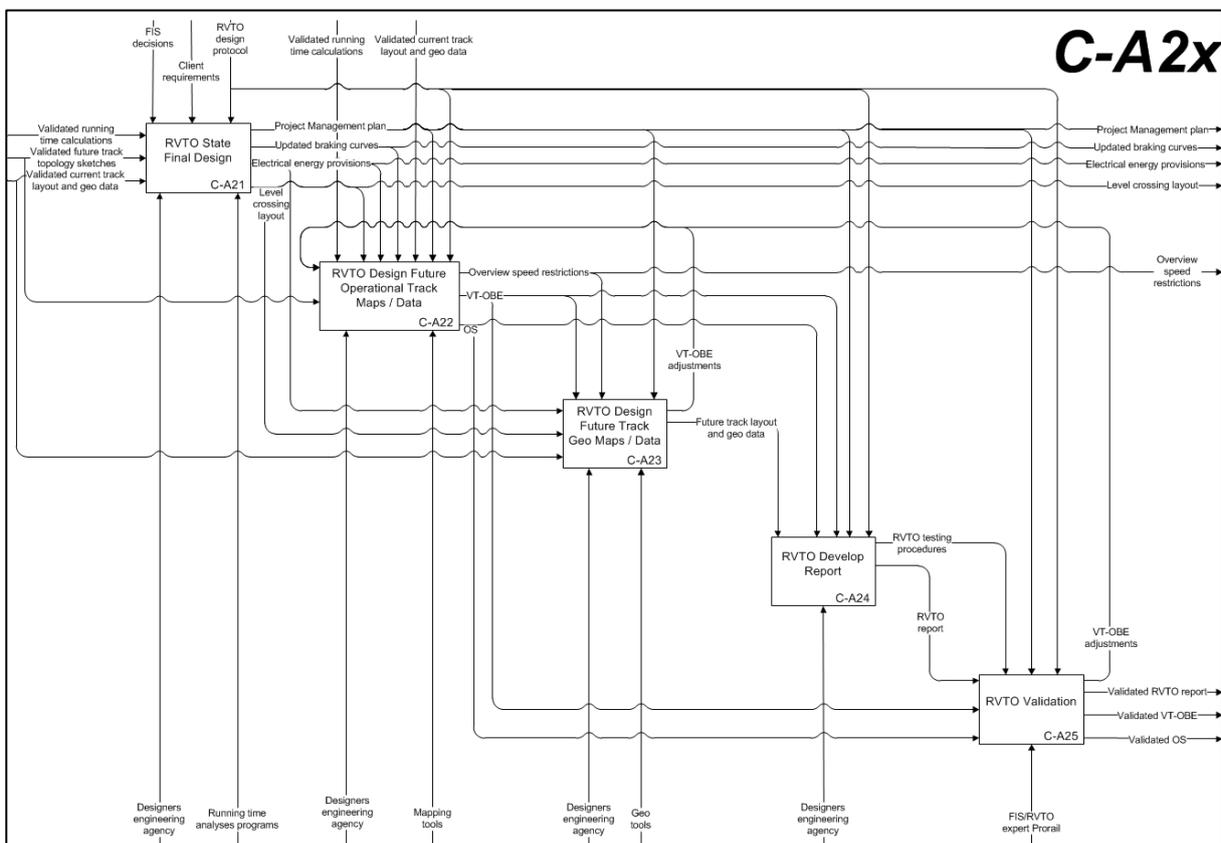
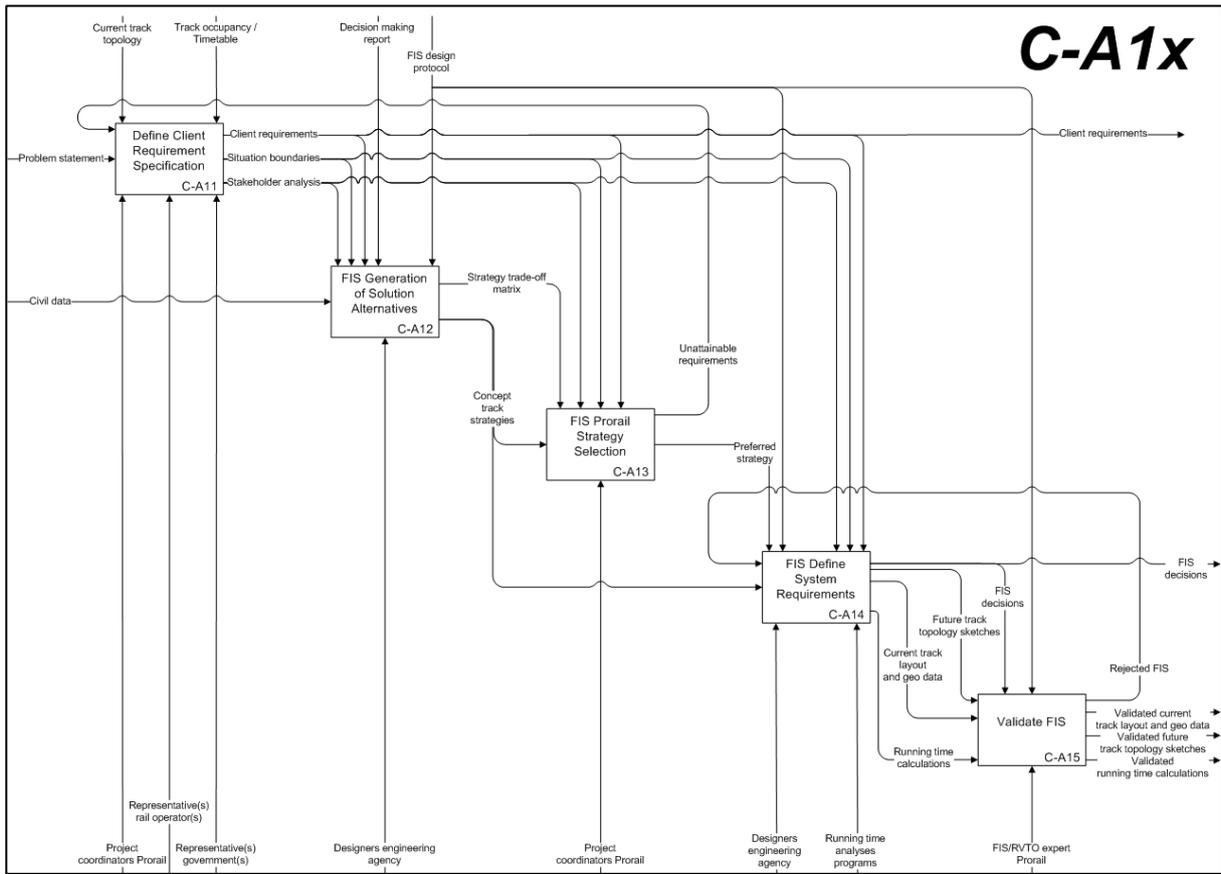
Aggregated engineering process	Input product	Output product
Data preparation	OBE	Machine readable OBE
	OS	Validated and machine readable OS
EVP engineering	Machine readable OBE	EVP
	Validated and machine readable OS	Interlocking RailML
EVP conversion	SVA	IXL channels
	EVP	IXL towers IXL software IXL IT architecture
EVP test	EVP	Updated EVP
	IXL channels	Test report
	IXL towers	
	IXL software	
IXL IT architecture		
Dry test	Updated EVP	Final EVP
	Interlocking RailML	Interlocking test results Prorail
	Interlocking system test report from Siemens	
	Interlocking test protocol for BITS	

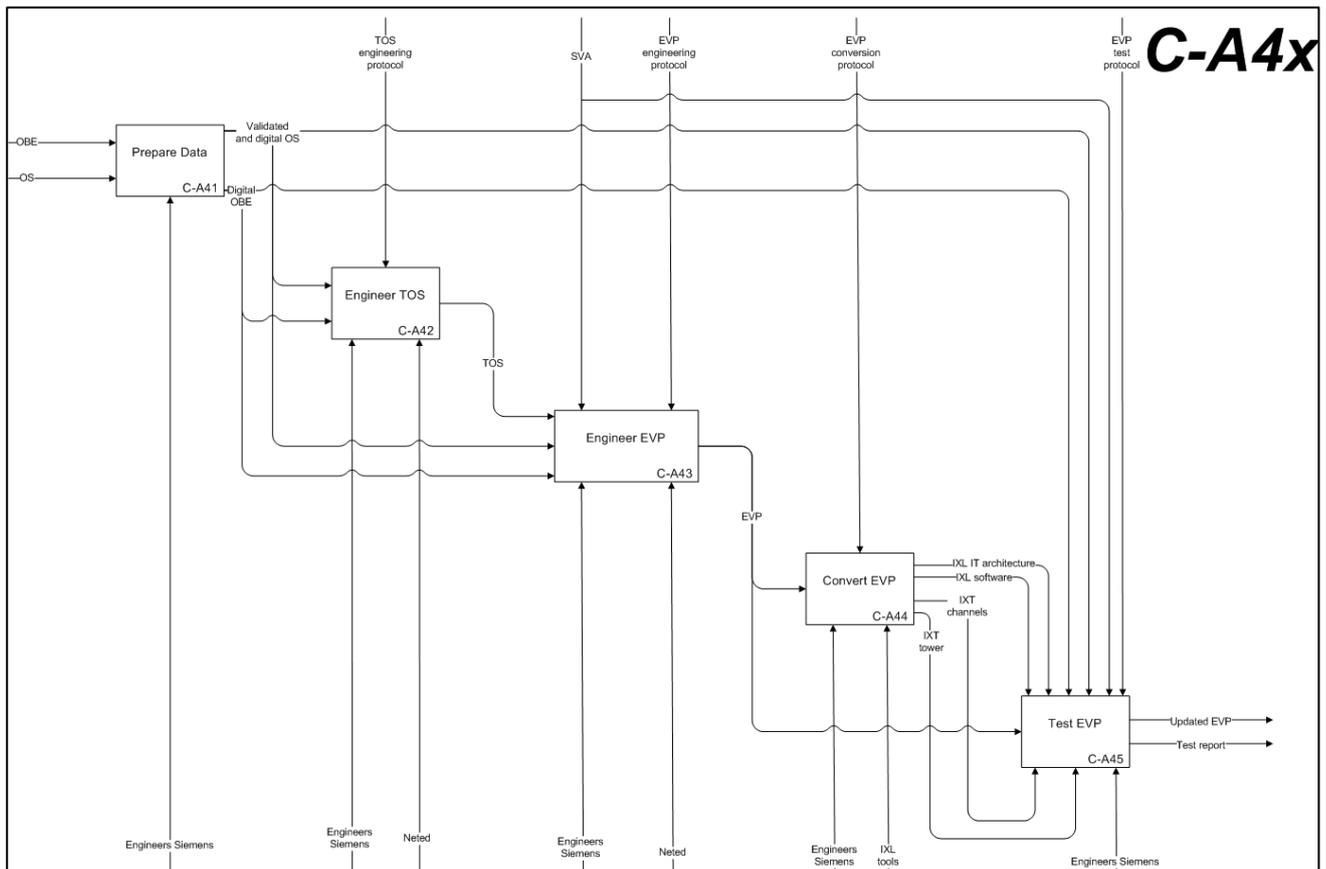
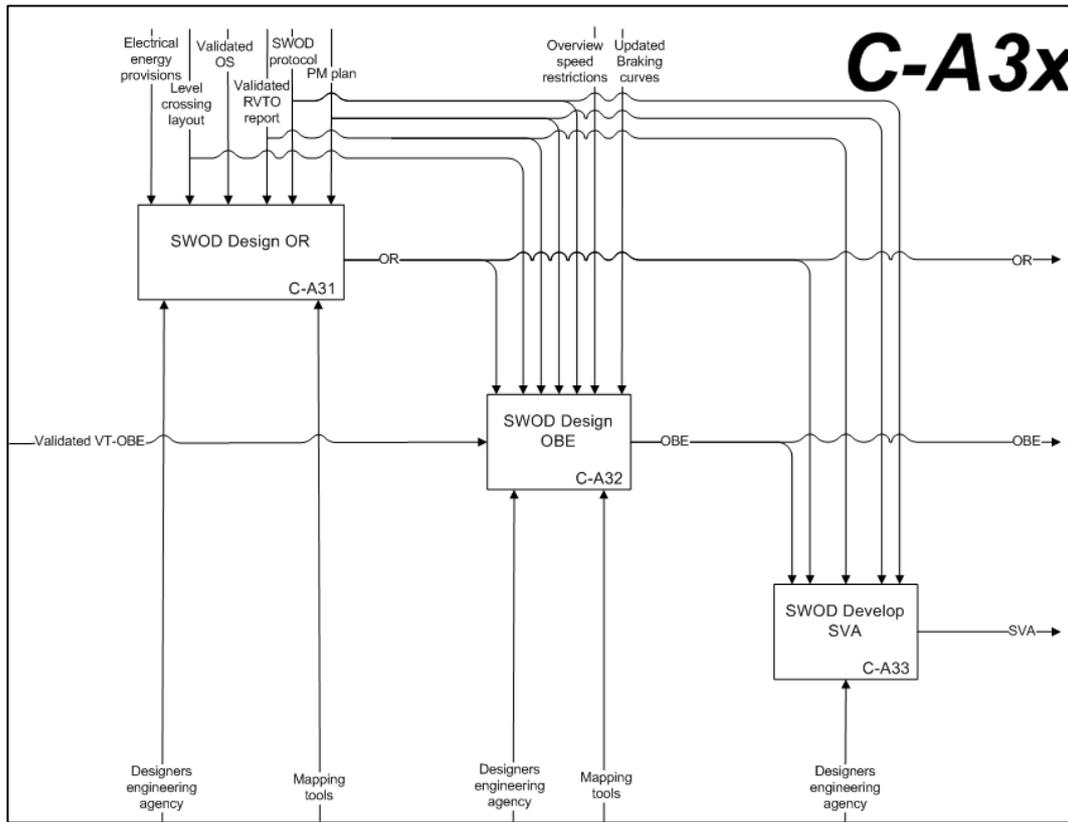
Appendix L - ARENA Model Setup

This appendix covers IDEF0 conceptualization, model verification, model validation and model parameters of the final simulation model.

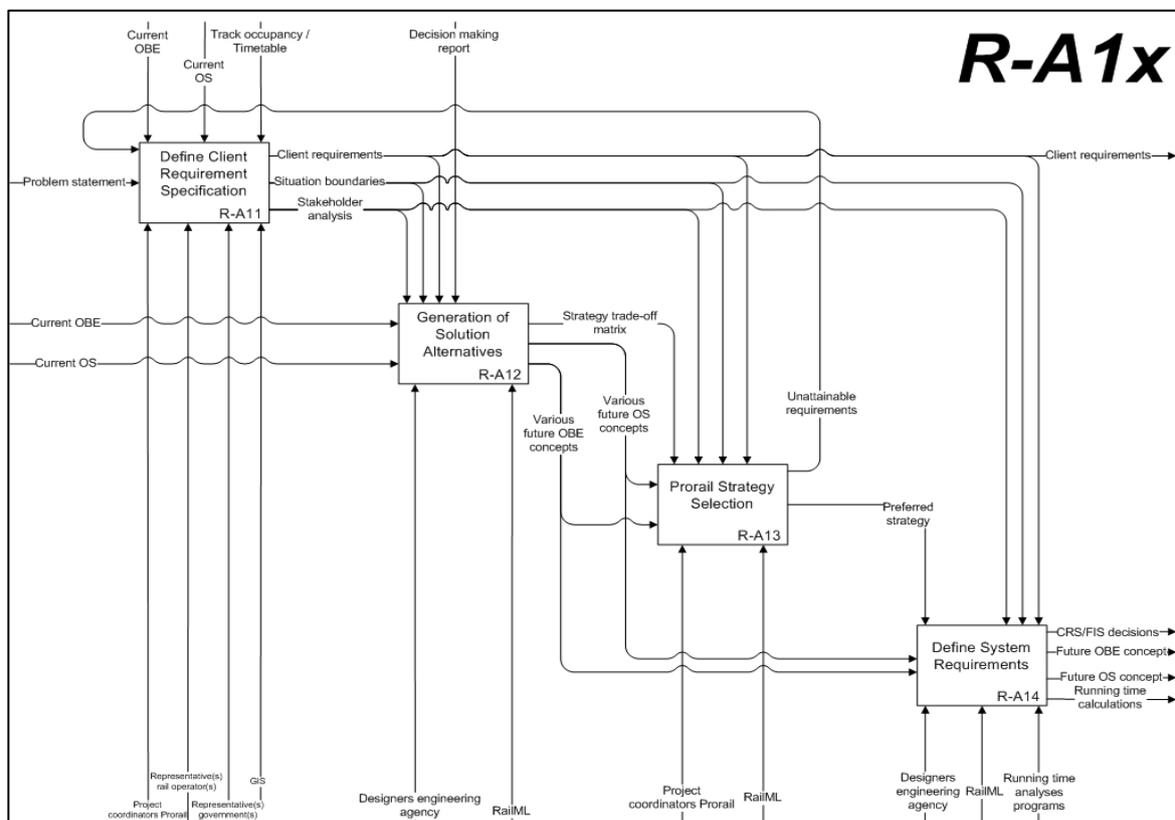
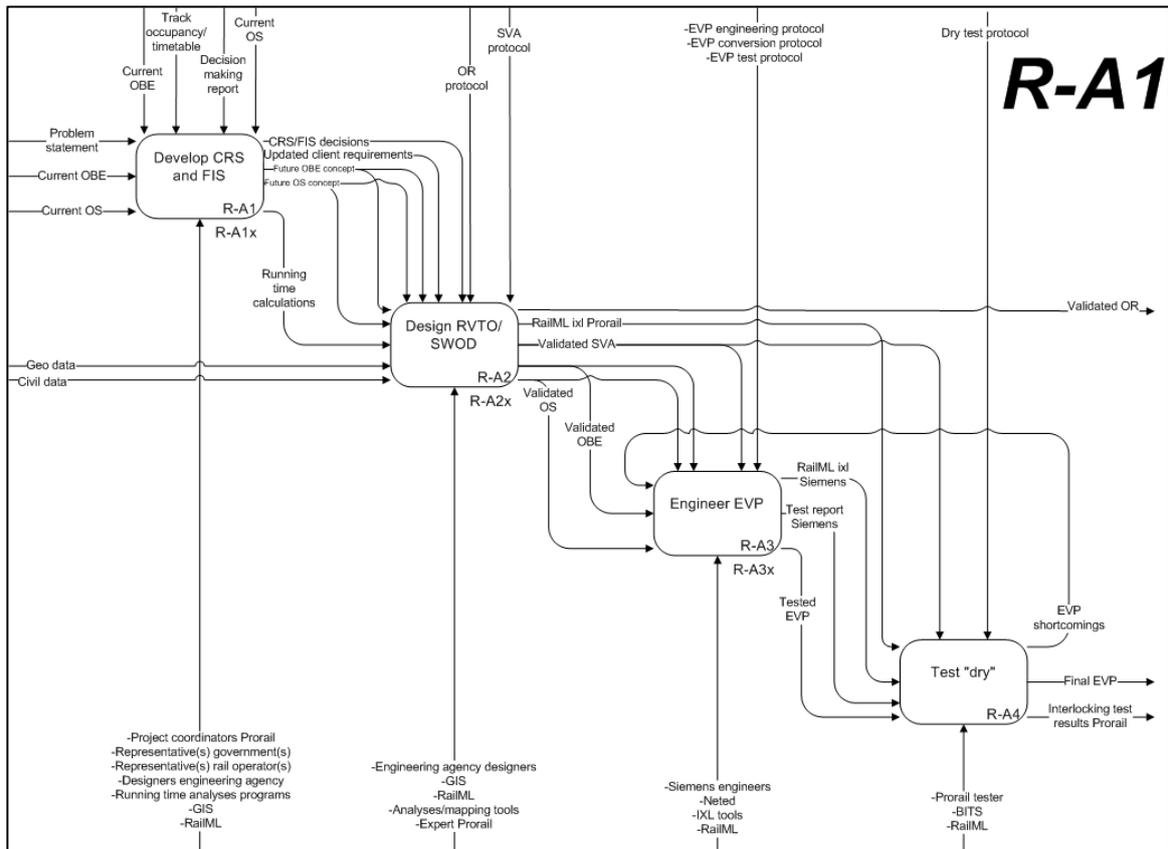
Model Conceptualization Status Quo Model

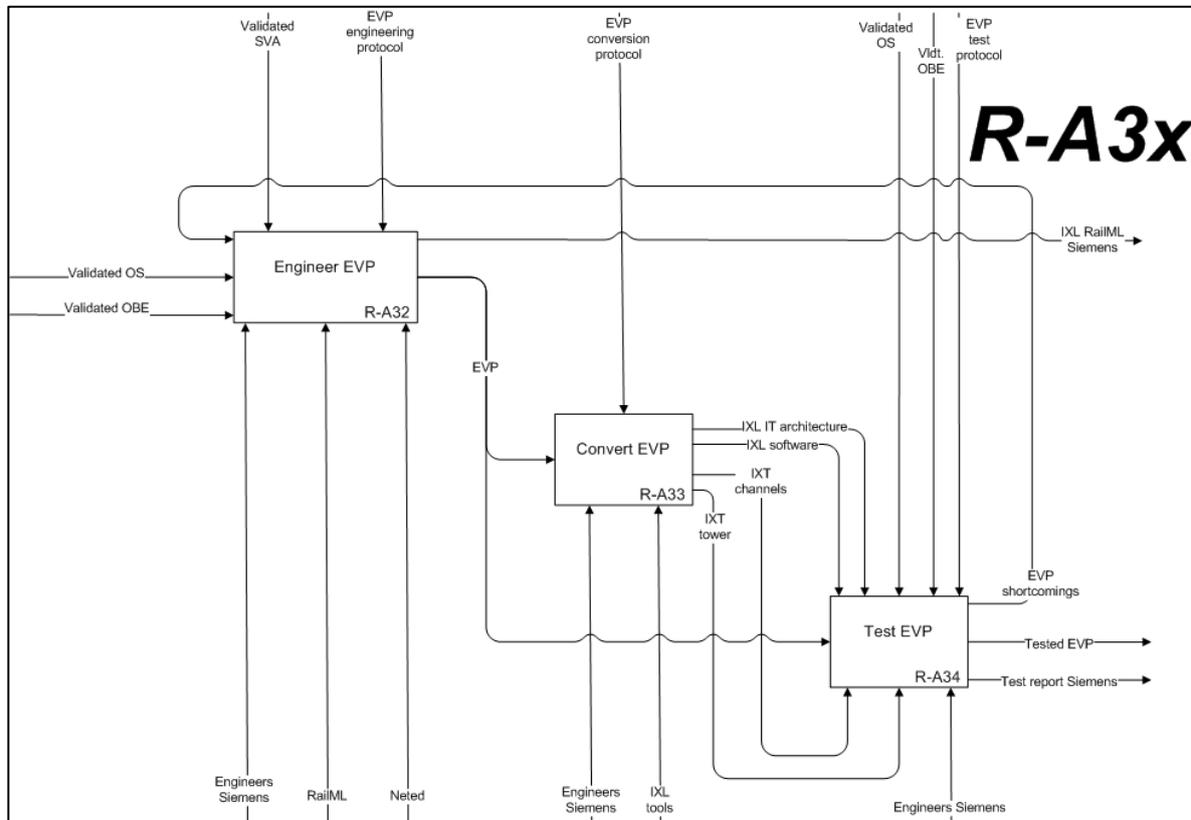
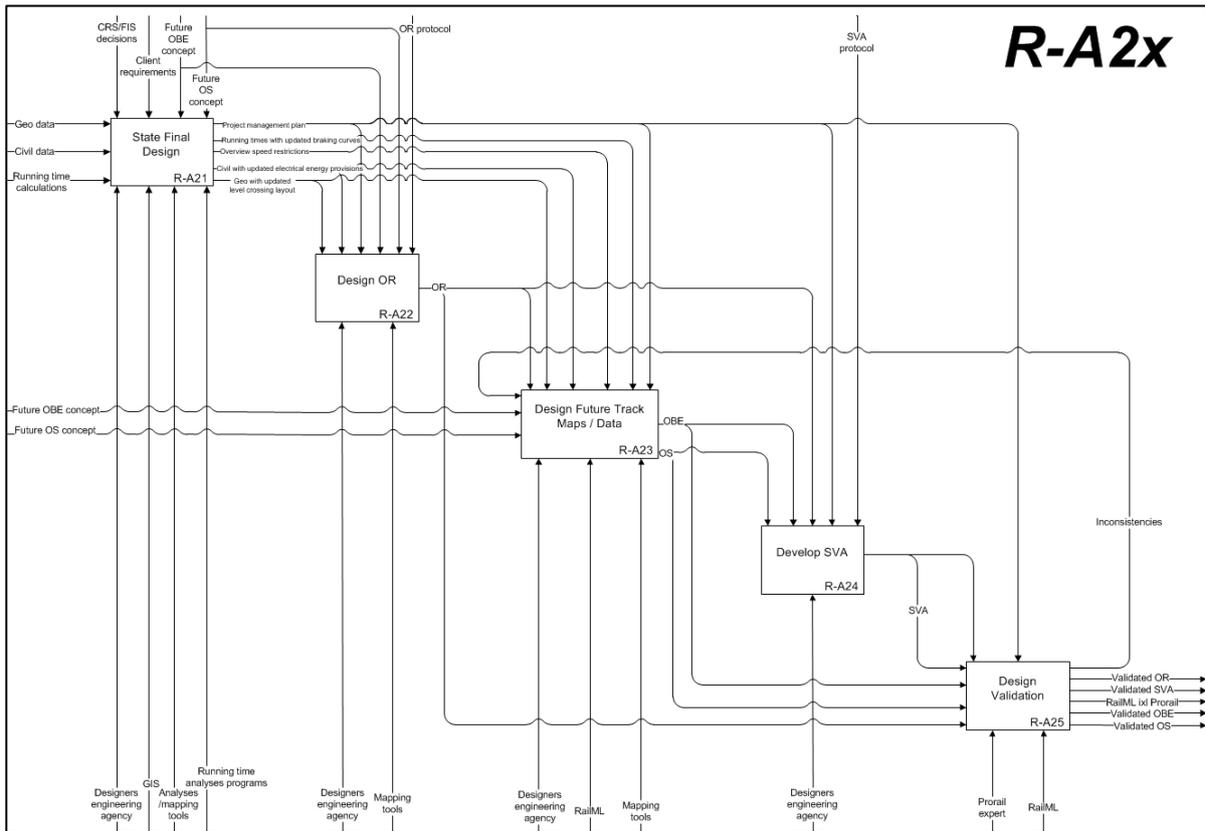




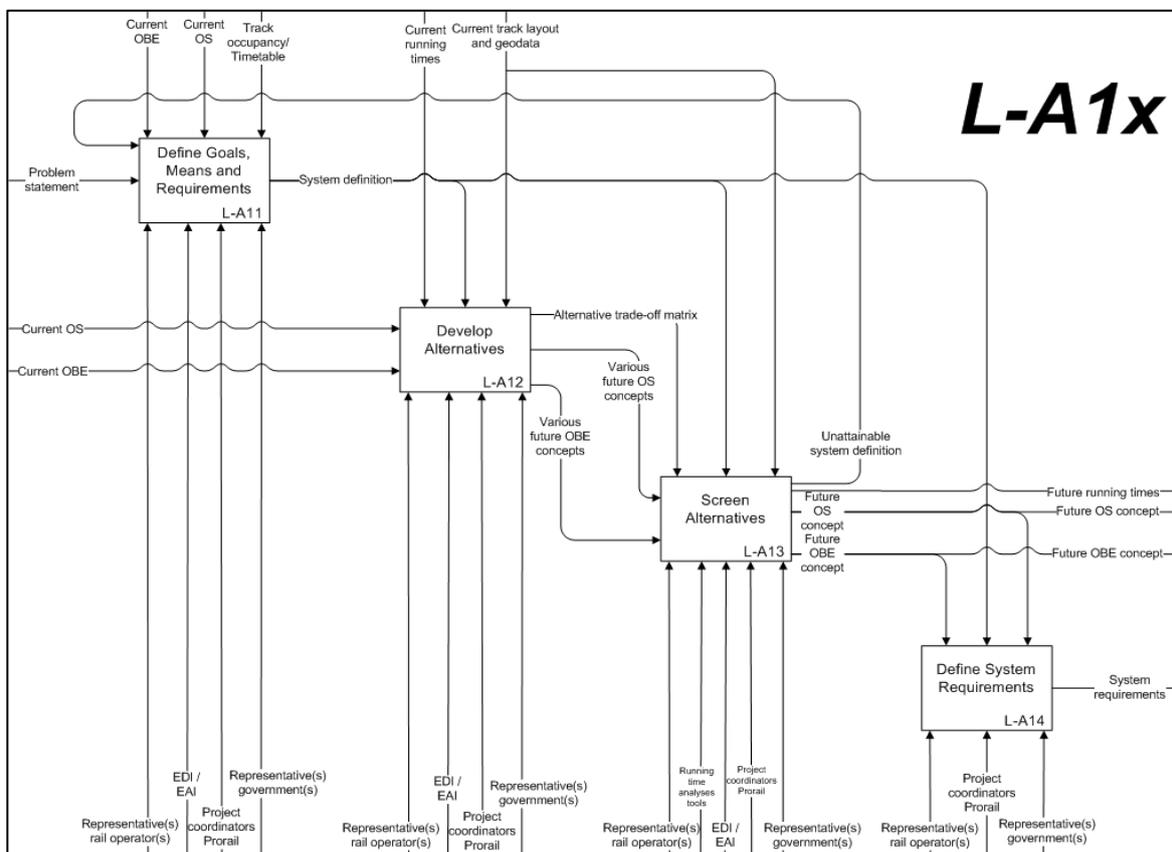
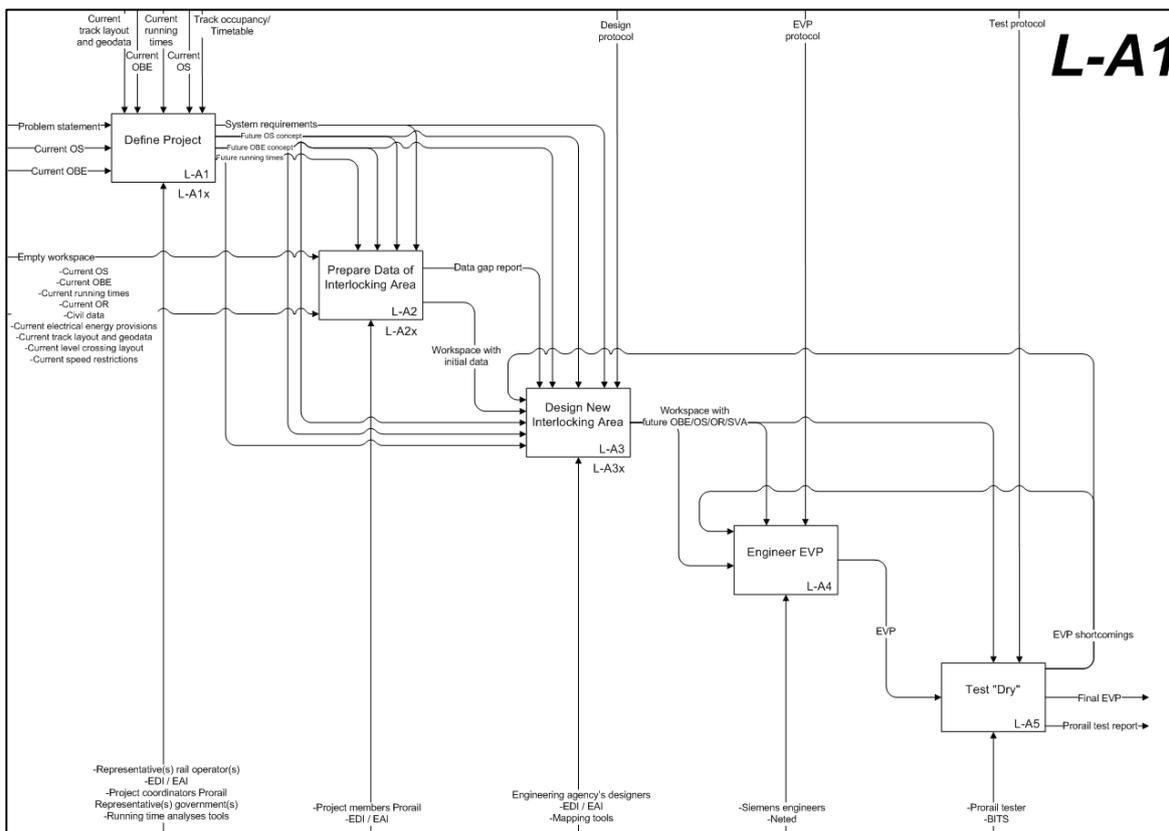


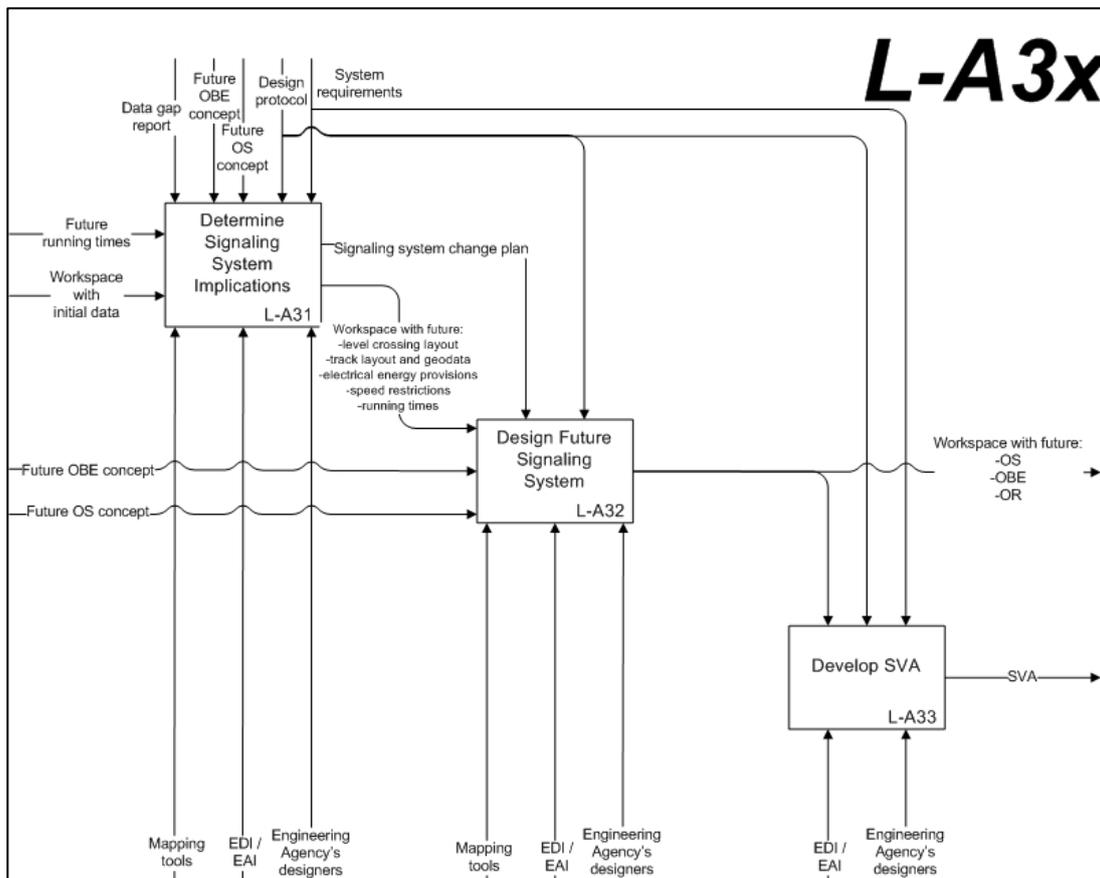
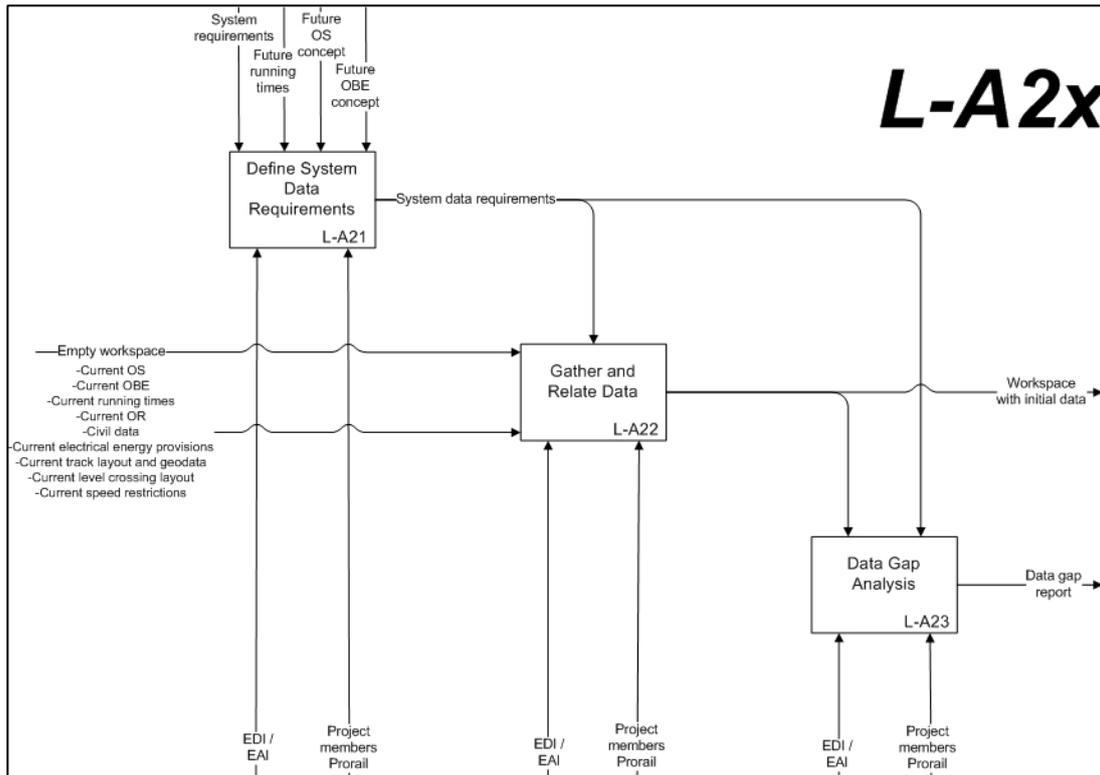
RailML Model





Lean Model





Model Verification

This part compares the previous with the ARENA model to ensure that the model encompasses all relevant processes. The next discusses the main differences for the base model (current situation), the RailML model and the perfect lean model. Verification for a discrete simulation model includes whether the model contains the correct input parameters, the correct dimensions, whether the model logic makes sense and whether the processes perform the right calculations (Thissen, Phaff, & Pruyt, 2008; Verbraeck & Valentin, 2006). This appendix uses the same classification of processes as in the previous SADTs.

The Base Model

C-A0) General:

- the model includes the cycle and idle times exactly as stated in the value stream map of Appendix D. Furthermore, the cycle times of Siemens' engineering processes correspond to the computed averages per project complexity level and engineering process as stated in the specification appendix. In addition, all processes follow the normal distribution in which the standard deviation always corresponds to 10% of the cycle and idle times (the mean);
- the model calculates process times as expected. For that purpose, each process is checked on containing the right distribution. Furthermore, the outcomes per process are investigated on mean, minimum and maximum. Some deviating minima and maxima occur every once in a while but this could happen in reality too;
- the model parameters are all expressed in days. The conversion from weeks or workweeks to days does not contain errors;
- the model only contains monetary values for the engineering processes of Siemens since Prorail could not provide monetary figures;
- the model includes one type of resources and does not distinguish for the variety of process supporting resources. The lack of data makes it impossible to define the exact number of specific resources per process. Therefore, the model assigns a project team to each project. The number of resources does however not reflect an actual number of project teams; let alone the number of specific resources. In reality, project teams deal with multiple projects at the time. There is however no data on the amount of projects a project team handles;
- the model specifies entities as projects and not as individual (data) products. In most of the cases where the output contains multiple products, those provide sufficient input to directly start the next process. As a result, there is no need to include more detail. Only in some cases that is not the case, e.g. at the parallel processes during the RVTO and SWOD. In that case an ARENA batch module provides the if-else start trigger for the next process.

C-A1x) Develop Client Requirement Specification and "Functioneel Integraal Systeemontwerp":

- the model does not include the arrival of civil data for use in the generation of solution alternatives process. It is assumed that the civil data is always available as it is a necessity for construction and spatial design;
- the model does not include the arrival of current track topology maps, a decision making report, track occupancy data and a FIS design protocol that serve as a trigger for the CRS and alternative development processes. The model assumes that the triggers are always available at the arrival of a project as they form the cause to start an engineering design process;
- the model does not include a possible reconsideration of the CRS and generated alternatives due to unattainable requirements. In reality, this feedback process is strongly iterative. This means that each project makes several cycles between CRS and alternative development before Prorail can select one final alternative. The amount of cycles is however hard to estimate and even harder to model. The process times already reflect that iterative behavior. Therefore, a somewhat worse performance estimation is chosen over the probability for structural modeling errors.

C-A2x) Design the "RailVerkeersTechnisch Ontwerp":

- the model does not include the arrival of the FIS decisions, the client requirements, RVTO design protocol, validated running time calculations and validated current track layout and geo data that serve as a trigger for various RVTO processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals;
- the model does not include a possible reconsideration of the VT-OBE due to a mismatch with the geo maps / data. The development of a VT-OBE and geo maps / data goes in parallel and the data from Prorail already reflects the various alignments between both processes.

C-A3x) Design "SeinWezen OverzichtsDossier"

- the model does not include the arrival of the electrical energy provisions, level crossing layout, validated OS, validated RVTO report, SWOD protocol, project management plan, overview speed restrictions and updated braking curves that serve as a trigger for the various SWOD processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals;
- the model does not include a separate end / disposal module for the OR result. The OR does not provide any further contribution to the engineering design process of interlocking systems.

C-A4x) EVP engineering

- the model does not include the arrival of the TOS engineering protocol, SVA, EVP engineering protocol, EVP conversion protocol and EVP test pro-

protocol that serve as a trigger of various EVP engineering processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals.

C-A5) Develop test protocol

- the model does not include the arrival of the test protocol guideline that serves as a trigger for the test protocol development. The model assumes that the trigger is always available at the arrival of a project since it concerns an archived instruction manual.

C-A6) Test "Dry"

- the model does not include the arrival of the dry test protocol that serves as a trigger for the dry test process. The model assumes that the trigger is always available at the arrival of a project since it concerns an archived instruction manual.

The RailML Model

R-A0) General:

- the worse case model includes the cycle and idle times exactly as stated in the value stream map of Appendix D. Furthermore, the cycle times of Siemens' engineering processes correspond to the computed averages per project complexity level and engineering process as stated in the specification appendix. In addition, all processes follow the normal distribution in which the standard deviation always corresponds to 10% of the cycle and idle times (the mean);
- the best case model alters some parameters. Table 16 shows those changes;
- the model calculates process times as expected. For that purpose, each process is checked on containing the right distribution. Furthermore, the outcomes per process are investigated on mean, minimum and maximum. Some deviating minima and maxima occur every once in a while but this could happen in reality too;
- the model parameters are all expressed in days. The conversion from weeks or workweeks to days does not contain errors;
- the model only contains monetary values for the engineering processes of Siemens since Prorail could not provide monetary figures;
- the model includes one type of resources and does not distinguish for the variety of process supporting resources. The lack of data makes it impossible to define the exact number of specific resources per process. Therefore, the model assigns a project team to each project. The number of resources does however not reflect an actual number of project teams; let alone the number of specific resources. In reality, project teams deal with multiple projects at the time. There is however no data on the amount of projects a project team handles;
- the model specifies entities as projects and not as individual (data) products. In most of the cases where the output contains multiple products, those provide sufficient input to directly start the

next process. As a result, there is no need to include more detail. Only in some cases that is not the case, e.g. at the parallel processes during the RVTO and SWOD. In that case an ARENA batch module provides the if-else start trigger for the next process.

R-A1x) Develop Client Requirement Specification and "Functioneel Integraal Systeemontwerp":

- the model does not include the arrival of current OBE and OS for use in the generation of solution alternatives process. It is assumed that those files are always available in the case of a modification project;
- the model does not include the arrival of current OBE and OS maps, current track layout and a decision making report that serve as a trigger for the CRS and alternative development processes. The model assumes that the triggers are always available at the arrival of a project as they form the cause to start an engineering design process;
- the model does not include a possible reconsideration of the CRS and generated alternatives due to unattainable requirements. In reality, this feedback process is strongly iterative. This means that each project makes several cycles between CRS and alternative development before Prorail can select one final alternative. The amount of cycles is however hard to estimate and even harder to model. The process times already reflect that iterative behavior. Therefore, a somewhat worse performance estimation is chosen over the probability for structural modeling errors.

R-A2x) Design the "RailVerkeersTechnisch Ontwerp"/"SeinWezen OverzichtsDossier":

- the model does not include the arrival of the CRS/FIS decisions, the client requirements, OR design protocol, SVA protocol, future OBE concept and future OS concept that serve as a trigger for various RVTO processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals;
- the model does not include a separate end / disposal module for the validated OR result. The OR does not provide any further contribution to the engineering design process of interlocking systems.

R-A3x) EVP engineering

- the model does not include the arrival of the validated SVA, EVP engineering protocol, EVP conversion protocol, EVP test protocol, validated OBE and validated OS that serve as a trigger of various EVP engineering processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals.

R-A4) Test "Dry"

- the model does not include the arrival of the dry test protocol that serves as a trigger for the dry test process. The model assumes that the trigger is always available at the arrival of a project since it concerns an archived instruction manual.

The Lean Design

L-A0) General:

- the model includes the cycle and idle times exactly as stated in the value stream map of Appendix D. Furthermore, the cycle times of Siemens' engineering processes correspond to the computed averages per project complexity level and engineering process as stated in the specification appendix. In addition, all processes follow the normal distribution in which the standard deviation always corresponds to 10% of the cycle and idle times (the mean);
- the model calculates process times as expected. For that purpose, each process is checked on containing the right distribution. Furthermore, the outcomes per process are investigated on mean, minimum and maximum. Some deviating minima and maxima occur every once in a while but this could happen in reality too;
- the model parameters are all expressed in days. The conversion from weeks or workweeks to days does not contain errors;
- the model only contains monetary values for the engineering processes of Siemens since Prorail could not provide monetary figures;
- the model includes various types of resources and does distinguish for the variety of process complexity types. The lack of data makes it impossible to define the exact number of specific resources per process. Therefore, the model assigns a project team to each project. The number of resources does however not reflect an actual number of project teams; let alone the number of specific resources. In reality, project teams deal with multiple projects at the time. There is however no data on the amount of projects a project team handles;
- the model specifies entities as projects and not as individual (data) products. In most of the cases where the output contains multiple products, those provide sufficient input to directly start the next process. As a result, there is no need to include more detail. Only in some cases that is not the case, e.g. at the parallel processes during the RVTO and SWOD. In that case an ARENA batch module provides the if-else start trigger for the next process.

L-A1x) Define project:

- the model does not include the arrival of current OBE and OS for use in the generation of solution alternatives process. It is assumed that those files are always available in the case of a modification project;
- the model does not include the arrival of current OBE, current OS, current running times, current track layout and geodata maps and a decision making report that serve as a trigger for the project definition. The model assumes that the triggers are always available at the arrival of a project as they form the cause to start an engineering design process;

- the model does not include a possible reconsideration of the goal, means and requirements process due to unattainable requirements. In reality, this feedback process is strongly iterative. This means that each project makes several cycles between requirements and alternative development before Prorail can select one final alternative. The amount of cycles is however hard to estimate and even harder to model. The process times already reflect that iterative behavior. Therefore, a somewhat worse performance estimation is chosen over the probability for structural modeling errors.

L-A2x) Data preparation:

- the model does not include the arrival of empty workspace for use in the generation of solution alternatives process. It is assumed that those files are always available in the case of a modification project;
- the model does not include the arrival of future OBE, future OS, system requirements and future running times that serve as a trigger for the project definition. The model assumes that the triggers are always available at the arrival of a project as they form the cause to start an engineering design process.

L-A3x) Design interlocking area:

- the model does not include the arrival of the design protocol, system requirements, data gap report, future OBE concept and future OS concept that serve as a trigger for various interlocking area design processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals.

L-A3x) EVP engineering

- the model does not include the arrival of the validated SVA, EVP engineering protocol, EVP conversion protocol, EVP test protocol, validated OBE and validated OS that serve as a trigger of various EVP engineering processes. The model assumes that the triggers are always available at the arrival of a project since they result from preceding processes or concern archived instruction manuals.

L-A4) Test "Dry"

- the model does not include the arrival of the dry test protocol that serves as a trigger for the dry test process. The model assumes that the trigger is always available at the arrival of a project since it concerns an archived instruction manual.

Model Validation

This part investigates whether the model corresponds to real life behavior: validation. Validation aims to achieve a model that enables the evaluation of effects associated to alternatives to the base model. Verbraeck and Valentin (2006) define two ways of validation: duplicative validation and structural validation. Duplicative validation compares process results from the simulation model with the values communicated by experts. Structural validation also investigates whether the effects of different input parameters results in desired model behavior.

Duplicative Validation

Verbraeck and Valentin (2006) explain one type of duplicative validation test that suffices for a validation of empirical versus calculated process results: a quantitative data process comparison. A quantitative data process comparison compares whether the calculated figures by the simulation model differ from the empirical values. In the case of inexplicable deviations, the model needs modifications. A series of iterations with different parameters leads to the final model.

The next investigates the simulated process times versus the empirical times communicated by experts, since they provided process times in most detail. When the simulated time allocation cannot be reasonably explained, parameters will be changed. This process does not lead to an exact match. Mainly three modeling characteristics and one practical reason prevent an exact match. The model contains stochastic process times, random project arrivals with consequent queues and possibilities for rework. The empirical data does not reflect the effects of those characteristics. Furthermore, an interrelated process, queue and loop time structure makes it hard to approach the empirical values. Practically, a trade-off between modeling time and model accuracy also prevents an exact match, if possible at all.

For the purposes of this study, an exactly matching model would not lead to more reliable results. This study focuses on the relative effects of the RailML interlocking engineering design chain compared to the current situation and the possible lean one. Therefore, the model must reflect all process concepts in the right way. Besides, the process times represent estimations as well. Although the model underwent many iterations, the next elaborates on the four main iteration steps. Table 40 compares the empirical values with those from the first model that follows directly from the specification. The table shows that each process takes more than the empirical total. In one case the difference is eleven times the empirical process time. The main reason for the difference arises from the additional queue time and rework time. The value added times actually correspond very well. Therefore, the total of simulated non value added times, waiting time during the process, queue time and rework, should decrease. The factor difference between the empirical non value added time and the simulated non value added time plus the queue time leads to the first set of adjusted non value added times for input in the simulation model (Equation 15). As a consequence, also the standard deviations changed and remained 10% of the process average.

Equation 15: Determination of new NVA times to account for queue and rework time.

$$\begin{aligned}
 &NVA \text{ process time} \\
 &= \left(\frac{\text{Empirical NVA process time}}{\text{Simulated NVA} + \text{Queue time}} \right) \\
 &\quad * \text{Empirical NVA process time}
 \end{aligned}$$

Besides changing the NVA process parameters, the SWOD and Test Protocol development processes got one additional resource due to relatively long queuing times.

Table 40: The empirical values versus the simulation values. The empirical process times decomposed in a value added part (VA) and non value added part (NVA). Furthermore, the table shows the simulation times decomposed in NVA process time, VA process time, queue time and rework time. In addition the total times and their ratio is shown. The model has no rework possibility in the definition of the final design as explained in the specification. Furthermore, Siemens provided only VA times for the engineering process. The values are all noted in days. For reasons of confidentiality, the absolute figures of Siemens are left out.

Low complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	70.0	70.0	14.0	56.0	10.9	0.0	80.9	70.0	70.0	1.2	0.8
FIS1	79.8	79.8	21.0	58.8	8.0	0.0	87.8	79.8	79.8	1.1	0.9
FIS2	53.2	53.2	14.0	39.2	16.8	0.0	70.0	53.2	53.2	1.3	0.7
FIS3	10.5	21.0	3.5	7.0	14.7	29.3	54.4	10.5	21.0	2.6	0.3
FIS4	7.0	14.0	2.5	4.5	1.4	12.6	21.0	7.0	14.0	1.5	0.8
RVTO1	28.0	28.0	23.0	5.0	45.5		73.5	28.0	28.0	2.6	0.1
RVTO2	21.0	42.0	10.5	10.5	45.5	77.4	143.9	21.0	42.0	3.4	0.2
RVTO validation	7.0	14.0	2.5	4.5	0.2	16.0	23.2	7.0	14.0	1.7	1.0
SWOD	28.0	28.0	24.5	3.5	64.8	11.6	104.4	28.0	28.0	3.7	0.1
Data prep										2.9	0.0
TOS										5.8	0.0
EVP										1.8	0.0
EVP conversion										2.0	0.0

Medium complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
EVP test										2.1	0.0
Test protocol	7.0	7.0	3.5	3.5	60.6	8.4	76.0	7.0	7.0	10.9	0.1
Dry test	7.0	7.0	3.5	3.5	17.0	3.0	27.0	7.0	7.0	3.9	0.2
CRS	140.0	140.0	28.1	112.0	10.5	0.0	150.6	140.0	140.0	1.1	0.9
FIS1	159.2	159.2	42.0	117.3	6.7	0.0	166.0	159.2	159.2	1.0	0.9
FIS2	106.4	106.4	28.0	78.5	16.0	0.0	122.4	106.4	106.4	1.2	0.8
FIS3	21.0	42.0	7.0	14.0	13.7	41.1	75.7	21.0	42.0	1.8	0.5
FIS4	14.0	28.0	5.0	9.0	0.9	21.3	36.2	14.0	28.0	1.3	0.9
RVTO1	84.0	84.0	70.0	14.0	45.8		129.8	84.0	84.0	1.5	0.2
RVTO2	63.0	126.0	31.5	31.5	45.8	133.2	242.0	63.0	126.0	1.9	0.4
RVTO validation	14.0	28.0	5.0	9.0	0.2	24.4	38.5	14.0	28.0	1.4	1.0
SWOD	56.0	56.0	49.0	7.0	45.2	13.3	114.5	56.0	56.0	2.0	0.1
Data prep										1.7	0.0
TOS										2.1	0.0
EVP										1.2	0.0
EVP conversion										1.3	0.0
EVP test										1.4	0.0
Test protocol	28.0	28.0	14.0	14.0	55.7	11.0	94.7	28.0	28.0	3.4	0.2
Dry test	14.0	14.0	7.0	7.0	12.8	3.5	30.3	14.0	14.0	2.2	0.4
High complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	560.0	560.0	112.1	448.5	10.0	0.0	570.6	560.0	560.0	1.0	1.0
FIS1	638.4	638.4	168.0	469.1	5.3	0.0	642.4	638.4	638.4	1.0	1.0
FIS2	425.6	425.6	111.7	314.4	12.9	0.0	439.1	425.6	425.6	1.0	1.0
FIS3	84.0	168.0	28.0	56.0	12.7	123.4	220.1	84.0	168.0	1.3	0.8
FIS4	28.0	56.0	10.0	18.0	0.6	38.6	67.2	28.0	56.0	1.2	1.0
RVTO1	672.0	672.0	557.7	111.6	40.2		709.5	672.0	672.0	1.1	0.7
RVTO2	1008.0	2016.0	504.9	504.4	40.2	1355.4	2404.8	1008.0	2016.0	1.2	0.9
RVTO validation	28.0	56.0	10.0	18.0	0.1	37.1	65.2	28.0	56.0	1.2	1.0
SWOD	196.0	196.0	175.0	21.0	57.9	36.3	290.3	196.0	196.0	1.5	0.3
Data prep										1.2	0.0
TOS										1.4	0.0
EVP										1.4	0.0
EVP conversion										1.2	0.0
EVP test										1.3	0.0
Test protocol	112.0	112.0	56.1	56.2	42.4	22.1	176.8	112.0	112.0	1.6	0.6
Dry test	21.0	21.0	10.5	10.5	11.0	4.6	36.6	21.0	21.0	1.7	0.5

Table 41 shows improved results for the first main iteration although some ratios still deviate too much; especially for low and high complexity projects. Therefore, Equation 15 has been applied once more. In addition, the Data Preparation, TOS, EVP and EVP test development got one additional resource to minimize the queue times. It is likely

that the engineering processes have non value added times, although Siemens did not provide that data. Therefore, the queues are kept to a minimum to reflect the provided data in the best way. Each alternative may or may not change the queue times with equal resources; most interesting is the extent to which this happens. As a result, queue times should not be left out of the model.

Table 41: The empirical values versus the simulation values after the first main iteration. The empirical process times decomposed in a value added part (VA) and non value added part (NVA). Furthermore, the table shows the simulation times decomposed in NVA process time, VA process time, queue time and rework time. In addition the total times and their ratio is shown. The model has no rework possibility in the definition of the final design as explained in the specification. Furthermore, Siemens provided only VA times for the engineering process. The values are all noted in days. For reasons of confidentiality, the absolute figures of Siemens are left out.

Low complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	70.0	70.0	14.0	46.9	5.3	0.0	66.2	70.0	70.0	0.9	1.1
FIS1	79.8	79.8	21.0	51.8	4.9	0.0	77.7	79.8	79.8	1.0	1.0
FIS2	53.2	53.2	14.0	27.4	5.5	0.0	46.9	53.2	53.2	0.9	1.2
FIS3	10.5	21.0	3.5	2.3	1.5	12.7	20.1	10.5	21.0	1.0	1.8
FIS4	7.0	14.0	2.5	3.4	1.3	5.5	12.7	7.0	14.0	0.9	1.0
RVTO1	28.0	28.0	23.0	0.5	10.1		33.6	28.0	28.0	1.2	0.5
RVTO2	21.0	42.0	10.5	2.0	10.1	33.3	55.9	21.0	42.0	1.3	0.9
RVTO validation	7.0	14.0	2.5	4.3	0.2	5.4	12.4	7.0	14.0	0.9	1.0
SWOD	28.0	28.0	24.5	0.2	5.2	9.2	39.1	28.0	28.0	1.4	0.6
Data prep										3.5	0.0
TOS										6.9	0.0
EVP										1.3	0.0
EVP conversion										1.9	0.0
EVP test										2.2	0.0
Test protocol	7.0	7.0	3.5	0.2	9.0	6.7	19.4	7.0	7.0	2.8	0.4
Dry test	7.0	7.0	3.5	0.6	3.5	2.4	10.0	7.0	7.0	1.4	0.9
Medium complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	140.0	140.0	28.0	102.2	5.0	0.0	135.3	140.0	140.0	1.0	1.0
FIS1	159.2	159.2	41.9	110.4	4.6	0.0	156.9	159.2	159.2	1.0	1.0
FIS2	106.4	106.4	28.1	65.0	5.7	0.0	98.7	106.4	106.4	0.9	1.1
FIS3	21.0	42.0	7.0	7.1	1.6	25.1	40.8	21.0	42.0	1.0	1.6
FIS4	14.0	28.0	5.0	8.2	1.1	13.0	27.3	14.0	28.0	1.0	1.0
RVTO1	84.0	84.0	70.2	3.3	10.2		83.7	84.0	84.0	1.0	1.0
RVTO2	63.0	126.0	31.5	12.8	10.2	76.6	131.1	63.0	126.0	1.0	1.4
RVTO validation	14.0	28.0	5.0	8.8	0.2	12.6	26.6	14.0	28.0	0.9	1.0
SWOD	56.0	56.0	49.0	0.9	4.6	10.7	65.2	56.0	56.0	1.2	1.3
Data prep										2.0	0.0
TOS										2.4	0.0
EVP										0.5	0.0
EVP conversion										1.4	0.0
EVP test										1.4	0.0
Test protocol	28.0	28.0	14.0	2.8	8.2	8.8	33.9	28.0	28.0	1.2	1.3
Dry test	14.0	14.0	7.0	2.5	3.1	2.8	15.5	14.0	14.0	1.1	1.2
High complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	560.0	560.0	112.2	437.9	4.5	0.0	554.7	560.0	560.0	1.0	1.0
FIS1	638.4	638.4	167.4	465.7	4.7	0.0	637.8	638.4	638.4	1.0	1.0
FIS2	425.6	425.6	112.1	300.2	4.8	0.0	417.1	425.6	425.6	1.0	1.0

FIS3	84.0	168.0	27.9	45.6	1.4	98.9	173.7	84.0	168.0	1.0	1.2
FIS4	28.0	56.0	10.0	17.4	0.8	30.9	59.2	28.0	56.0	1.1	1.0
RVTO1	672.0	672.0	558.2	82.5	10.6		651.2	672.0	672.0	1.0	1.2
RVTO2	1008.0	2016.0	504.1	466.0	10.6	1242.1	2222.7	1008.0	2016.0	1.1	1.1
RVTO validation	28.0	56.0	10.0	17.9	0.1	33.9	61.9	28.0	56.0	1.1	1.0
SWOD	196.0	196.0	175.1	5.6	4.6	31.1	216.4	196.0	196.0	1.1	2.1
Data prep										1.3	0.0
TOS										1.7	0.0
EVP										1.2	0.0
EVP conversion										1.2	0.0
EVP test										1.3	0.0
Test protocol	112.0	112.0	56.1	31.6	7.0	18.9	113.7	112.0	112.0	1.0	1.4
Dry test	21.0	21.0	10.5	5.1	2.8	3.9	22.3	21.0	21.0	1.1	1.3

The third model shown in Table 42 approaches a match but does still contain some inconsistencies in behavior. The average time for a third or fourth version of the FIS or RVTO should not be incorporated in the empirical total. Therefore, the simulated figures should actually be somewhat higher here. The same goes for rework of the SWOD and engineering process when the dry test indicates inconsistencies or errors. For that purpose, the non value

added process times of the FIS definition of system requirements, RVTO and Test Protocol processes have been manually adjusted until a decent model outcome was found. An additional resource for the RVTO processes was necessary to mitigate the effect of a long queue duration for low complexity projects (which actually took longer than the predefined non value added time in total).

Table 42: The empirical values versus the simulation values after the second main iteration. The empirical process times decomposed in a value added part (VA) and non value added part (NVA). Furthermore, the table shows the simulation times decomposed in NVA process time, VA process time, queue time and rework time. In addition the total times and their ratio is shown. The model has no rework possibility in the definition of the final design as explained in the specification. Furthermore, Siemens provided only VA times for the engineering process. The values are all noted in days. For reasons of confidentiality, the absolute figures of Siemens are left out.

Low complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	70.0	70.0	14.0	46.8	6.5	0.0	67.3	70.0	70.0	1.0	1.1
FIS1	79.8	79.8	21.0	51.8	7.3	0.0	80.1	79.8	79.8	1.0	1.0
FIS2	53.2	53.2	14.0	27.4	6.8	0.0	48.2	53.2	53.2	0.9	1.1
FIS3	10.5	21.0	3.5	2.3	2.1	13.3	21.2	10.5	21.0	1.0	1.6
FIS4	7.0	14.0	2.5	3.4	1.4	5.7	13.0	7.0	14.0	0.9	0.9
RVTO1	28.0	28.0	23.0	0.5	6.3		29.8	28.0	28.0	1.1	0.7
RVTO2	21.0	42.0	10.5	2.0	6.3	28.6	47.4	21.0	42.0	1.1	1.3
RVTO validation	7.0	14.0	2.5	4.3	0.2	4.6	11.6	7.0	14.0	0.8	1.0
SWOD	28.0	28.0	24.5	0.2	5.4	5.6	35.7	28.0	28.0	1.3	0.6
Data prep										1.4	0.0
TOS										1.8	0.0
EVP										1.3	0.0
EVP conversion										2.0	0.0
EVP test										1.3	0.0
Test protocol	7.0	7.0	3.5	0.2	0.9	4.0	8.7	7.0	7.0	1.2	3.1
Dry test	7.0	7.0	3.5	0.6	3.8	1.4	9.3	7.0	7.0	1.3	0.8
Medium complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	140.0	140.0	28.0	102.3	6.7	0.0	137.1	140.0	140.0	1.0	1.0
FIS1	159.2	159.2	41.9	110.9	6.2	0.0	159.0	159.2	159.2	1.0	1.0

FIS2	106.4	106.4	28.0	65.3	6.4	0.0	99.8	106.4	106.4	0.9	1.1
FIS3	21.0	42.0	7.0	7.1	2.0	25.4	41.5	21.0	42.0	1.0	1.5
FIS4	14.0	28.0	5.0	8.2	1.1	13.2	27.5	14.0	28.0	1.0	1.0
RVTO1	84.0	84.0	70.0	3.3	6.1		79.4	84.0	84.0	0.9	1.5
RVTO2	63.0	126.0	31.5	12.8	6.1	72.0	122.4	63.0	126.0	1.0	1.7
RVTO validation	14.0	28.0	5.0	8.8	0.2	11.8	25.8	14.0	28.0	0.9	1.0
SWOD	56.0	56.0	49.0	0.9	5.1	9.2	64.2	56.0	56.0	1.1	1.2
Data prep										1.2	0.0
TOS										1.3	0.0
EVP										0.6	0.0
EVP conversion										1.4	0.0
EVP test										1.1	0.0
Test protocol	28.0	28.0	14.0	2.8	0.7	7.6	25.2	28.0	28.0	0.9	4.0
Dry test	14.0	14.0	7.0	2.5	3.5	2.4	15.5	14.0	14.0	1.1	1.2
High complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	560.0	560.0	111.9	438.5	5.9	0.0	556.2	560.0	560.0	1.0	1.0
FIS1	638.4	638.4	168.2	466.9	5.5	0.0	640.5	638.4	638.4	1.0	1.0
FIS2	425.6	425.6	112.0	300.3	5.4	0.0	417.8	425.6	425.6	1.0	1.0
FIS3	84.0	168.0	27.9	45.7	2.0	102.0	177.7	84.0	168.0	1.1	1.2
FIS4	28.0	56.0	10.0	17.4	0.9	31.9	60.2	28.0	56.0	1.1	1.0
RVTO1	672.0	672.0	558.3	82.6	5.9		646.8	672.0	672.0	1.0	1.3
RVTO2	1008.0	2016.0	503.7	465.7	5.9	1221.0	2196.3	1008.0	2016.0	1.1	1.1
RVTO validation	28.0	56.0	10.0	17.9	0.1	33.3	61.3	28.0	56.0	1.1	1.0
SWOD	196.0	196.0	174.8	5.6	5.0	27.9	213.3	196.0	196.0	1.1	2.0
Data prep										1.1	0.0
TOS										1.2	0.0
EVP										1.1	0.0
EVP conversion										1.2	0.0
EVP test										1.2	0.0
Test protocol	112.0	112.0	56.0	31.9	0.6	17.0	105.5	112.0	112.0	0.9	1.7
Dry test	21.0	21.0	10.5	5.1	3.1	3.5	22.2	21.0	21.0	1.1	1.3

Table 43 indicates that the final model's time results almost correspond with the empirical values for the CRS and FIS processes. The FIS process that may require rework, the definition of system requirements, takes longer in the simulation model than empirically defined. This makes sense as the empirical value accounts for one process repetition only while the model includes the probability for a second and third repetition. The same goes for the RVTO process: the model shows a longer duration in the simulation model for each complexity level than the empirical one. The simulated process times of the SWOD until the dry test exceed the empirical values as well, but to an extent in which the rework accounts for that increase. Again, this mostly results from required rework due to

inconsistencies found in the dry test. This does not account for the engineering processes at Siemens (data prep., TOS, EVP, EVP conversion and EVP test). The queuing time takes a big portion of the increase because Siemens did not provide the non value added process times. In that case, the queue times especially form a big portion at low complexity projects because all three kinds of projects compete for the same resource with consequent equal queue times for each complexity level. Thus, the portion of queue time at low complexity is high, but low for high complexity projects. This does not impose any issues as the values seem fair and eventually most interesting remains the relative performance to the RailML and lean model.

Table 43: The empirical values versus the simulation values after the third main iteration. The empirical process times decomposed in a value added part (VA) and non value added part (NVA). Furthermore, the table shows the simulation times decomposed in NVA process time, VA process time, queue time and rework time. In addition the total times and their ratio is shown. The model has no rework possibility in the definition of the final design as explained in the specification. Furthermore, Siemens provided only VA times for the engineering process. The values are all noted in days. For reasons of confidentiality, the absolute figures of Siemens are left out.

Low complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	14.0	56.0	70.0	70.0	14.0	46.8	4.9		65.8	0.9	1.1
FIS1	21.0	58.8	79.8	79.8	21.0	51.8	4.6		77.4	1.0	1.0
FIS2	14.0	39.2	53.2	53.2	14.0	30.0	6.4		50.4	0.9	1.1
FIS3	3.5	7.0	10.5	21.0	3.5	2.5	3.8	18.1	28.0	1.3	1.1
FIS4	2.5	4.5	7.0	14.0	2.5	2.0	6.5	7.8	18.8	1.3	0.5
RVTO1	23.0	5.0	28.0	28.0	23.0	0.1	8.7		31.8	1.1	0.6
RVTO2	10.5	10.5	21.0	42.0	10.5	1.0	8.7	36.6	56.8	1.4	1.1
RVTO validation	2.5	4.5	7.0	14.0	2.5	2.5	7.9	5.9	18.8	1.3	0.4
SWOD	24.5	3.5	28.0	28.0	24.5	0.2	6.7	6.4	37.9	1.4	0.5
Data prep	8.7	0.0	8.7	8.7	8.7	0.0	2.4	1.5	12.7	1.5	0.0
TOS										1.9	0.0
EVP										1.4	0.0
EVP conversion										2.1	0.0
EVP test										1.5	0.0
Test protocol	3.5	3.5	7.0	7.0	3.5	0.5	3.7	4.7	12.5	1.8	0.8
Dry test	3.5	3.5	7.0	7.0	3.5	0.6	4.6	1.7	10.3	1.5	0.7
Medium complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	28.0	112.0	140.0	140.0	28.1	102.4	5.2		135.7	1.0	1.0
FIS1	42.0	117.2	159.2	159.2	41.9	110.6	4.3		156.8	1.0	1.0
FIS2	28.0	78.4	106.4	106.4	28.0	70.0	5.4		103.3	1.0	1.0
FIS3	7.0	14.0	21.0	42.0	7.0	12.0	3.6	33.7	56.3	1.3	0.9
FIS4	5.0	9.0	14.0	28.0	5.0	7.5	5.5	17.5	35.4	1.3	0.7
RVTO1	70.0	14.0	84.0	84.0	70.2	1.0	8.1		79.3	0.9	1.5
RVTO2	31.5	31.5	63.0	126.0	31.5	18.0	8.1	88.3	145.8	1.2	1.2
RVTO validation	5.0	9.0	14.0	28.0	5.0	8.0	7.5	14.5	35.0	1.2	0.6
SWOD	49.0	7.0	56.0	56.0	49.0	0.9	5.5	11.0	66.4	1.2	1.1
Data prep										1.3	0.0
TOS										1.3	0.0
EVP										1.2	0.0
EVP conversion										1.5	0.0
EVP test										1.2	0.0
Test protocol	14.0	14.0	28.0	28.0	14.0	9.0	3.2	9.1	35.3	1.3	1.2
Dry test	7.0	7.0	14.0	14.0	7.0	3.0	4.0	2.9	17.0	1.2	1.0
High complexity	Empirical VA	Empirical NVA	Empirical total	Empirical total corrected	Average of VA model	Average of NVA model	Queue time	Rework time	Total model	Model time / Empirical total time	Empirical NVA / Model NVA plus queue time
CRS	112.0	448.0	560.0	560.0	111.9	436.7	4.5		553.1	1.0	1.0
FIS1	168.0	470.4	638.4	638.4	167.5	465.4	3.7		636.6	1.0	1.0
FIS2	112.0	313.6	425.6	425.6	111.7	301.8	5.6		419.1	1.0	1.0

FIS3	28.0	56.0	84.0	168.0	28.0	55.6	3.1	114.2	200.9	1.2	1.0
FIS4	10.0	18.0	28.0	56.0	10.0	17.4	4.1	35.7	67.2	1.2	0.8
RVTO1	560.0	112.0	672.0	672.0	560.0	74.9	7.9		642.8	1.0	1.4
RVTO2	504.0	504.0	1008.0	2016.0	503.7	490.5	7.9	1337.5	2339.6	1.2	1.0
RVTO validation	10.0	18.0	28.0	56.0	10.0	17.9	6.7	36.5	71.1	1.3	0.7
SWOD	175.0	21.0	196.0	196.0	175.0	10.0	4.1	31.2	220.4	1.1	1.5
Data prep										1.2	0.0
TOS										1.2	0.0
EVP										1.2	0.0
EVP conversion										1.2	0.0
EVP test										1.2	0.0
Test protocol	56.0	56.0	112.0	112.0	55.9	50.0	2.4	19.0	127.3	1.1	1.1
Dry test	10.5	10.5	21.0	21.0	10.5	7.0	3.4	3.9	24.8	1.2	1.0

Structural Validation

The structural validation evaluates the performance of the model after an extreme condition test, sensitivity analyses, modified behavior prediction test and face validation test (Thissen et al., 2008; Verbraeck & Valentin, 2006).

Extreme Condition Test

The extreme condition test investigates whether the model shows the predicted effects of some extreme situations. The analysis includes two extreme tests: an arrival of many projects and the arrival of just one project. One would expect that with many arrivals, the resource utilization rates go to 1 and the queue times increase substantially. With one arrival, one would expect utilization rates to go to 0, no queue times and no additional rework (because of a very low probability).

Table 44 shows that the waiting times in queues go through the roof. In addition, the occupancy rates of resources become or approach 1 or always occupied. A few queue times and utilization rates go down. This behavior makes sense because an upstream bottleneck limits downstream flow. Especially with an increase in high complexity projects that require longer process times.

Table 44: The waiting times in queues when arrival rate goes to 73 low complexity units a year, 43 medium complexity projects a year and 12 high complexity projects a year. Normal and test units are in days.

	Normal	Test	Percent increase
Low complexity project			
CRS	5.9	8865.555	151498.8%
FIS1	6.1	243.9559	4026.1%
FIS2	5.5	30.09375	548.6%
FIS3	4.3	13.00109	302.2%
FIS4	1.3	6.60381	501.0%
RVTO1	14.4	5.682914	39.5%
RVTO2	14.4	5.682914	39.5%
RVTO validation	0.2	1.915664	1000.1%
SWOD	6.8	67.62505	1001.3%
Data prep	2.0	2.738164	139.1%

TOS	4.8	17.37999	364.4%
EVP	7.4	177.8206	2418.6%
EVP conversion	5.8	7.539616	129.5%
EVP test	11.1	1784.171	16106.3%
Test protocol	2.3	2.09147	91.5%
Dry test	4.0	5.331901	132.5%
Medium complexity project			
CRS	5.5	8863.639	160223.0%
FIS1	5.8	238.2413	4105.5%
FIS2	5.6	26.3904	469.3%
FIS3	4.2	11.60181	275.2%
FIS4	1.0	5.021257	494.3%
RVTO1	15.1	5.550962	36.7%
RVTO2	15.1	5.550962	36.7%
RVTO validation	0.2	1.639629	934.4%
SWOD	6.2	67.66839	1093.7%
Data prep	1.7	2.099724	125.7%
TOS	3.7	14.04262	380.8%
EVP	5.8	170.9833	2923.9%
EVP conversion	4.5	4.658206	104.2%
EVP test	9.7	1778.304	18401.8%
Test protocol	2.0	1.669616	84.4%
Dry test	3.7	4.800694	129.6%
High complexity project			
CRS	5.4	8737.574	161459.2%
FIS1	4.3	205.5854	4761.9%
FIS2	4.4	15.5794	356.5%
FIS3	3.7	9.206104	251.1%
FIS4	0.7	1.67185	251.3%
RVTO1	16.5	3.964133	24.1%
RVTO2	16.5	3.964133	24.1%
RVTO validation	0.1	1.275788	1074.2%
SWOD	6.0	61.9965	1026.4%
Data prep	1.4	1.331931	98.5%
TOS	2.8	6.400128	228.4%
EVP	4.7	141.3802	3037.0%
EVP conversion	3.3	3.469426	106.5%
EVP test	6.6	1738.684	26509.6%
Test protocol	1.6	0.93877	60.3%
Dry test	3.0	4.539635	150.8%

Table 45: The utilization rates of process resources when arrival rate goes to 73 low complexity units a year, 43 medium complexity projects a year and 12 high complexity projects a year. Zero means (almost) no occupation; one means always occupied. Normal and test units are in days.

	Normal	Test	Percent difference
Project teams CRS	0.71	1.00	140.46%
Project teams FIS1	0.49	1.00	205.78%
Project teams FIS2	0.53	0.93	174.98%
Project teams FIS3	0.65	0.86	131.12%
Project teams FIS4	0.55	0.81	145.76%
Project teams RVTO	0.74	0.80	108.11%
Project teams RVTO validation	0.72	0.66	92.56%
Project teams SWOD	0.68	0.92	135.69%
Project teams Data prep	0.64	0.68	105.97%
Project teams TOS	0.52	0.85	162.71%
Project teams EVP	0.79	0.97	123.56%
Project teams EVP conversion	0.42	0.74	177.74%
Project teams EVP test	0.65	1.00	154.52%
Project teams test protocol	0.46	0.59	127.18%
Project teams dry test	0.56	0.70	124.53%

Table 46 shows that no queues would exist in the case of one project of each complexity type. As a result, resource utilization rates go down to zero. The amount of replications in sometimes depicts a very small amount. Those little numbers are caused by the fact that the simulation is replicated 30 times. This concedes with the expectations.

Table 46: The waiting times in queues when arrival rate goes to 1 low complexity unit, 1 medium complexity project and 1 high complexity project. Normal and test column units are in days.

	Normal	Test	Percent difference
Low complexity project			
CRS	5.9	0	0.0%
FIS1	6.1	0	0.0%
FIS2	5.5	0	0.0%
FIS3	4.3	0	0.0%
FIS4	1.3	0	0.0%
RVTO1	14.4	0	0.0%
RVTO2	14.4	0	0.0%
RVTO validation	0.2	0	0.0%
SWOD	6.8	0	0.0%
Data prep	2.0	0	0.0%
TOS	4.8	0	0.0%
EVP	7.4	0	0.0%
EVP conversion	5.8	0	0.0%
EVP test	11.1	0	0.0%
Test protocol	2.3	0	0.0%
Dry test	4.0	0	0.0%
Medium complexity project			
CRS	5.5	0	0.0%
FIS1	5.8	0	0.0%
FIS2	5.6	0	0.0%
FIS3	4.2	0	0.0%
FIS4	1.0	0	0.0%
RVTO1	15.1	0	0.0%
RVTO2	15.1	0	0.0%

RVTO validation	0.2	0	0.0%
SWOD	6.2	0	0.0%
Data prep	1.7	0	0.0%
TOS	3.7	0	0.0%
EVP	5.8	0	0.0%
EVP conversion	4.5	0	0.0%
EVP test	9.7	0	0.0%
Test protocol	2.0	0	0.0%
Dry test	3.7	0	0.0%
High complexity project			
CRS	5.4	0	0.0%
FIS1	4.3	0	0.0%
FIS2	4.4	0	0.0%
FIS3	3.7	0	0.0%
FIS4	0.7	0	0.0%
RVTO1	16.5	0	0.0%
RVTO2	16.5	0	0.0%
RVTO validation	0.1	0	0.0%
SWOD	6.0	0	0.0%
Data prep	1.4	0	0.0%
TOS	2.8	0	0.0%
EVP	4.7	0	0.0%
EVP conversion	3.3	0	0.0%
EVP test	6.6	0	0.0%
Test protocol	1.6	0	0.0%
Dry test	3.0	0	0.0%

Table 47: The utilization rates of process resources when arrival rate goes to 1 low complexity unit, 1 medium complexity project and 1 high complexity project. Zero means (almost) no occupation; one means always occupied.

	Normal	Test
Project teams CRS	0.71	0.01
Project teams FIS1	0.49	0.01
Project teams FIS2	0.53	0.01
Project teams FIS3	0.65	0.01
Project teams FIS4	0.55	0.00
Project teams RVTO	0.74	0.01
Project teams RVTO validation	0.72	0.00
Project teams SWOD	0.68	0.00
Project teams Data prep	0.64	0.00
Project teams TOS	0.52	0.00
Project teams EVP	0.79	0.00
Project teams EVP conversion	0.42	0.00
Project teams EVP test	0.65	0.00
Project teams test protocol	0.46	0.00
Project teams dry test	0.56	0.00

Table 48: The amount of process replications when arrival rate goes to 1 low complexity unit, 1 medium complexity project and 1 high complexity project.

	Normal	Test
Count 2nd High Validations	15.39	0.17
Count 2nd Low Validations	127.96	0.26
Count 2nd Medium Validations	50.87	0.26
Count 3th High Validations	0.70	0.00
Count 3th Low Validations	6.09	0.04
Count 3th Medium Validations	2.83	0.00

Count RVTO 2nd High Validations	14.30	0.13
Count RVTO 2nd Low Validations	126.04	0.26
Count RVTO 2nd Medium Validations	53.35	0.35
Count RVTO 3th High Validations	0.87	0.00
Count RVTO 3th Low Validations	6.83	0.00
Count RVTO 3th Medium Validations	2.65	0.04
Redo SWOD EVP high	10.26	0.13
Redo SWOD EVP low	75.61	0.17
Redo SWOD EVP medium	31.52	0.26

the inter arrival rate, sometimes an additional 5% decrease does not lead to a substantially lower queue time. For one bottlenecks in the process which have a much longer processing time on average such as RVTO and SWOD, could dampen the effect of a relatively small decrease in inter arrival rate. Especially for the subsequent process after a bottleneck. Furthermore, some of the waiting times are close to zero which means that they cannot decrease any further. As a consequence, one does not see a difference when decreasing the inter arrival rate. In general, the model scores well on this particular type of sensitivity analysis.

Sensitivity Analyses

The sensitivity analyses evaluates the effect of different inter arrival rates and different standard deviations of processes. When the inter arrival rate increases, one would expect that queues increase as well and v.v. Table 49 proves this expectation for the developed model. The rate in which waiting times increase or decrease per 5% of inter arrival rate however differs. Especially when decreasing

The sensitivity test of the standard deviations tests whether an increase would result in a larger confidence interval and v.v.. Table 50 shows that the model reflects that statement. In fact, the half width appears to change in a linear way: for every 5% of the average process time added to the standard deviation, the half width increases with about 15%. Therefore, the model replies very well to this test.

Table 49: The percentage time of the base model's queue time per process in the engineering design chain when changing the inter arrival rate.

	CRS	FIS1	FIS2	FIS3	FIS4	RVTO1	RVTO2	RVTO val.	SWOD	Data prep	TOS	EVP	EVP conv.	EVP test	Test prtcl	Dry test
Low complexity projects																
Minus 15% entities	24%	10%	21%	41%	50%	5%	5%	57%	31%	56%	56%	38%	61%	23%	58%	77%
Minus 10% entities	44%	33%	43%	59%	72%	6%	6%	82%	46%	65%	68%	54%	80%	39%	65%	87%
Minus 5% entities	53%	64%	71%	76%	78%	26%	26%	88%	56%	69%	68%	59%	72%	47%	67%	89%
Plus 5% entities	136%	152%	134%	123%	110%	107%	107%	126%	152%	114%	119%	137%	100%	169%	116%	106%
Plus 10% entities	200%	213%	196%	164%	138%	287%	287%	171%	210%	136%	150%	188%	119%	297%	131%	116%
Plus 15% entities	256%	259%	228%	200%	151%	416%	416%	190%	313%	147%	170%	222%	130%	353%	145%	125%
Medium complexity projects																
Minus 15% entities	27%	10%	20%	41%	53%	5%	5%	68%	37%	59%	60%	36%	65%	24%	62%	76%
Minus 10% entities	49%	34%	41%	55%	70%	4%	4%	88%	46%	68%	70%	51%	82%	39%	65%	83%
Minus 5% entities	56%	60%	65%	72%	81%	25%	25%	91%	61%	65%	70%	59%	77%	45%	66%	83%
Plus 5% entities	157%	140%	121%	119%	102%	107%	107%	126%	164%	116%	127%	157%	101%	188%	114%	103%
Plus 10% entities	203%	208%	177%	161%	131%	275%	275%	162%	219%	118%	149%	201%	119%	308%	119%	111%
Plus 15% entities	252%	250%	194%	190%	141%	394%	394%	180%	333%	144%	178%	240%	126%	373%	141%	120%
High complexity projects																
Minus 15% entities	23%	11%	25%	52%	58%	3%	3%	51%	39%	52%	62%	35%	75%	18%	56%	80%
Minus 10% entities	39%	41%	40%	60%	74%	3%	3%	82%	50%	74%	66%	52%	90%	33%	69%	91%
Minus 5% entities	48%	66%	76%	86%	81%	23%	23%	100%	66%	83%	76%	58%	85%	42%	81%	87%
Plus 5% entities	148%	161%	125%	129%	95%	89%	89%	127%	145%	115%	107%	145%	102%	210%	117%	111%
Plus 10% entities	179%	238%	174%	160%	113%	225%	225%	165%	208%	121%	128%	180%	118%	341%	117%	113%
Plus 15% entities	263%	288%	196%	198%	133%	336%	336%	201%	284%	134%	137%	199%	114%	401%	130%	121%

Table 50: The percentage of half width compared to the base model (10% of average process time to account as SD).

Half width	Data															
	CRS	FIS1	FIS2	FIS3	FIS4	RVTO1	RVTO2	RVTO val.	SWOD	prep	TOS	EVP	EVP conv.	EVP test	Test prtcl	Dry test
Low complexity projects																
SD 5%μ	84%	85%	88%	90%	86%	86%	85%	86%	88%	86%	81%	85%	85%	85%	87%	85%
SD 20%μ	121%	122%	127%	130%	133%	128%	128%	126%	130%	131%	118%	126%	124%	127%	140%	128%
SD 30%μ	147%	171%	156%	164%	165%	159%	157%	159%	155%	175%	152%	158%	149%	150%	154%	161%
Medium complexity projects																
SD 5%μ	89%	86%	84%	88%	87%	87%	87%	84%	87%	88%	84%	85%	84%	86%	87%	80%
SD 20%μ	135%	123%	126%	135%	135%	133%	133%	124%	128%	129%	130%	129%	128%	126%	129%	115%
SD 30%μ	158%	153%	152%	144%	176%	169%	150%	150%	156%	155%	146%	143%	144%	143%	161%	152%
High complexity projects																
SD 5%μ	88%	85%	82%	86%	86%	90%	94%	87%	89%	88%	84%	93%	84%	87%	85%	85%
SD 20%μ	119%	119%	123%	120%	130%	130%	140%	117%	130%	123%	120%	121%	117%	126%	118%	122%
SD 30%μ	144%	142%	148%	150%	157%	156%	167%	148%	155%	144%	148%	151%	157%	146%	155%	153%

Modified Behavior Prediction

This test investigates two types of behavior in particular: the absence of additional process replications and an abundance of process replications.

In the case of absent process replications, one would expect the replication time to be minimal for the FIS and RVTO processes and zero for the SWOD/engineering processes. After all, the FIS and RVTO always require a replication and the SWOD/engineering does not. Table 51 shows that the SWOD/engineering replication time goes to zero as expected. Table 51 learns that the replication times that reflect the RVTO and FIS are very close to the minimal process times.

In the case of an abundant amount of replications, i.e. double the probability, one would expect the base model's replication time to increase by half the (near) minimal time. With exception of the FIS replication for high complexity projects, this statement is always correct.

Table 51: The average amount of days consumed for replication of processes per type of project complexity.

	No additional replication	Normal	Double amount of replications
	Time FIS replication high	114.03	145.48
Time FIS replication low	16.38	23.41	33.79
Time FIS replication medium	32.27	43.15	57.66
Time RVTO replication high	1033.48	1298.1	1621.44
Time RVTO replication low	25.72	42.10	58.24
Time RVTO replication medium	73.34	104.46	132.75
Time SWOD / engineering replication high	0	127.83	254.31
Time SWOD / engineering replication low	0	24.76	71.32
Time SWOD / engineering replication medium	0	69.24	159.85

Face Validity

The modeler and author of this study ran through the model several times to verify and validate the model. The validation and verification chapters elaborate on the big adjustments made, but the modeler also adjusted a lot of smaller aspects like type errors, wrong references, little process time mistakes, sequence of process events in the model and so on. After this series of adjustments, the modeler declared this model sufficiently validated for the purposes of the study.

Furthermore, experts gave their comments on the model as well. Separate sessions with mister Dragt, mister Janssen and mister Beelaerts van Blokland revealed improvement areas that have been incorporated in the model.

The Composition of the Final ARENA Model

The Status Quo / Base Model

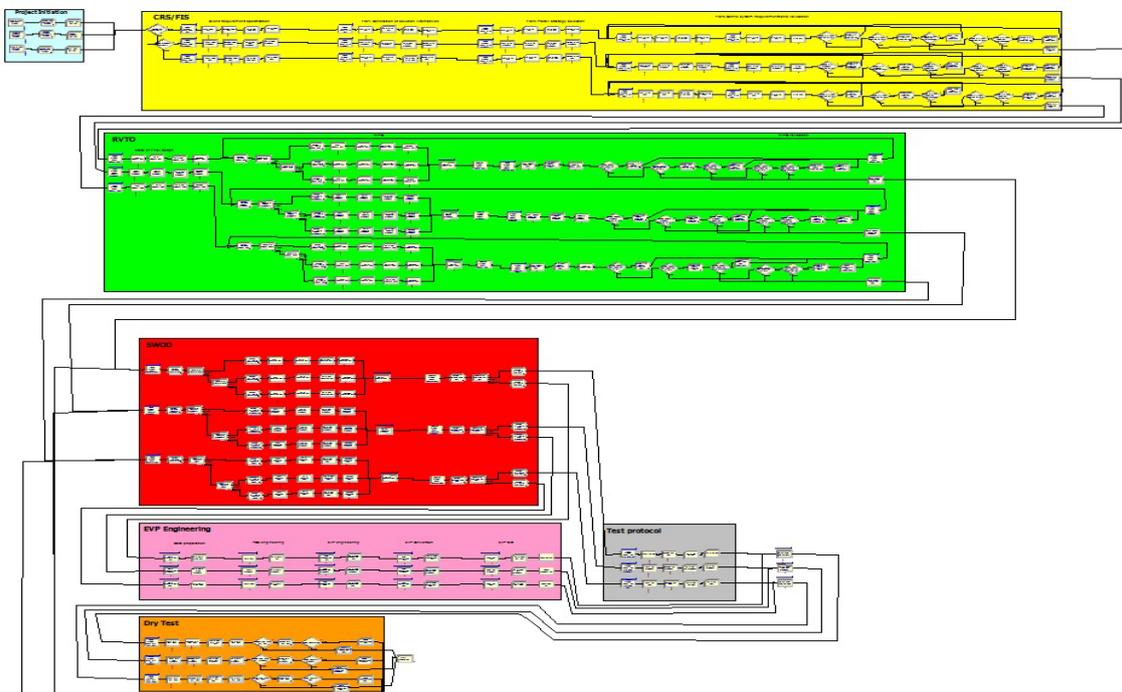


Table 52: Input process parameters of the base model. The figures' unit is in days.

Name	StDev	Mean	Name	StDev	Mean	Name	StDev	Mean
Develop CRS low	1.4	14	Wait RVTO2 low	.1	1	Develop SWOD1 high	17.5	175
Develop CRS medium	2.8	28	Develop RVTO3 low	1.05	10.5	Wait SWOD1 high	1	10
Develop CRS high	11.2	112	Wait RVTO3 low	.1	1	Develop SWOD2 high	17.5	175
Wait CRS low	4.69	46.9	Develop RVTO4 low	1.05	10.5	Wait SWOD2 high	1	10
Wait CRS medium	10.24	102.4	Wait RVTO4 low	.1	1	Develop SWOD3 high	17.5	175
Wait CRS high	43.78	437.8	Develop RVTO2 medium	3.15	31.5	Wait SWOD3 high	1	10
Design FIS1 low	2.1	21	Wait RVTO2 medium	1.8	18	Wait FIS4 low	.2	2
Design FIS1 medium	4.2	42	Develop RVTO3 medium	3.15	31.5	FIS4 low	0.25	2.5
Design FIS1 high	16.8	168	Wait RVTO3 medium	1.8	18	Data preparation low	0.87	8.7
Wait FIS1 low	5.18	51.8	Develop RVTO4 medium	3.15	31.5	Data preparation medium	2.13	21.3
Wait FIS1 medium	11.08	110.8	Wait RVTO4 medium	1.8	18	Data preparation high	7.85	78.5
Wait FIS1 high	46.64	466.4	Develop RVTO2 high	50.4	504	TOS low	0.88	8.8
Design FIS2 low	1.4	14	Wait RVTO2 high	49	490	TOS medium	3.55	35.5
Design FIS2 medium	2.8	28	Develop RVTO3 high	50.4	504	TOS high	6.32	63.2
Design FIS2 high	11.2	112	Wait RVTO3 high	49	490	EVP engineering low	2.06	20.6
Wait FIS2 low	3	30	Develop RVTO4 high	50.4	504	EVP engineering medium	8.29	82.9
Wait FIS2 medium	7	70	Wait RVTO4 high	49	490	EVP engineering high	14.76	147.6
Wait FIS2 high	30.05	300.5	Wait RVTO Validation Low	.25	2.5	EVP conversion low	0.65	6.5
Design FIS3 low	0.35	3.5	Validate RVTO Low	0.25	2.5	EVP conversion medi-	1.5	15

						um		
Design FIS3 medium	0.7	7	Wait RVTO Validation Medium	.8	8	EVP conversion high	6.44	64.4
Design FIS3 high	2.8	28	Validate RVTO Medium	0.5	5	EVP test low	5.47	54.7
Wait FIS3 low	.25	2.5	Wait RVTO Validation High	1.79	17.9	EVP test medium	21.03	210.3
Wait FIS3 medium	1.2	12	Validate RVTO High	1	10	EVP test high	19.38	193.8
Wait FIS3 high	5.56	55.6	Develop SWOD1 low	2.45	24.5	Develop TP low	0.35	3.5
Wait FIS4 medium	.75	7.5	Wait SWOD1 low	0.02	0.2	Develop TP medium	1.4	14
FIS4 medium	0.5	5	Develop SWOD2 low	2.45	24.5	Develop TP high	5.6	56
Wait FIS4 high	1.74	17.4	Wait SWOD2 low	0.02	0.2	Wait TP low	.5	0.5
FIS4 high	1	10	Develop SWOD3 low	2.45	24.5	Wait TP medium	.9	9
Develop RVTO1 low	2.3	23	Wait SWOD3 low	0.02	0.2	Wait TP high	5	50
Develop RVTO1 medium	7	70	Develop SWOD1 medium	4.9	49	Develop dry test low	0.35	3.5
Develop RVTO1 high	56	560	Wait SWOD1 medium	0.09	0.9	Develop dry test medium	0.7	7
Wait RVTO1 low	0.01	0.1	Develop SWOD2 medium	4.9	49	Develop dry test high	1.05	10.5
Wait RVTO1 medium	.1	1	Wait SWOD2 medium	0.09	0.9	Wait dry test low	0.06	0.6
Wait RVTO1 high	7.5	75	Develop SWOD3 medium	4.9	49	Wait dry test medium	.3	3
Develop RVTO2 low	1.05	10.5	Wait SWOD3 medium	0.09	0.9	Wait dry test high	.7	7

Table 53: Input resource parameters of the base model.

Name	# of resources	Name	# of resources	Name	# of resources
Project teams CRS	12	Project teams RVTO	30	Project teams EVP	6
Project teams FIS1	14	Project teams RVTO validation	2	Project teams EVP conversion	2
Project teams FIS2	9	Project teams SWOD	6	Project teams EVP test	11
Project teams FIS3	4	Project teams Data prep	3	Project teams test protocol	3
Project teams FIS4	2	Project teams TOS	3	Project teams dry test	1

The RailML Model

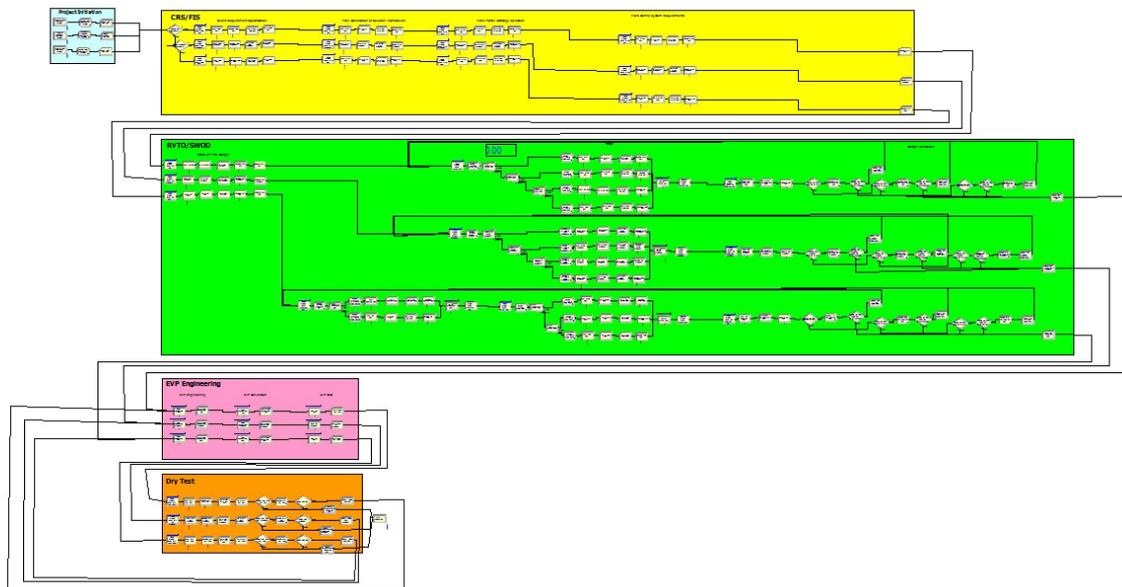


Table 54: Input process parameters of the RailML worst case model. The figures' unit is in days.

Name	StDev	Value	Name	StDev	Value	Name	StDev	Value
Develop CRS low	1.4	14	Develop FD high	56	560	Develop OR high	17.5	175
Develop CRS medium	2.8	28	Wait FD low	.01	.1	Wait OR high	1	10
Develop CRS high	11.2	112	Wait FD medium	.1	1	Develop OBE high	17.5	175
Wait CRS low	4.69	46.9	Wait FD high	7.5	75	Wait OBE high	1	10
Wait CRS medium	10.24	102.4	Develop gis low	3.5	35	Develop SVA high	17.5	175
Wait CRS high	43.78	437.8	Wait gis low	.12	1.2	Wait SVA high	1	10
Design FIS1 low	2.1	21	Develop gis medium	8.05	80.5	EVP engineering low	2.06	20.6
Design FIS1 medium	4.2	42	Wait gis medium	1.89	18.9	EVP engineering medium	8.29	82.9
Design FIS1 high	16.8	168	Wait DV Validation Low	.25	2.5	EVP engineering high	14.76	147.6
Wait FIS1 low	5.18	51.8	Validate DV Low	0.25	2.5	EVP conversion low	0.65	6.5
Wait FIS1 medium	11.08	110.8	Wait DV Validation Medium	.8	8	EVP conversion medium	1.5	15
Wait FIS1 high	46.64	466.4	Validate DV Medium	0.5	5	EVP conversion high	6.44	64.4
Design FIS2 low	1.4	14	Wait DV Validation High	1.79	17.9	EVP test low	5.47	54.7
Design FIS2 medium	2.8	28	Validate DV High	1	10	EVP test medium	21.03	210.3
Design FIS2 high	11.2	112	Develop OR low	2.45	24.5	EVP test high	19.38	193.8
Wait FIS2 low	3	30	Wait OR low	.02	.2	Develop dry test low	0.35	3.5
Wait FIS2 medium	7	70	Develop OBE low	2.45	24.5	Develop dry test medium	0.7	7
Wait FIS2 high	30.05	300.5	Wait OBE low	.02	.2	Develop dry test high	1.05	10.5
Design FIS3 low	0.35	3.5	Develop SVA low	2.45	24.5	Wait dry test low	0.06	0.6
Design FIS3 medium	0.7	7	Wait SVA low	.02	.2	Wait dry test medium	.3	3
Design FIS3 high	2.8	28	Develop OR medium	4.9	49	Wait dry test high	.7	7
Wait FIS3 low	.25	2.5	Wait OR medium	0.09	0.9	Develop gis high	50.4	504

Wait FIS3 medium	1.2	12	Develop OBE medium	4.9	49	Wait gis high	49	490
Wait FIS3 high	5.56	55.6	Wait OBE medium	0.09	0.9	Develop VTOBE high	50.4	504
Develop FD low	2.3	23	Develop SVA medium	4.9	49	Wait VTOBE high	49	490
Develop FD medium	7	70	Wait SVA medium	0.09	0.9			

Table 55: Input resource parameters of the RailML worst case model.

Name	Capacity	Name	Capacity	Name	Capacity
Project teams CRS	12	Project teams Design validation	2	Project teams Maps	34
Project teams FIS1	14	Project teams EVP	6	Project teams DV validation	2
Project teams FIS2	9	Project teams EVP conversion	2	Project teams Premaps	15
Project teams FIS3	4	Project teams EVP test	11		
Project teams FD	8	Project teams dry test	1		

Table 56: Input process parameters of the RailML best case model. The figures' unit is in days.

Name	StDev	Value	Name	StDev	Value	Name	StDev	Value
Develop CRS low	1.4	14	Develop FD high	56	560	Develop OR high	13.125	131.25
Develop CRS medium	2.8	28	Wait FD low	.01	.1	Wait OR high	.667	6.67
Develop CRS high	11.2	112	Wait FD medium	.1	1	Develop OBE high	13.125	131.25
Wait CRS low	4.69	46.9	Wait FD high	7.5	75	Wait OBE high	.667	6.67
Wait CRS medium	10.24	102.4	Develop gis low	2.048	20.48	Develop SVA high	14.569	145.69
Wait CRS high	43.78	437.8	Wait gis low	.0167	.167	Wait SVA high	.78	7.8
Design FIS1 low	2.1	21	Develop gis medium	4.08	40.8	EVP engineering low	2.06	20.6
Design FIS1 medium	4.2	42	Wait gis medium	.078	.78	EVP engineering medium	8.29	82.9
Design FIS1 high	16.8	168	Wait DV Validation Low	.25	2.5	EVP engineering high	14.76	147.6
Wait FIS1 low	5.18	51.8	Validate DV Low	0.25	2.5	EVP conversion low	0.65	6.5
Wait FIS1 medium	11.08	110.8	Wait DV Validation Medium	.8	8	EVP conversion medium	1.5	15
Wait FIS1 high	46.64	466.4	Validate DV Medium	0.5	5	EVP conversion high	6.44	64.4
Design FIS2 low	1.4	14	Wait DV Validation High	1.79	17.9	EVP test low	5.47	54.7
Design FIS2 medium	2.8	28	Validate DV High	1	10	EVP test medium	21.03	210.3
Design FIS2 high	11.2	112	Develop OR low	1.838	18.38	EVP test high	19.38	193.8
Wait FIS2 low	3	30	Wait OR low	.015	.15	Develop dry test low	0.35	3.5
Wait FIS2 medium	7	70	Develop OBE low	1.838	18.38	Develop dry test medium	0.7	7
Wait FIS2 high	30.05	300.5	Wait OBE low	.015	.15	Develop dry test high	1.05	10.5
Design FIS3 low	0.35	3.5	Develop SVA low	2.048	20.48	Wait dry test low	0.06	0.6
Design FIS3 medium	0.7	7	Wait SVA low	.0167	.167	Wait dry test medium	.3	3
Design FIS3 high	2.8	28	Develop OR medium	3.676	36.76	Wait dry test high	.7	7
Wait FIS3 low	.25	2.5	Wait OR medium	.068	0.68	Develop gis high	37.8	378
Wait FIS3 medium	1.2	12	Develop OBE medium	3.676	36.76	Wait gis high	36.75	367.5
Wait FIS3 high	5.56	55.6	Wait OBE medium	.068	.68	Develop VTOBE high	37.8	378
Develop FD low	2.3	23	Develop SVA medium	4.08	40.8	Wait VTOBE high	36.75	367.5
Develop FD medium	7	70	Wait SVA medium	.078	.78			

Table 57: Input resource paramaters of the RailML best case model.

Name	Capacity	Name	Capacity	Name	Capacity
Project teams CRS	12	Project teams Design validation	2	Project teams Maps	21
Project teams FIS1	14	Project teams EVP	6	Project teams DV validation	2
Project teams FIS2	9	Project teams EVP conversion	2	Project teams Premaps	15
Project teams FIS3	4	Project teams EVP test	11		
Project teams FD	8	Project teams dry test	1		

The Lean Model

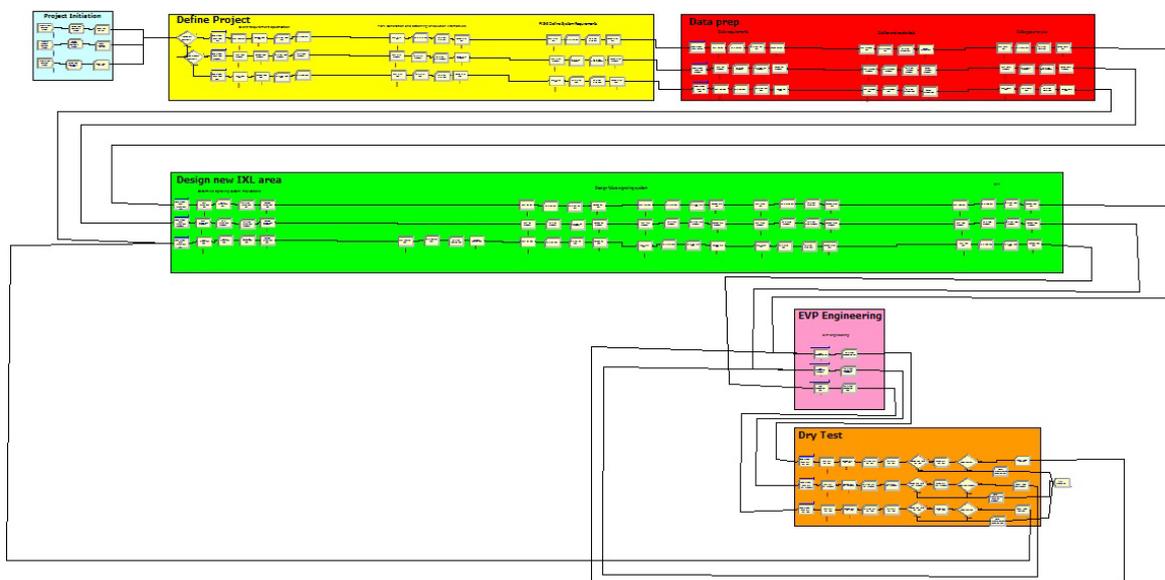


Table 58: Input process paramaters of the RailML lean model. The figures' unit is in days.

Name	StDev	Value	Name	StDev	Value	Name	StDev	Value
Develop CRS low	1.4	14	Develop gis medium	1.575	15.75	Wait dry test low	0.06	0.6
Develop CRS medium	2.8	28	Wait gis medium	.9	9	Wait dry test medium	.3	3
Develop CRS high	11.2	112	Develop OR low	1.225	12.25	Wait dry test high	.7	7
Wait CRS low	4.69	46.9	Wait OR low	.01	.1	Develop gis high	25.2	252
Wait CRS medium	10.24	102.4	Develop OBE low	1.225	12.25	Wait gis high	24.5	245
Wait CRS high	43.78	437.8	Wait OBE low	.01	.1	Develop VTOBE high	25.2	252
Design FIS1 low	2.1	21	Develop SVA low	1.225	12.25	Wait VTOBE high	24.5	245
Design FIS1 medium	4.2	42	Wait SVA low	.01	.1	Design DR low	.117	1.17
Design FIS1 high	16.8	168	Develop OR medium	2.45	24.5	Design DR medium	.233	2.33
Wait FIS1 low	5.18	51.8	Wait OR medium	.045	.45	Design DR high	.933	9.33
Wait FIS1 medium	11.08	110.8	Develop OBE medium	24.5	24.5	Wait DR low	.083	.83
Wait FIS1 high	46.64	466.4	Wait OBE medium	.045	.45	Wait DR medium	.4	4
Design FIS3 low	.117	1.17	Develop SVA medium	24.5	24.5	Wait DR high	1.853	18.53

Design FIS3 medium	.233	2.33	Wait SVA medium	.045	.45	Design GandR low	.117	1.17
Design FIS3 high	.933	9.33	Develop OR high	8.75	87.5	Design GandR medium	.233	2.33
Wait FIS3 low	.083	.83	Wait OR high	.5	5	Design GandR high	.933	9.33
Wait FIS3 medium	.4	4	Develop OBE high	8.75	87.5	Wait GandR low	.083	0.83
Wait FIS3 high	1.853	18.53	Wait OBE high	.5	5	Wait GandR medium	.4	4
Develop Implications low	1.15	11.5	Develop SVA high	8.75	87.5	Wait GandR high	1.853	18.53
Develop Implications medium	3.5	35	Wait SVA high	.5	5	Design DGA low	1.15	11.5
Develop Implications high	28	280	EVP engineering low	2.06	20.6	Design DGA medium	3.5	35
Wait Implications low	.01	0.05	EVP engineering medium	8.29	82.9	Design DGA high	28	280
Wait Implications medium	.1	.5	EVP engineering high	14.76	147.6	Wait DGA low	0.005	0.05
Wait Implications high	7.5	37.5	Develop dry test low	0.35	3.5	Wait DGA medium	0.05	0.5
Develop gis low	0.525	5.25	Develop dry test medium	0.7	7	Wait DGA high	3.75	37.5
Wait gis low	.05	0.5	Develop dry test high	1.05	10.5			

Table 59: Input resource parameters of the lean model.

Name	Capacity	Name	Capacity
Project teams Project definition low	8	Project teams Project definition medium	8
Project teams EVP	5	Project teams Project definition high	10
Project teams dry test	1	Project teams IXL area low	3
Project teams Data prep low	1	Project teams IXL area medium	4
Project teams Data prep medium	2	Project teams IXL area high	13
Project teams Data prep high	4		

Literature

- Agyapong-Kodua, K., Ajaefobi, J. O., & Weston, R. H. (2009). Modelling dynamic value streams in support of process design and evaluation. *International Journal of Computer Integrated Manufacturing*, 22(5), 411-427. doi: 10.1080/09511920802527574
- Beelaerts van Blokland, W., van der Meer, O., & Rakers, R. (2009). Co-innovation and the value time curve: A case study on the Dassault Falcon 7X and Embraer 170/190 series (A. T. Operations, Trans.). Delft: TU Delft.
- Beelaerts van Blokland, W. W. A. (2013, March 14). [Meeting: introduction to Lean].
- Beelaerts van Blokland, W. W. A., Fiksinski, M. A., Amoa, S. O. B., Santema, S. C., van Silfhout, G. J., & Maaskant, L. (2012). Measuring value-leverage in aerospace supply chains. *Int. J. Oper. Prod. Manage. International Journal of Operations and Production Management*, 32(8), 982-1007.
- Beelaerts van Blokland, W. W. A., Santema, S. C., & Curran, R. (2010). Lean Supply Chain Management in Aerospace. Delft: Delft University of Technology.
- Behrouzi, F., Kuan Yew, W., & Behrouzi, F. (2011, 6-9 Dec. 2011). *A study on lean supply chain performance measures of SMEs in the automotive industry*. Paper presented at the Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on.
- Blaauboer, M. (2012). *SIL 4 interlocking based on COTS hardware*. Paper presented at the The Institution of Railway Engineers Australasia Technical Meeting, Brisbane.
- Boxmeer, M. (2012). *Offerte interlocking Delft tunnel*. offerte tunnel op traject Delft zuid-Rijswijk.
- Bramet, W. (2013, February 10). Station Santpoort-Noord. Retrieved June, 2013, from <http://www.stationsweb.nl/station.asp?station=santpoortnoord>
- Bundes, U. d. E.-U. d. (2011). Zugkollision mit sich anschließender Entgleisung Muenchen Lochhausen - Olching 24.07.2007. Bonn: Bundesministerium für Verkehr, Bau und Stadtentwicklung.
- Bundes, U. d. E.-U. d. (2012). Untersuchungsbericht Kollision, 25.11.2008, Recklingshausen Ost. Bonn: Bundesministerium für Verkehr, Bau und Stadtentwicklung.
- Buseyne, E., Wego, A., Vaux, R., & Secci, C. (2011). Seventh Framework Programme theme 7 Transport: INTeGrated European Signalling System. In P. Cicco & G. Barbu (Eds.), *Annex 1 - "Description of Work"* (Vol. Annex 1). Brussels: INESS Consortium.
- Cuatrecasas-Arbos, L., Fortuny-Santos, J., & Vintro-Sanchez, C. (2011). The Operations-Time Chart: A graphical tool to evaluate the performance of production systems - From batch-and-queue to lean manufacturing. *Computers and Industrial Engineering*, 61(3), 663-675. doi: 10.1016/j.cie.2011.04.022
- de Haan, A. (2009). *Inleiding technische bestuurskunde : een raamwerk voor het systematisch oplossen van complexe multi-actorproblemen*. Den Haag: LEMMA.
- de Jong, S. J., Beelaerts van Blokland, W. W. A., Curran, R., & de Jong, Y. E. (2013). Measurement & Optimisation of Value Flow Processes in MRO. Delft: TU Delft.
- Della Bruna, E., Ensslin, L., & Ensslin, S. R. (2011, 27-30 June 2011). *Supply chain performance evaluation: A case study in a company of equipment for refrigeration*. Paper presented at the Technology Management Conference (ITMC), 2011 IEEE International.
- Dotoli, M., Fantì, M. P., Iacobellis, G., & Rotunno, G. (2012). *A lean manufacturing strategy using Value Stream Mapping, the Unified Modeling Language, and discrete event simulation*, Seoul.
- Dragt, S. (2013). [Data source GIS].
- Duinkerken, M. (2011). ME1410 Quantitative Methods for Logistics lecture 2: Queuing Theory and Application.
- ESRI. (2013). What is GIS? Retrieved July, 2013, from www.esri.com/what-is-gis/overview
- European Commission. (2013). The fourth railway package – completing the single european railway area to foster european competitiveness and growth (M. a. Transport, Trans.) (final ed., Vol. COM(2013)). Brussels: Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions.
- Fries, N. (2003). Modellierung einer Eisenbahn-Infrastruktur in RailML. Dresden: TU Dresden.
- Goverde, R. M. P. (2012a). CT4811 Design and Control of Public Transport Systems: Railway Signalling. Retrieved from BlackBoard TU Delft website:
- Goverde, R. M. P. (2012b). Lecture Railway Traffic Management: ERTMS.
- Goverde, R. M. P. (2012c). Lecture Railway Traffic Management: Interlocking.
- Goverde, R. M. P. (2012d). Lecture Railway Traffic Management: Introduction and basics.
- Gregg, M. L., Van Andel, S. M., & Saylor, S. E. (2011). *Lean+ manufacturing process analysis simulation (LPAS+)*, Phoenix, AZ.
- Hallam, C. R. A. (2010, 18-22 July 2010). *Lean supply chain management techniques for complex aerospace systems: Using discrete event simulation to mitigate programmatic cost and schedule risk*. Paper presented at the Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET '10:.

- Hengartner, M. (2003). *Graphikeditor für RailML-basierte Eisenbahninfrastrukturdaten*. (Master), ETH Zürich, Zürich.
- Hillier, F. S., & Liebermann, G. J. (2001). *Queueing Theory Introduction to Operations Research* (7 ed., pp. 834-891). New York: McGraw-Hill International Editions
- Hindle, T. (2008). *Guide to management ideas and gurus*. London: Profile.
- Hoogendoorn, S., Bliemer, M., & van Nes, R. (2007, Oktober 18). Modellen voor netwerkmanagement - een overzicht. *NM Magazine*, 2007.
- Hürlimann, D., & Krauss, V. P. (2003, August 20). *RailML - Einheitliche Datenschnittstellen für Eisenbahnen*. Paper presented at the 19. VWT, Dresden.
- Inspectie Leefomgeving en Transport. (2012). *Frontale botsing tussen twee reizigerstreinen bij Amsterdam Westerpark*. The Hague: Ministerie van Infrastructuur en Milieu.
- InStall. (2011). *Seventh framework Programme Sustainable Surface Transport: Competitiveness through innovation*. Utrecht.
- IVU Traffic Technologies AG. (2010). *IVU ist Gastgeber der 18. railML-Konferenz in Berlin*. In I. T. T. AG (Ed.). Berlin: IVU Traffic Technologies AG.
- Janssen, B. (2012). *L' encrier presentation: Siemens Lean Engineering of rail interlocking*.
- Janssen, B. (2013a). *CIE5803-09 Railway Traffic Management - Lecture ERTMS signaling: The industrial view*.
- Janssen, B. (2013b). [ETCS irt relay interlockings].
- Janssen, B., & Bosschaart, M. (2013). *RailML Modelling: Translating Dutch Railways into RailML*. The Hague: Siemens.
- Janssen, B., & Quaglietta, E. (2013). [Signal aspect dependencies in RailML IXL].
- Jeong, K.-Y., & Phillips, D. T. (2011). Application of a concept development process to evaluate process layout designs using value stream mapping and simulation. *Journal of Industrial Engineering and Management*, 4(2). doi: 10.3926/jiem.2011.v4n2.p206-230
- Kanoun. (2003). *Projektierungsschritt TOS-plan erstellen* (R. Automation, Trans.). In Wacha (Ed.), *Projektierungsrichtlinie* (3 ed.). Barunshweig: Siemens.
- Kanoun. (2011). *Projektierungsschritt Elementverbindungsplan erstellen* (R. Automation, Trans.). In Wacha (Ed.), *Projektierungsrichtlinie* (12 ed.). Barunshweig: Siemens.
- Koelewijn, H., de Rijk, W., & Dragt, S. (2013). [Interview: Prorail Rail Technology Projects].
- Kolic, D., Fafandjel, N., & Zamarin, A. (2012). *Lean Manufacturing Methodology for Shipyards*. *Brodo Gradnja*, 63(1), 18-29.
- Kuijper, N. J. A., & Hendriks, R. (2011). *Themaonderzoek gladheid en detectieproblemen: onderzoek naar de oorzaken en achtergronden*. Utrecht: Ministerie van Infrastructuur en Milieu.
- Lehmann, M., & Albrecht, T. (2008). *Erarbeitung eines XML-basierten Schemas zur Darstellung der sicherungstechnischen Streckenausüstung in einem Fahrsimulator*. In J. Krimmling (Ed.). Dresden: TU Dresden.
- Linder, C., & Grimm, M. (2012). *Datenmodellanalyse zum Austausch von Projektierungsdaten für Stellwerkssysteme in INESS*. *Signal und Draht*, 104(9), 16-21.
- Lu, J. C., Yang, T., & Wang, C. Y. (2011). A lean pull system design analysed by value stream mapping and multiple criteria decision-making method under demand uncertainty. *International Journal of Computer Integrated Manufacturing*, 24(3), 211-228. doi: 10.1080/0951192X.2010.551283
- Ludema, M. (2012). *Lecture SPM4620 Supply Chain Engineering - Lean Thinking, Value Stream and Waste Management*. 13.
- Mahfouz, A., Shea, J., & Arisha, A. (2011, 11-14 Dec. 2011). *Simulation based optimisation model for the lean assessment in SME: A case study*. Paper presented at the Simulation Conference (WSC), Proceedings of the 2011 Winter.
- Mansveld, W. J. (2013). *Toezegging onderbouwing ERTMS reservering € 2 miljard*. The Hague: Dutch ministry of Infrastructure and Environment.
- Marzouk, M., Bakry, I., & El-Said, M. (2012). *Assessing design process in engineering consultancy firms using lean principles*. *Simulation*, 88(12), 1522-1536. doi: 10.1177/0037549712459772
- Mason-Jones, R., Naylor, B., & Towill, D. R. (2000). *Engineering the leagile supply chain*. *International Journal of Agile Management Systems*, 2(1), 54-61.
- Middelkoop, D. (2013, July 4). [Interview with Prorail simulation experts].
- Mol, J. P. M. (2009). *Railverkeer technisch ontwerp: Mistral Apeldoorn(incl.) - Deventer(excl.)*. Utrecht: Movares.
- Nash, A., Hürlimann, D., Schuette, J., & Krauss, V. P. (2004, January 9). *RailML - A standard data interface for railroad applications*. Paper presented at the 9. COMPRAIL, Dresden.
- ODETTE International Ltd. (2013, 2013). *Integrating SMEs in the automotive digital supply chain*. Retrieved July, 2013, from <http://www.odette.org/html/home.htm>
- Ohno, T. (1988a). *Preface in the English edition*. Cambridge, Mass.: Productivity Press.
- Ohno, T. (1988b). *Publisher's Foreword Toyota production system: beyond large-scale production* (pp. ix - xii). Cambridge, Mass.: Productivity Press.
- Ohno, T. (1988c). *Toyota production system: beyond large-scale production*. Cambridge, Mass.: Productivity Press.
- Palacios, E. G. (2013). [Interview: Siemens IT Development].
- Parry, G., Mills, J., & Turner, C. (2010). *Lean competence: integration of theories in operations management practice*. *Supply Chain Management: An International Journal*, 15(3), 216-226. doi: 10.1108/13598541011039974
- Prorail. (2010a). *Ontwerpvoorschrift EBS+ Inleiding* (001 ed.). Utrecht: Prorail.

- Prorail. (2010b). Ontwerpvoorschrift EBS+ Voorbereiding (001 ed., pp. 17-35). Utrecht: AM kwaliteitsmanagement.
- Prorail. (2010c). Ontwerpvoorschrift: Handleiding voor het opzetten voor OBE-bladen. In M. A. Treinbeveiligingssytemen (Ed.). Utrecht: AM kwaliteitsmanagement.
- Prorail. (2010d). Richtlijn voor het maken van FIS en RVTO. Utrecht: Prorail.
- Prorail. (2012a). Ontwerpvoorschrift flankbeveiliging bij rijwegen (003 ed.). Utrecht: AM infrasytemen.
- Prorail. (2012b). Ontwerpvoorschrift voor OR-bladen; Overzicht van symbolen. In M. A. Treinbeveiligingssytemen (Ed.), (002 ed.). Utrecht: Architectuur en Techniek.
- Prorail. (2012c). Programma van Eisen Interlocking - main-line corridors (002 ed.). Utrecht: AM kwaliteitsmanagement.
- Prorail. (2013a, 2013). Jaarverslag 2012. Retrieved July, 2013, from <http://www.jaarverslagprorail.nl/>
- Prorail. (2013b). Waar werkt Prorail aan het spoor? Retrieved July, 2013, from <http://www.prorail.nl/projecten-textueel>
- PTV AG. (2011). VISUM-RailML interface presented in Karlsruhe. In PTV AG (Ed.), *Software developers meet at 20th RailML Conference*. Karlsruhe: PTV AG.
- Quaglietta, E. (2013, February). *An overview on RailML data*. Paper presented at the Fourth RailML.org-interlocking-workconference, Berlin.
- Radtke, A. (2008). Infrastructure Modelling. In I. A. Hansen & J. Pacht (Eds.), *Railway Timetable & Traffic* (1 ed., pp. 43-57). Hamburg: Eurailpress.
- RailML.org. (2013a). RailML v2.2: RailML.org.
- RailML.org. (2013b, February 4 2013). RailML: Die XML-Schnittstelle für Eisenbahnanwendungen 2013, from <http://www.railml.org/index.php/index.html>
- Reemst, E. J. (2007). Op zaterdag 13 januari 2007 ontspoord om 15:43 uur een reizigerstrein op een wissel bij station Tilburg. (U. I. Onderzoek, Trans.). The Hague: Inspectie Verkeer en Waterstaat, Toezichteenheid Rail.
- Reuter, B., & Rohde, J. (2007). Enterprise Application Integration. In H. Stadtler & C. Kilger (Eds.), *Supply Chain Management and Advanced Planning* (4 ed., pp. 257-259). Berlin: Springer-Verlag
- RIGD-Loxia. (2011). Producten en Diensten Catalogus. In Prorail (Ed.), *Prorail*. Utrecht: Prorail.
- Rio, A. F., Rasoamanana, M., & Dumont, p. (2012). Belgium Patent No. European Patent Office: A. T. SA.
- Rockwell Automation Inc. (2010). ARENA online help: Expression module (Version 13.90). Wexford: Rockwell Automation Inc.
- Rother, M., & Shook, J. (2009). *Learning to see : value-stream mapping to create value and eliminate muda; a lean tool kit method and workbook*. Cambridge, Mass.: The Lean Enterprise Institute.
- Schut, G., & Dragt, S. (2013). [Interview: Loxia].
- Shararah, M. A., El-Kilany, K. S., & El-Sayed, A. E. (2011). *Value Stream Map Simulator using ExtendSim*, London.
- Sheahan, J., & Jones, R. (2012, November 29). Siemens says Invensys Rail deal will boost profits. Retrieved May 5, 2013, from <http://www.reuters.com/article/2012/11/29/us-invensys-siemens-idUSBRE8AS0AH20121129>
- Siefer, T. (2008). Simulation. In I. A. Hansen & J. Pacht (Eds.), *Railway Timetable & Traffic* (1 ed., pp. 155-169). Hamburg: Eurailpress.
- Siemens. (2012). Company Report 2012 (pp. 78). Berlin and Munich: Siemens.
- Siemens. (2013a, April 19). Anti-trust authorities give their consent to Siemens' acquisition of Invensys Rail. Retrieved May 5, 2013, from <http://www.siemens.com/press/en/pressrelease?press=/en/pressrelease/2013/infrastructure-cities/lic20130419004.htm>
- Siemens. (2013b). Rail Automation Prozesshaus: Siemens AG.
- Siemens AG. (2013). Systembeschreibung Elektronisches Stellwerk Trackguard Simis W Release 3.6 (R. Automation, Trans.) (Release 3.6 ed.). Braunschweig: Siemens AG.
- SporenplanOnline (Cartographer). (2010). Tekeningen met spoornummers en seinen Nederland [Railway track maps].
- Sprengers, Y. L. J., & Beelaerts van Blokland, W. W. A. (2013). *Development and application of a value assesment model to translate customer value on product functinonality to an optimal vlaued product cost during new product development in a concurrent environment*. Delft University of Technology. Delft.
- Stadtler, H. (2007). Supply Chain Management - An Overview. In H. Stadtler & C. Kilger (Eds.), *Supply Chain Management and Advanced Planning* (pp. 9-36). Berlin: Springer-Verlag.
- Storck, J., & Dragt, S. (2013). [Interview: Prorail Safety Systems].
- Supply Chain Council. (2010). Supply Chain Operations Reference Model (SCOR) model: Overview version 10.0. Cypress, TX, USA: Supply Chain Council.
- Surie, C., & Wagner, M. (2008). Supply Chain Analysis. In H. Stadtler & C. Kilger (Eds.), *Supply Chain Management and Advanced Planning* (pp. 41-48). Berlin: Springer-Verlag.
- Taguchi, G. (1986). *Introduction to quality engineering : designing quality into products and processes*. Tokyo: Asian Productivity Organization.
- Teisman, G. R. (2000). Models for Research into Decision-making Processes: on Phases, Streams and Decision-making Rounds. *Public Administration*, 78(4), 937-956.
- The Lean Enterprise Institute. (2009). What is lean? Retrieved April 9th, 2013, from <http://www.lean.org/WhatsLean/>
- Theeg, G., & Lykov, A. (2009). Movable Track Elements. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 149-178). Hamburg: Eurailpress.

- Theeg, G., Maschek, U., & Nasedkin, O. (2009). Interlocking Principles. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 61-112). Hamburg: Eurailpress.
- Theeg, G., Nasedkin, O., Maschek, U., Stratton, D., Tillmanns, H., White, T., & Mongardi, G. (2009). Interlocking Machines. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 252-305). Hamburg: EurailPress.
- Theeg, G., Svalov, D., & Schoene, E. (2009). Opening and closing of level crossings. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 380-383). Hamburg: Eurailpress.
- Theeg, G., & Vlasenko, S. (2009a). ETCS. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 240-251). Hamburg: Eurailpress.
- Theeg, G., & Vlasenko, S. (2009b). Group 5: Systems with Continuous Transmission at High Data Volume and Dynamic Speed Supervision. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 238-240). Hamburg: Eurailpress.
- Thissen, W. A. H., Phaff, H. W. G., & Pruyt, E. (2008). Model Testing SPM 2313 - Continuous Models 2, System Dynamics Lecture Notes (pp. 101-114). Delft: TU Delft.
- Thompson, H. S., Mendelsohn, N., Beech, D., & Maloney, M. (2012, April 5). W3C XML Schema Definition language (XSD) 1.1. Retrieved July 18, 2013, from <http://www.w3.org/TR/xmlschema11-1/>
- Trinckauf, J. (2009). Basic Characteristics of Railway Systems and the Requirements for Signaling. In G. Theeg & S. Vlasenko (Eds.), *Railway Signalling & Interlocking* (pp. 18-21). Hamburg: Eurailpress.
- van 't Hoff, E. (2013). [Siemens' planning of interlocking projects].
- van de Luytgaarden, E. (2007). *Reorganize the work package structure of Siemens Transportation Systems to be prepared for ProRails project Mistral*. (Master of Science Master), Delft University of Technology, Delft.
- van den Nieuwendijk, S. (2010). *Offerte Mistral corridors*. Offerte Apeldoorn-Deventer, offerte Maastricht-Sittard and offerte Den Dolder-Amersfoort.
- van der Meij, D., van der Werff, M., Janssen, B., Bartholomeus, M., & Dragt, S. (2013, May 21). [Intermediary Prorail RailML Project Meeting].
- van Vliet, J. H. (2004a). Op 27 maart 2004 omstreeks 13:00 uur rijdt trein 2249 ter hoogte van station Den Haag Mariahoeve voorbij stoptonend sein 184. Utrecht: Inspectie Verkeer en Waterstaat divisie Rail.
- van Vliet, J. H. (2004b). Seinveiligheidsstoring te Rotterdam-Lombardijen op 20 augustus 2003: Inspectie Verkeer en Waterstaat divisie rail.
- Verbraeck, A., & Valentin, E. C. (2006). *Discrete modellen Deel 2: Discrete Simulatie* (Vol. 2). Delft: TU Delft.
- Vocht, A. d. (2008). *Basishandboek SPSS 16 : statistiek met SPSS 16*. Utrecht: Bijleveld Press.
- Wendler, E. (2008). Queueing. In I. A. Hansen & J. Pachl (Eds.), *Railway Timetable & Traffic* (1 ed., pp. 106-117). Hamburg: Eurailpress.
- Wiggenraad, P. (2012). CT4811 Design and Control of Public Transport Systems lecture 3: Timetable Design, Blocking Times. Retrieved from BlackBoard TU Delft website:
- Williamson, O. E. (1981). The Economics of Organization: The Transaction Cost Approach. *American Journal of Sociology*, 87, 548-577.
- Winter, P. (2009a). *Compendium on ERTMS European rail traffic management system*. Hamburg: Eurailpress.
- Winter, P. (2009b). Train control-command: the ETCS developments. In I. U. o. Railways (Ed.), *Compendium on ERTMS European rail traffic management system* (pp. 69-144). Hamburg: Eurailpress.
- Wisotzki, B., & Muehlhause, M. (2013). [Interview: Lean Engineering / RailML].
- Womack, J. P., & Jones, D. T. (1996). From lean production to the lean enterprise. *IEEE Engineering Management Review*, 24(4), 38-46.
- Womack, J. P., & Jones, D. T. (2003a). *Lean thinking banish waste and create wealth in your corporation* (2nd ed.). New York: Free Press.
- Womack, J. P., & Jones, D. T. (2003b). Lean thinking versus German technik *Lean thinking banish waste and create wealth in your corporation* (2nd ed., pp. 189-218). New York: Free Press.
- Womack, J. P., Jones, D. T., & Roos, D. (1990). *The machine that changed the world; how Japan's secret weapon in the global auto wars will revolutionize western industry*. New York: Harper.

Abbreviations

ARENA	A discrete event simulation program	SIL	Safety Integrity Level
ATB	Automatische TreinBeïnvloeding	SIMIS-W	Siemens' IXL a.o. for the Netherlands
ATB-EG	Automatische TreinBeïnvloeding Eerste Generatie	SIMIS-C	Siemens' former IXL a.o. for the Netherlands
ATB-NG	Automatische TreinBeïnvloeding Nieuwe Generatie	SPSS	Statistical Package for Social Sciences
ATB-Vv	Automatische TreinBeïnvloeding Verbeterde Versie	Sptn	Santpoort Noord
ATP	Automatic Train Protection	SVA	Staat Van Aanwijzingen
AWS	Automatic Warning System	SWOD	SeinWezen OverzichtsDossier
Bv	Beverwijk	TEI	Total amount of Existing Interfaces
BITS	Beveiliging, Infrastructuur en Treinbewegingen Simulator	TOS	Technisch OS
C/T	Cycle Time	TPI	Total amount of Possible Interfaces
CIMOSA	Computer INtegrated Manufacturing Open System Architecture	VT-OBE	VerkeersTechnisch Overzicht Baan & Emplacement
CRS	Client Requirement Specification	TVD	Track Vacancy Detection
DES	Discrete Event Simulation	TVM	Transmission Voie-Machine
EAI	Enterprise Application Integration	UML	Unified Modeling Language
EDI	Electronic Data Interchange	Utg	Uitgeest
ERTMS	European Rail Traffic Mangement System	VCT	Value Creating Time
ETCS	European Train Control System	VSM	Value Stream Mapping
ETML	European Train Management Layer	WIP	Work-in-Progress
EU	European Union	WW2	World War 2
EVP	Element VerbindungsPlan	XML	Extensible Markup Language
FIS	Functioneel Integraal Systeemontwerp	XSD	XML Schema Definition
GIS	Geographic Information System		
GUI	Graphical User Interface		
GSM-R	Global System for Mobile communication Rail		
Hlm	Haarlem		
HSL	High Speed Line		
HTML	Hypertext Markup Language		
ICT	Information & CommunicationTechnology		
IDEFO	Integration DEfinition for Function. The 0 refers to the most aggregated process level.		
INCA	EDI of InfraAtlas		
INESS	Integrated European Signaling System		
IT	Information Technology		
IXL	Interlocking system		
L/T	Lead Time		
NVAW	Non Value Added Work		
NVA WIP	Non Value Added Work-in-Progress		
OBE	Overzicht Baan en Emplacement		
OR	Overzicht Retourstromen		
OTS	Operations Time-Charts		
OS	Overzicht Seinbeelden		
PLC	Programmable Logic Controller		
PPE	Property, Plant & Equipment		
RAMS	Reliability, Availability, Maintenance and Safety		
RailML	Rail XML		
RBC	Radio Block Center		
RVTO	RailVerkeers Technisch Ontwerp		
SC	Supply Chain		
SCLI	System Level Coupling Index		
SCOR	Supply Chain Operations Reference		