Informatics

Coordinated Multi-View Graph Analysis and Improvement

with Applications in Cyber-Physical Production Systems Engineering

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Christian Engelbrecht, BSc
Matrikelnummer 01529362

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl
Mitwirkung: apl. Prof. Dr.-Ing. habil. Arndt Lüder

Wien, 10. September 2021

Christian Engelbrecht	Stefan Biffl

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Informatics

Coordinated Multi-View Graph Analysis and Improvement

with Applications in Cyber-Physical Production Systems Engineering

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Christian Engelbrecht, BSc
Registration Number 01529362

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl
Assistance: apl. Prof. Dr.-Ing. habil. Arndt Lüder

Vienna, 10th September, 2021

Christian Engelbrecht	Stefan Biffl

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Erklärung zur Verfassung der Arbeit

Christian Engelbrecht, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. September 2021

Christian Engelbrecht

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Ich bin allen Personen, die mein Studium und meinen akademischen Werdegang beeinflusst haben, zutiefst dankbar. Mein besonderer Dank gilt meinem Supervisor, Stefan Biffl, und Co-Supervisor, Arndt Lüder, für ihre wissenschaftlichen Einblicke, ihre detaillierte Betreuung und ihre allgemeine Unterstützung bei der Erstellung dieser Arbeit.

Weiters möchte ich mich bei allen Beteiligten des MDRE-Forschungsprojekts bedanken, das den Grundstein für diese Arbeit gelegt hat, speziell bei Kristof Meixner und Dietmar Winkler.

Mein aufrichtiger Dank gilt meinen Mitstreitern und Freunden, Christoph Burger, Dominik Kretz, Mustafa Isikoglu und Alexander Prock. Was als eine Gruppe von Studenten begonnen hat, die (glücklicherweise!) die gleichen Vorlesungen besuchten, hat sich zu einer großartigen Freundschaft entwickelt, die ich nicht mehr missen möchte. Ich bin ihnen dankbar für die Unterstützung, die sie mir während meines gesamten Studiums und beim Verfassen der Thesis zuteil haben werden lassen.

Des Weiteren möchte ich mich bei meiner Familie, meinen Freunden, meiner Partnerin und allen Personen, die mir nahe stehen, für die fortlaufende Unterstützung bei der Erstellung dieser Arbeit bedanken, insbesondere während der zeitintensiven letzten Monate. Besonders möchte ich mich bei meinen Eltern bedanken, die mir mein Leben lang die notwendige Neugierde und Arbeitsmoral, die für diese Diplomarbeit erforderlich waren, beigebracht haben und mich über mein ganzes Studium hinweg unterstützt haben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

I am profoundly grateful for every person that influenced my studies and research. Special thanks to my supervisor, Stefan Biffl, and co-supervisor, Arndt Lüder, for their scientific insights, detailed supervision and overall support in writing this thesis.

Furthermore, I want to thank everyone involved in the MDRE research project, that laid the foundation for this thesis, namely Kristof Meixner and Dietmar Winkler.

My sincere thanks to my peers and friends, Christoph Burger, Dominik Kretz, Mustafa Isikoglu and Alexander Prock. What started as a group of students that (luckily!) shared the same courses, turned into a great friendship and group of mutual support in achieving our goals. I am eternally grateful for the support you provided me during my studies and thesis writing.

I further want to thank my family, friends, partner and every person involved in my life for the ongoing support in writing this thesis, especially during the time-intensive finishing months. Last, but not least, I especially want to thank my parents, who instilled the curiosity and work ethic in me that helped me finish this thesis and my master's degree.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Graphen sind eine allgemeine Klasse von Modellen, die aus Knoten und Kanten bestehen. In technischen Bereichen, wie der diskreten Fertigung, modellieren diese Graphen reale Artefakte. Cyber-Physical Production Systems (CPPS) Engineering, das sich mit vernetzten Produktionsanlagen befasst, verwendet Graphen mit mehreren Sichten, um Beiträge aus verschiedenen Ingenieursdisziplinen darzustellen, wie z.B. Pläne für einen Motor aus den Bereichen Mechanik, Elektrotechnik und Software/Automatisierungstechnik. Multidisziplinäres Engineering mit Abhängigkeiten zwischen den Modellen, die mehrere Sichten auf dasselbe reale Artefakt darstellen, erfordert Koordination zwischen den Ingenieuren, um verknüpfte Multi-View-Graphen zu analysieren und zu verbessern.

Dies führt zu Herausforderungen, die Koordination notwendig machen. Zu diesen Herausforderungen gehören die Darstellung und Verknüpfung gemeinsamer Knoten, z. B. von Motoren, in mehreren technischen Disziplinen mit gemeinsamen Eigenschaften, z. B. der Motorleistung. Darüber hinaus bestehen Herausforderungen bei der Koordination von Tasks in einem Graphen in CPPS Engineering, da die Unterstützung für integrierte Koordination auf Graphen fehlt.

Diese Arbeit schlägt den Ansatz von Coordinated Model Analysis and Improvement (CMAI) vor, um die Koordination zwischen Ingenieuren im Bereich des CPPS-Engineering zu verbessern. Um die Darstellung von Modellen im CPPS-Engineering zu erleichtern, führt diese Arbeit die CMAI-Domain-Specific Language (DSL) ein. Auf der Grundlage der CMAI-DSL wird in dieser Arbeit die CMAI Methode vorgestellt. Diese Methode nutzt die vorgeschlagenen Ansätze zur Verbesserung der Koordination zwischen den Ingenieuren, indem sie Daten zwischen Modellen automatisch synchron hält und Marker zur Koordination benutzt. Diese Marker bieten ein effizientes und integriertes Koordinationswerkzeug für Graphen. Darüber hinaus wird in dieser Arbeit ein technischer Prototyp vorgestellt, der Generic CPPS Modeling (GCM) Editor.

Diese Arbeit zeigt anhand einer qualitativen Evaluierung, wie ausgewählte Anwendungsfälle im Bereich des CPPS Engineering durchgeführt werden können und von der vorgestellten CMAI-Methode profitieren.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

Graphs are a general class of models that consist of nodes and edges. In technical domains, such as discrete manufacturing, these graphs model real-world artefacts. The field of Cyber-Physical Production Systems (CPPS) Engineering, concerned with interconnected production plants, uses multi-view graphs to represent contributions coming from several engineering disciplines, such as plans for a motor in mechanical, electrical, and software/automation engineering. Multi-disciplinary engineering with dependencies between the models, that represent multiple views on the same real-world artefact, require coordination between engineers for analysing and improving linked multi-view graphs.

This leads to challenges, founding the need for coordination. These challenges include the representation and linking of shared nodes, e.g., motors, in multiple engineering disciplines with shared properties, e.g., motor power. Furthermore, challenges exist in the coordination of tasks in a graph in CPPS Engineering, due to missing support for integrated coordination on graphs.

This thesis proposes the Coordinated Model Analysis and Improvement (CMAI) approach to improve coordination between engineers in the field of CPPS Engineering. To represent models in CPPS Engineering, this thesis introduces the CMAI-Domain-Specific Language (DSL). Based on the CMAI-DSL, this thesis introduces the CMAI method. This method leverages the proposed approaches to improve coordination between engineers by providing an automated way of keeping data in sync between models and introducing coordination markers. These markers provide an efficient and integrated coordination tool on graphs. Furthermore, this thesis introduces a technical prototype, the Generic CPPS Modeling Editor.

This thesis uses a qualitative evaluation to show how selected use cases in the field of CPPS Engineering can be conducted and profit off the introduced CMAI method.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

1.1 Motivation

Graphs are a general class of models that consist of nodes connected by edges and are widely used in information systems engineering and business informatics [West et al., 2001]. In the context of *Cyber-Physical Production Systems (CPPS) Engineering* [Lasi et al., 2014; Biffl et al., 2017], multiple engineering disciplines share selected model elements, often nodes that represent common concepts, such as motors or other parts of production systems. These multi-disciplinary models are often complex and may consist of hundreds or even thousands of nodes. Due to their multi-disciplinary nature, current research calls these graphs multi-view graphs, since they allow multiple views from different engineering perspectives [Biffl et al., 2021a].

Founded on these multi-view graphs, a minimal, exemplary use case for this thesis can be defined, introducing the issues and solutions treated in this work.

A typical use case in the field of CPPS Engineering, consists of designing a powertrain to move a load, e.g., designing a powertrain of a conveyor belt in a production system. For this, multiple different engineering disciplines have to be considered. Figure 1.1 illustrates an exemplary use case.

First, a **Requirements Engineer** needs to model the requirements that need to be fulfilled by the powertrain. For this, the engineer models requirements related to the motor, as Figure 1.1 shows. A **Mechanical Engineer** models all mechanical parts connecting to the motor, e.g., a transmission. Lastly, an **Electrical Engineer** is, e.g., concerned with designing the power supply for the drivetrain.

This simple use case shows multiple challenges, founding the need for coordination. As Figure 1.1 shows, a node represents the motor in each model and engineering discipline, whilst sharing properties in between them e.g., the motor power. This leads to the need that shared properties on cross-discipline artefacts have to be kept in sync. Since each engineering discipline uses its own Domain-Specific Language (DSL), these cross-discipline artefacts may not share all properties across engineering disciplines. Furthermore, engineers require coordination in their respective engineering view. This is needed to track the progress of a task on a model element, e.g., tracking the design or validation status of a model element. The outlined minimal use case shows the strong need for coordinated analysis and improvement in the context of multi-view graphs.

This need for coordinated analysis and improvement aims at improving efficiency, effectiveness, and correctness of the designed models. As checking and improving a model requires knowledge of the model and the application domain, in many cases, automated model checking has to be complemented with human-based reviewing. However, the coordination of humans and computers for analysing and improving models in CPPS Engineering requires improved methods and mechanisms to achieve an improvement in efficiency, effectiveness, and correctness. Furthermore, human-based reviews take significant effort and are prone to overlook issues. This is especially relevant in cross-disciplinary domains, where multiple engineering disciplines need to manually review a big set of relevant nodes, each relevant to multiple engineering disciplines.

To improve on the status-quo, human-human and human-computer based interaction needs to be improved, when conducting tasks to analyse and improve models, resulting in an overall increase in efficiency, effectiveness and correctness, freeing resources of domain experts. Furthermore, enforcing human-computer based interaction allows leveraging new approaches, such as querying models using a user-friendly model query language, e.g., *Cypher* [Neo4J, 2020]. These queries allow domain experts to iteratively explore the model and analyse relevant subsets of nodes and links for issues. This allows the reviewer to mark the involved nodes or edges. These marks could allow automated processing, e.g., the creation of *Model Analysis and Improvement (MAI)* tasks based on selected marker types. Furthermore, these markers could facilitate the coordination of improvement tasks between the domain experts.

Finally, the results of model analysis and improvement tasks could be input to follow-up processes and engineers can integrate them into the underlying large multi-view model, e.g., designing a production plant consisting of potentially hundreds of interconnected conveyor belts. From this input, improvement tasks could be created that can be implemented by domain experts to improve the underlying large multi-view model.

In the current state of the art, coordination often happens via external tools, e.g., Issue Trackers like JIRA [1] or via E-Mail, which leads to a very coarse level of detail and is detached from the actual, underlying model for which coordination happens. This is very inefficient and error-prone, further founding the need for improving coordination. The goal of improving coordination is to make engineering in the field of CPPS Engineering more agile and efficient, while reducing overall risk and increasing overall quality of models in CPPS Engineering.

[1]`https://www.atlassian.com/de/software/jira` Last accessed: 03/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

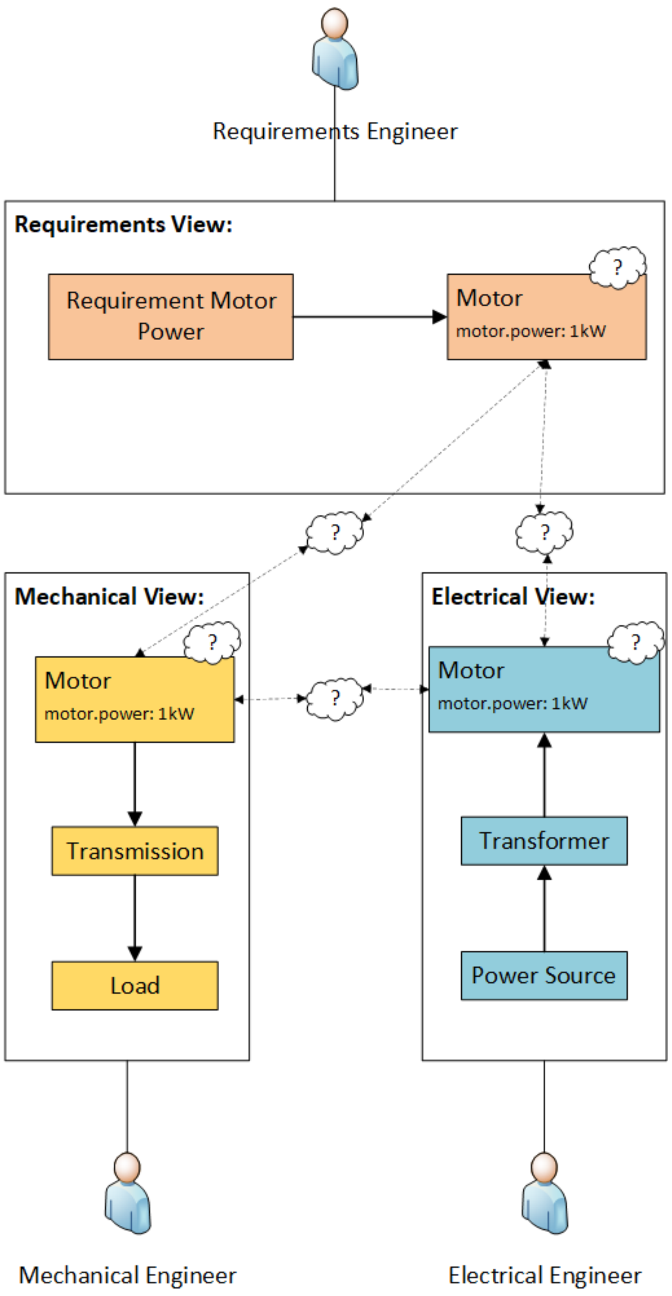


Figure 1.1: Simple use case outlining main issues addressed by this thesis. The question marks represent the challenges outlined by the use case.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1.2 Problem Statement

As outlined, the motivation of this thesis lies in the coordinated model analysis and improvement. Main contributing factors are challenges existing in the field of CPPS Engineering.

Major challenges in *model analysis and improvement* include

- C1. *Limited analysis and improvement resources*, often experts with sufficient domain/world and modelling knowledge. This is outlined in the above use case. Domain experts are difficult to obtain and limited in the available time for their reviews [Biffl et al., 2016].
- C2. *Large models and expensive modelling tools* that are difficult to use. Existing modelling tools like AutoCAD [2] are expensive and very complex. Furthermore, they focus on the designing of artefacts, instead of their analysis and improvement.
- C3. *Limited approaches for coordinating analysis and improvement processes* of a group of human experts with automation support. Relating to the use case and C1, only limited approaches for coordinating analysis and improvement processes exist. Improving this, and providing automation support, would improve the efficiency of these processes, freeing up resources for further analysis and improvements [Biffl et al., 2016].

This work aims at addressing these challenges by designing and evaluating representations of coordination data in multi-disciplinary multi-view graphs, methods for *Model Analysis and Improvement (MAI)* in a team, and software prototypes in MAI application scenarios. Due to the multi-disciplinary nature of the MAI application scenarios, the term of Coordinated Model Analysis and Improvement (CMAI) is introduced, specifically focusing on the coordinated nature of these *analysis and improvement* tasks.

1.3 Overview on Expected Results

Figure 1.2 represents the expected results of this thesis as outputs in an IDEF0 diagram.

Use Case Descriptions. As a first step, this thesis will elicit use cases in the field of CPPS Engineering to guide the design and evaluation tasks. These use case descriptions form part of Chapter 4.

Detailed requirements. Based on the use case descriptions, more detailed requirements will be outlined for each use case. The requirements will allow to further settle the thesis in the field of CPPS Engineering. Section 3.4 defines an initial, high scope, set of requirements for this thesis.

[2]`https://www.autodesk.de/products/autocad/overview` Last accessed: 05/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

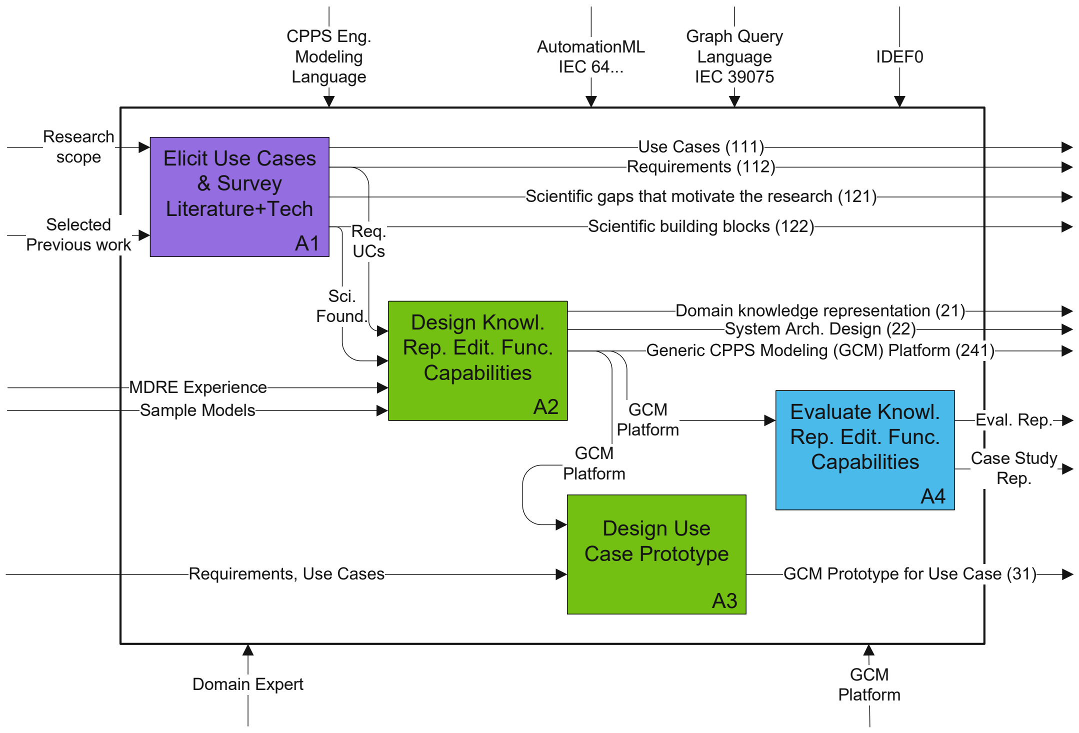

CPPS Eng. Modeling Language
AutomationML IEC 64...
Graph Query Language IEC 39075
IDEF0
Research scope
Selected Previous work
Elicit Use Cases & Survey Literature+Tech A1
Use Cases (111)
Requirements (112)
Scientific gaps that motivate the research (121)
Scientific building blocks (122)
Req. UCs
Sci. Found.
Design Knowl. Rep. Edit. Func. Capabilities A2
Domain knowledge representation (21)
System Arch. Design (22)
Generic CPPS Modeling (GCM) Platform (241)
MDRE Experience
Sample Models
GCM Platform
Evaluate Knowl. Rep. Edit. Func. Capabilities A4
Eval. Rep.
Case Study Rep.
Design Use Case Prototype A3
Requirements, Use Cases
GCM Prototype for Use Case (31)
Domain Expert

Figure 1.2: Outlook on work results and work packages.

Scientific gaps that motivate the research. The use case descriptions and the requirements allow identifying gaps in the research to motivate the research in this thesis.

Domain knowledge representation. A major result of this thesis will be the CMAI-DSL. This DSL allows definition of all needed model elements introduced by the use cases in Chapter 4.

This thesis will build on strengths of existing approaches for designing applications for model analysis and improvement powered by coordinated human experts and computers. To allow for a coordinated analysis and improvement by the domain experts, this thesis will introduce the concept of *coordination markers*. Coordination markers can represent the goals, model issues and improvement tasks to address. These annotations are configurable and allow the configuration of a set of states, e.g., representing the resolution state of tasks. Expected strengths of this approach include (a) these markers to be both usable by the human review team and computer algorithms and (b) enabling the configuration of markers for users, not only software developers (c) integrated placement on specific model elements, not relying on external tools.

System architecture design. An expected result of this thesis is to provide a system architecture design that allows addressing a range of typical use cases. Chapter 5 outlines

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

these theoretical and technical solutions for the architecture design.

Evaluation & Discussion. Founded on the system architecture design, the technical feasibility will be evaluated. Based on this evaluation, a report on the technical feasibility will be compiled and discussed. This will be conducted in Chapter 6 and Chapter 7.

Software Prototype. This software prototype will provide an automation of selected steps of the CMAI method to make the process more efficient and allow human experts to focus on contributions that only they can provide, e.g., taking decisions that require deep domain knowledge. The proposed prototype is the Generic CPPS Modeling (GCM) Editor. This name ties to the field of application of this thesis, CPPS Engineering.

1.4 Reader Guidance

This thesis provides results for several groups of readers: (1) practitioners in the field of CPPS Engineering, (2) readers working in management positions, (3) researchers, and (4) software engineers.

(1) Practitioners in the field of CPPS Engineering can focus on the results of typical use cases in their fields of CPPS Engineering. This provides new insights and approaches on real-world applications in the field. Therefore, Chapter 4 and Chapter 6 may prove helpful to these readers. (2) Readers working in management positions may be interested in the proposed approaches, that improve overall model quality and coordination, which may lead to overall better results in real-world CPPS applications. For these readers, the results evaluated and discussed in Chapter 6 and Chapter 7 may prove helpful. (3) Researchers may pick up results from this thesis and build their own research on it. Chapter 8 outlines future work. (4) Software Engineers can use the GCM Editor platform and architecture as base for further development and augmentations. (Code) Artefacts developed in this thesis can be found in an external repository [3].

1.5 Thesis Structure

The remainder of this thesis follows this structure: To establish the connection to the current state of research and state of practice in CPPS Engineering, Chapter 2 introduces the state of the art.

Based on the state of the art, Chapter 3 outlines the research questions and methodological approach for this thesis.

To evaluate the research questions, Chapter 4 introduces illustrative use cases in the field of CPPS Engineering. This section leverages the state of the art on requirements engineering to define all used use cases.

[3] Model Configuration Repository: `https://bitbucket.org/tuw-qse/mdre-thesis-cengelbrecht`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Founded on the resulting requirements of the research questions and use cases, Chapter 5 introduces the solution approach. This is done first from a theoretical standpoint, to allow for a conceptual architecture. To evaluate the use cases, these theoretical concepts then lead to technical implementations, leading to the GCM Editor.

Chapter 6 reports on the evaluation of the use cases.

Chapter 7 discusses this evaluation.

Finally, Chapter 8 concludes the results of the thesis and gives an outlook on future work in the research field of this thesis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 2

State of the Art

This section summarizes background and related work on Cyber-Physical Production Systems (CPPS) Engineering, multi-view graphs in CPPS Engineering, model representation and querying, requirements engineering and the topic of stigmergic coordination. The goal of this chapter is to provide an overview on these topics and to settle the work in the current state of the art.

This thesis choses these topics to define the state of the art, that this thesis bases on. First, this chapter introduces the state of the art in CPPS Engineering to outline the state of the art in the field of application. Second, this chapter introduces the state of the art on multi-view models in CPPS Engineering to outline current limitations in the engineering domain. Third, to aid the definition of the software architecture, this chapter outlines the state of the art on model representation, storage, and querying. Fourth, to allow the definition of the illustrative use cases in Chapter 4, this chapter introduces the state of the art approaches in Requirements Engineering. Lastly, this chapter outlines the term and state of the art of Stigermgy, with an added focus on existing coordination approaches in Collective Intelligence Systems (CIS).

2.1 Cyber-Physical Production Systems Engineering

This section introduces the term of Cyber Physical Systems (CPS) and its foundation for CPPS Engineering. It settles the concept of CPPS Engineering in its current state of the art, its unique characteristics and its current limitations, leading to gaps in the research that this thesis aims to fill. It provides an overview of the current state of the art in CPPS Engineering, allowing to gain insight on the field of application of the use cases of this thesis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2.1.1 Cyber Physical Systems (CPS)

This section introduces the underlying concept of CPPS, the CPS. One of the first definitions of the concept of CPS can be found in Lee [2006]. Lee [2006] defines CPS as a combination of computation and the physical processes in the real world. These physical processes impact the computations, which lead to a reaction in the system, e.g., the anti-lock braking system in a car continuously reacts to new input of the physical world to adapt the braking. This leads to a continuous feedback loop between the computation and the physical world.

The possible applications for CPS that Lee [2006] outlines are numerous. Possible applications are in the field of transportation logistics, where interconnected autonomous vehicles could improve safety, networked building control systems or the possibility to track goods via radio-frequency identification (RFID) chips. All these application possibilities fall under the broad term of the Internet of Things. Overall Lee [2006] concludes that with a tight interconnection and integration of CPS in the real world a "programmable matter" could become a reality.

Based on the introduction of the concept of CPS in Lee [2006], the scientific community conducted further research and the term of CPS gained traction. This is outlined by Cardin [2019], that outlines that between 2007 and 2017 the scientific published works containing the term "cyber-physical" rose from 19 publications in 2007 to over 5000 publications in 2017. This immense growth of publications in the field of CPS also led to new and improved definitions of the term.

Monostori [2014] defines Cyber Physical Systems (CPS) in the following way: *"Cyber-Physical Systems (CPS) are systems of collaborating computational entities which are in intensive connection with the surrounding physical world and its ongoing processes, providing and using, at the same time, data-accessing and data-processing services available on the internet"* [Monostori, 2014]. For Cardin [2019] it is necessary that the CPS have sensors and actors to react to the environment and take actions. Furthermore, they need the capability to communicate with humans via interfaces.

Figure 2.1 shows a schematic overview over the different parts of a CPS. As can be seen, the software and hardware drive the sensors and actuators of the system. Uniting the software, hardware, sensors, and actuators is the so called *embedded system*. This embedded system is a part of the physical system, where the naming of Cyber Physical Systems (CPS) becomes clear. The system as a whole can then connect and interact with other systems, but also with humans via human-machine interfaces.

As Cardin [2019] outlines, based on this definition, the term CPS covers a very broad field of systems and applications. This can be seen as an opportunity, since one broad term and definition consolidates a large field of possible applications. Similarly to Lee [2006], Cardin [2019] also sees possible applications based in the field of autonomous vehicles and building control.

This thesis builds on the term of CPS as introduction into the field of CPPS, which

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

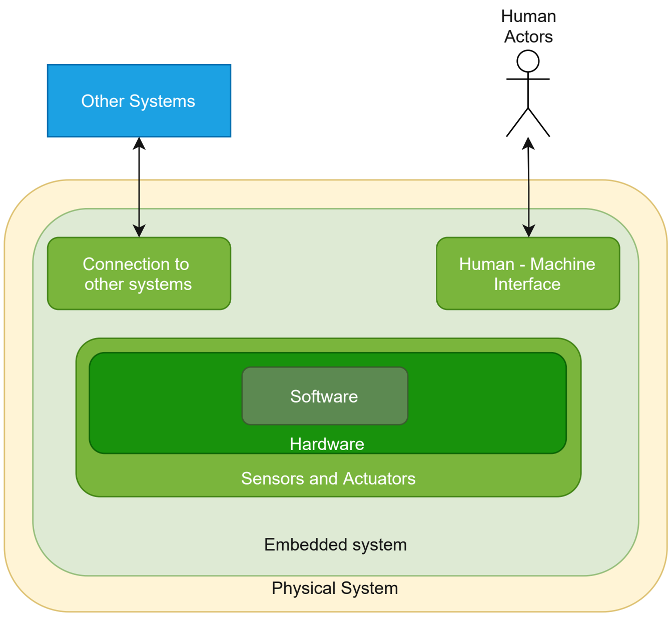


Figure 2.1: Schema of CPS, based on [Monostori, 2014].

represents the application context of the Model Analysis and Improvement (MAI) tasks and the nature of CPPS Engineering Networks (CENs). Furthermore, due to the nature of the proposed MAI flow, the flow can also be applied to CPS or any other system that is represented by a (multi-view) graph.

2.1.2 From CPS to CPPS

As outlined, the field of applications for CPS is extremely wide. This is why the production domain is very interested in the field of CPS [Cardin, 2019]. Due to this interest of the production domain on CPS, the term of Cyber-Physical Production Systems (CPPS) is introduced [Monostori, 2014].

The following definition of a CPPS is suggested by Cardin [2019]. *"Cyber-Physical Production Systems are systems of systems of autonomous and cooperative elements connecting with each other in situation dependent ways, on and across all levels of production, from processes through machines up to production and logistics networks, enhancing decision-making processes in real-time, response to unforeseen conditions and evolution along time"* [Cardin, 2019]. The adaptability and decision-making of CPPS is especially focused on in this definition. Furthermore, the definition encompasses the distributed nature of CPPS, where coordination and cooperative elements between the systems in the production domain are relevant.

Due to these characteristics of CPPS, Monostori [2014] sees many expectations and applications for these systems, although expectations for CPPS are sometimes exaggerated. Due to the possibility of communication between humans, products, and machines in

TU Bibliothek, WIEN Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CPPS the possibilities for applications in the real world are manifold. Possible applications include production systems, logistical systems and even smart cities, where technology interconnects all aspects of the cities, which could lead to an improvement in quality of life [Monostori, 2014].

This thesis builds on the term of CPPS as a field to conduct the use cases in the Coordinated Model Analysis and Improvement (CMAI) method. The distributed nature of CPPS, as outlined above, involves multiple engineering disciplines and actors. This leads to the need of coordinated representation and also coordinated improvement of graphs in the field of CPPS Engineering.

2.1.3 Foundation of Industry 4.0

As outlined by Monostori [2014] and Cardin [2019], CPPS are a very disruptive concept in the industry domain. According to Monostori [2014] and Cardin [2019], CPPS could even lead to the fourth industrial revolution, due to the possibilities of application in the production domain.

Due to this, the German Government introduces the term Industry 4.0 in an effort to prepare and to name the next industrial revolution [Vogel-Heuser et al., 2020]. The goal of the Industry 4.0 is the digitization and linking of production systems in flexible manufacturing plants [BMBF, 2018]. This is why the term Industry 4.0 strongly builds on the concept of CPPS. As outlined before, the term CPPS refers to a production system with full integration between IT systems, the production system, and interconnected components. The goal is to improve the overall efficiency and effectiveness in modelling and collaboration in the CPPS context [Lasi et al., 2014].

As shown in the exemplary use case introduced in Chapter 1, the multi-disciplinary nature and the parallel engineering in the field of Industry 4.0 fuels the need for improved coordination in the domain.

For this thesis, the disruptive possibilities of CPPS and the development of the term Industry 4.0 in Germany warrant the conduction of typical use cases in the field of CPPS Engineering. This shows that the conducted use cases are related to real life issues that have gained traction in the last years in the research field and build on state of the art issues.

2.2 Multi-view Graphs in CPPS Engineering

As shown in the exemplary use case in Chapter 1, typical use cases in CPPS Engineering consist of multiple views on the same real world artifact. These concept of multiple views, are called multi-view graphs.

As the context of CPPS Engineering is a heterogeneous and a multi-disciplinary engineering domain, it leads to typical problems in coordination, data exchange and overall data integration. This is further enforced due to multi-view graphs, e.g., properties of shared

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

nodes between different views in multi-view graphs [Biffl et al., 2017, 2021a]. Based on this, further problems in validation of models and risk-mitigation between engineering disciplines arise. Typical problems consist of working on stale data, that already was modified by another engineer in a different view, or missing local coordination approaches to aid coordination between engineers of one discipline [Biffl et al., 2017, 2021a]. To improve on this, different approaches have been proposed in the research community.

These approaches, namely CENs and Round Trip Engineering, are introduced in this section. Chapter 5 uses the introduced concepts as building blocks for the solution approach of this thesis.

2.2.1 CPPS Engineering Networks (CENs)

The work of Atkinson et al. [2015] works on so called multi-view models. In this approach, each engineering discipline models their respective domain specific models in their own views. This leads to one view per engineering discipline, but multiple views per system, due to the multi-disciplinary nature of CPPS.

A strength of this approach is that engineers can work following their normal working habits. Since every view has its own Domain-Specific Language (DSL), engineers can work with their domain-specific modelling tools and representations. This results in multiple domain specific representations of the system [Atkinson et al., 2015; Lüder et al., 2019; Biffl et al., 2021a].

These domain specific representations allow to form a network between the engineers and different parts of the system [Schäffler et al., 2013]. This approach also has its limitations, especially concerning data exchange between domain experts. Due to the scattered nature of the domain, data aggregation, integration, and transformation is a resource intensive and difficult task [Biffl et al., 2021a].

In the context of CPPS Engineering, the term of CPPS Engineering Networks (CENs) was introduced by Biffl et al. [2021a]. This term describes multi-view models, as defined by Atkinson et al. [2015], but in the domain of CPPS Engineering. In this thesis, the term of CENs will also be used when talking about multi-view models provided by different engineering disciplines.

As outlined above, the multi-view modelling approach has its limitations regarding data exchange and data transformation. As a way to combat this, work on Round Trip Engineering has been done and will be introduced in the following section.

2.2.2 Round Trip Engineering

To improve collaboration and to foster interaction between engineering disciplines, a way to exchange modelling data is crucial, as Lüder et al. [2018] outline. Limitations in this field result from the diversity of the tools used by engineers, but also by the engineering habits of the different engineers. These engineers normally use domain specific tools, tool chains, and optimized processes for their field. The usage of data logistics in general, more

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

specifically an approach of round-trip engineering, leads to a disruption of their working environment. This may lead to backlash by the engineers, since they lose efficiency by changing their optimal local working environment [Lüder et al., 2018].

For Lüder et al. [2018], the engineering networks in production systems are a data logistics systems in itself. Due to this, data exchange is necessary in the multi-disciplinary network. Based on the data exchange and data logistic approach, the research community conducted works on Model Improvement and Risk Assessment in multi-disciplinary Engineering [Winkler et al., 2017; Biffl et al., 2020, 2021a].

As a subset of data logistic in multi-disciplinary systems, Round Trip Engineering is a common use case in the field. It allows distributed and parallel work and collaboration. In general, this Round Trip Engineering describes a flow where a data provider provides data, also called data source. Based on this data, an engineer transforms it to its domain specific data format, works on it, and provides it to the next engineer. This engineer again transforms the data to its local format, works on it, and provides it again. This can be done as many times as needed. As a last step, the data can be transformed into the origin format or be provided to a data sink, which uses the resulting modelling data [Winkler et al., 2017; Lüder et al., 2019].

This thesis uses the concept of Round Trip Engineering in Chapter 4 and Chapter 5. It allows to represent the "perfect engineering world", as outlined by Winkler et al. [2017]. To better understand this concept, see Figure 2.2.

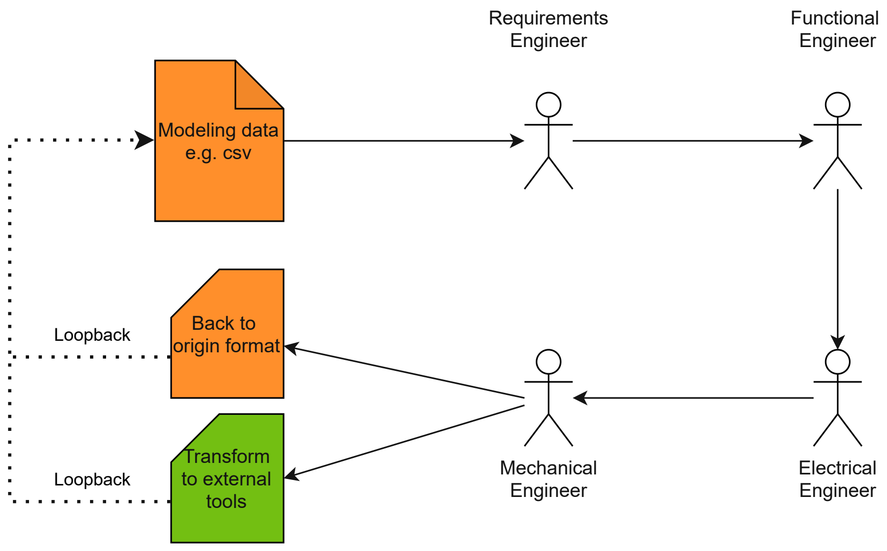


Figure 2.2: Round Trip Engineering flow in the context of a MAI task in the context of a CEN. Based on Winkler et al. [2017] and Lüder et al. [2019]. Due to the usage of domain-specific DSLs by each Engineer between each engineer, a transformation step happens. This allows each engineer to work in its own DSL.

The technical architecture introduced in Chapter 5 leverages the concept of Round

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Trip Engineering. Round Trip Engineering furthermore allows settling the technical architecture into the current state of the art in the engineering world, by allowing import and export of different formats, e.g., a list of comma-separated values (CSV). The usage of external tools, such as Neo4J, also leverages this approach. This technical state of the art will be introduced in the following section.

2.3 Model Representation, Storage, and Querying

To develop the prototype proposed in Chapter 5, the state of the art for model representation, storage, and querying needs to be established for this thesis. This allows to propose technical state of the art solutions while leveraging current developments in graph storing and querying.

This section outlines the relevant topics as follows: First, this section establishes the current state of the art solutions for model representation and exchange. Secondly, it introduces the technical solutions used for Model Storage and Querying. Lastly, this section reasons the technical decision taken for this thesis in the context of model representation, exchange, and querying. Chapter 5 then applies these technical solutions.

Model Representation and exchange. A well established approach in the field of *Model Driven Systems Engineering (MDSE)* is the *Eclipse Modeling Framework (EMF)* [Brambilla et al., 2012]. EMF aims at bridging the gap between modelling and developers by treating models as first class entities. Based on these models, developers can create code, which allows for an easier adaption [Steinberg et al., 2008]. While EMF is strong for code generation from models, multi-disciplinary CPPS Engineering typically requires a heterogeneous network of discipline-specific engineering tools that require the integration of multi-view models. Therefore, EMF is more focused on building code from models automatically. In contrast, CPPS Engineering aims at modelling real world artefacts, e.g., a production line.

As outlined in Section 2.2.2, the approach of *round-trip engineering* is common in the field of CPPS Engineering. To allow for such an approach, model transformation and exchange forms an important part. In the last years, for model representation and exchange in the field of CPPS Engineering, *Automation Markup Language (AML)* has emerged as the open IEC 62714 standard [IEC 62714-1, 2018] for technology- and vendor-agnostic industry data representation in an XML Format. It aims at providing a data format that allows domain and company-wide data exchange by providing a common standard [Biffl et al., 2017, 2019b; Lüder et al., 2019; Lüder et al., 2020].

Based on the Automation Markup Language (AML) standard, other data logistics and data exchange approaches have been gaining traction in recent years [Biffl et al., 2019a]. Such an approach has been introduced by Biffl et al. [2019a], the Engineering Data Exchange (EDEx). The approach aims at providing a more general model exchange approach, that is not restricted to the AML environment.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

	Neo4J	MySQL
100 000int	18.0 ms	69.8 ms

Table 2.1: Query time for a structural query in milliseconds. The graph consists of 100 000 nodes. Based on the results of Vicknair et al. [2010].

This work will take into account Automation Markup Language (AML) and other data exchange approaches and provide an interface, that allows model exchange and transformation. This allows the integration of the results of this work into existing round-trip engineering tool chains.

Model storage and querying. To store and query graph-oriented engineering models, *Graph Database Management System (GDBMS)* Platforms are a natural match. These platforms are optimized for storing hierarchical data, e.g., graphs and are based on *Not only SQL (NoSQL)* databases [Li, 2019]. To standardize the *Graph Query Languages (GQLs)*, the GDBMS community continues to develop the proposed standard *ISO/IEC WD 39075*. This standardization aims at providing a standard query language for GDBMS, similar to the SQL standard [ISO/IEC 9075, 2016] for relational databases. ISO/IEC WD 39075 [2020] develops this standard.

NoSQL databases, such as the platform *MongoDB*[1], use several storage formats, such as JSON-Objects[2]. For graph traversal queries, a graph database like *Neo4j*[3] performs substantially better than "classic" relational databases [Vicknair et al., 2010]. The speed difference is especially noticeable when working with bigger and more complex graphs, containing more nodes and edges. The work of Vicknair et al. [2010] measures this speed improvement. Table 2.1 shows selected values, comparing structural queries on stored graphs, between Neo4J and an SQL database.

Since the queries aimed at finding nodes on certain levels in the graph, e.g., finding nodes at a depth of 128, Neo4J performed substantially better. This is more noticeable when working on bigger graphs, as the runtime in Table 2.1 shows.

This work builds on *NoSQL* databases to facilitate the efficient and scalable storage and querying of large multi-disciplinary engineering models. This also influences the chosen data model for the proposed configuration of DSL as Chapter 5 outlines. The GDBMS approach used by Neo4J is leveraged as an approach to easily query and traverse graphs. For this, model transformation from the Generic CPPS Modeling (GCM) editor specific DSL is necessary. This allows users to use a GQL to semi-automatically modify existing graphs.

[1]`https://www.mongodb.com/de`Last accessed: 05/09/2021

[2]`https://www.ecma-international.org/publications/standards/Ecma-404.htm`Last accessed: 05/09/2021

[3]`https://neo4j.com/` Last accessed: 05/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.4 Requirements Engineering

As Hofmann and Lehner [2001] outline, Requirements Engineering is an integral part in achieving success within a software project. Requirements Engineering is especially useful in software-intensive systems and plays an important part in Software Engineering. Due to the nature of CPPS Engineering, it can profit off of established approaches of Requirements Engineering in Software Engineering.

To elicit, validate and verify requirements relevant to the use cases in this thesis, this section introduces three approaches to requirements engineering. Each of these approaches allows to elicit a unique view on the requirements in this thesis. This section starts from a more general, system-wide approach, focusing on agents and processes, represented by I-Star (I*) and IDEF0, to a more use case specific approach, represented by *Restricted Use Case Modeling (RUCM)*. The following sections introduce the State of the Art of these approaches to aid the definition of the use cases elicited and conducted in Chapter 4 and Chapter 5.

2.4.1 I*

I*, also known as I-Star, is a modelling framework for requirement engineering, proposed by Yu [1997]. Its aim is to focus on and to understand the *Why?* in the requirement elicitation process for software, instead of focusing on the *What?*, e.g., not asking *What* features do certain stakeholders need, rather asking *Why* do stakeholders require certain features. This modelling framework is especially tailored to be used in the early phases of requirements engineering. The I-Star approach allows to gain a broad overview over the involved stakeholders and computer systems, especially when involving multiple actors and subsystems. Furthermore, I-Star allows an overview over the relations and dependencies in-between these involved parties [Yu and Mylopoulos, 1994; Yu, 1995, 1997, 2001; Atkinson et al., 2015].

The main concept of the I* modelling approach are the so-called *agents*. For Yu [2001] the modelling concept of *agents* has certain properties, that affect the modelling process:

Intentionality. Each agent in the system has its own goals, abilities, etc. Based on these goals, each agent acts intentionally according to its own beliefs. Following these goals and abilities of each agent, it is possible for the modeller to conduct a high-scope elicitation of requirements in the system.

Autonomy. As outlined by Yu [2001], each agent acts on its own initiative and autonomous from other agents in the system. This means that an agent is not fully predictable or knowledgeable beforehand to the modeller.

Sociality. Since agents are social, the relationships between the different agents in the system are very important. These relationships are complex and multilateral, but settle the agents in a social network.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contingent Identity and Boundaries. Agents can be abstract in I*. This allows the representation of higher-scope agents, e.g., groups and teams of social agents. Furthermore, the boundaries of responsibility for each agent are changeable and not fixed from the start. These can be changed as needed, e.g., when the responsibility for a task shifts between different agents the boundary for responsibility changes as well.

Strategic reflectivity. Agents are capable of reflecting on their own operations. This means that agents can be involved into the development of requirements and can offer insights and improvements. Furthermore, each agent has strategic interests in the operational world. When modelling with the I* modelling approach, these strategic interests should be reflected upon.

Rational self-interest. As last property, Yu [2001] outlines that every agent acts on self-interest and according to its own goals. This does not mean that agents act only on their self-interest, but rather that they try to fulfill their commitments to themselves and their relation to other agents in the system.

Based on these properties of agents, the I*-framework defines two different model types. The goal of these model types is to represent distributed intentionality, introduced by the agents and their properties. These two types are the so-called *Strategic Dependency Model (SD)* and the *Strategic Rationale Model (SR)* [Yu and Mylopoulos, 1994; Yu, 1995, 1997, 2001; Atkinson et al., 2015].

Strategic Dependency Model (SD). The Strategic Dependency Model aims at representing dependencies between agents in the system. A dependency concerns one agent depending on the work, input, etc. of another agent to achieve its own goals and work. Therefore, the SD-Model is able to represent internal structures of the process, the different stakeholders and their relations, and the dependency chain between the stakeholders. Based on this model type and the resulting dependency chains one can analyse vulnerabilities and opportunities for each agent in the system [Yu, 1995, 1997].

Strategic Rationale Model (SR). Whilst the Strategic Dependency Model focuses on the relationship and the dependencies between the agents, the Strategic Rationale Model focuses on the agents and their internal goals, tasks, resources and softgoals. Therefore, the SR-Model allows to analyse the internal interests of stakeholders and provides a way of exploring alternative approaches, since the SR-Model shows different ways how stakeholders can achieve their goals [Yu, 1995, 1997].

In summary, Yu and Mylopoulos [1994]; Yu [2001] see advantages in using an agent and goal-based approach in requirements engineering. This approach is especially beneficial when developing software processes, which normally have to tackle a high-complexity field and rely on collaborative problem-solving [Yu, 1997].

This thesis builds on I* as a building block to allow an elicitation of the high-scope requirements, as shown in Chapter 4 by Figure 4.1. Furthermore, I* allows to gain a better overview over the goals of each stakeholder taking part in the conducted use cases.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2.4.2 Modelling with the Integration Definition 0 (IDEF0) Standard

Integration Definition 0 (IDEF0) [ISO/IEC/IEEE 31320-1:2012] is a modelling method, focused on the structured representation of functional flows in the system. Integration Definition 0 (IDEF0) is therefore a *function* model, based mainly on the *Structured Analysis (SA)* Language, introduced by Ross [1977].

Structured Analysis (SA) Language. Ross [1977] motivates the introduction of the SA Language with the need for concise, simple and clear communication when conveying ideas in software engineering. SA is based on the principle of *"Everything worth saying about anything worth saying something about must be expressed in six or fewer pieces."* [Ross, 1977]

Following this commitment to a simple, structured language, the SA Language uses a simple building block, the so-called *SA box*. For visual representation of the main building block, see Figure 2.3. The SA box is surrounded by arrows, representing the **Input**, **Output**, **Control** and **Mechanism**. Ross [1977] sees the SA box as a transformation function that transforms the *Input* according to the *Control* imposing constraints or restrictions, and taking *Mechanisms* containing resources to complete the transformation. The result of this transformation is the *Output*. This *Output* can be an Input to other SA boxes. This property allows interaction and chaining of multiple functional steps [Ross, 1977].

Ross [1977] sees applications of the SA Language beyond requirements engineering. Since the end result is a structured model, the SA Language can be used in any field that requires conveying and presenting new ideas in a structured manner. Furthermore, the SA Language can be used by a single person or a group to approach a problem or idea in a structured way.

Limitations of the SA Language concern the loss of detailed representation of information. The SA Language allows a good representation of the high scope view on the system and flow, while minor details get lost. To represent details, other approaches like RUCM can be used.

Evolution of the SA Language into IDEF0. The Integrated Computer Aided Manufacturing (ICAM) Program of the U.S. Air Force identified the need to improve productivity in its manufacturing process and aimed to achieve this by improving analysis and communication techniques for the involved engineers. This need introduces the IDEF (ICAM Definition) family of modelling techniques. The IDEF family contains three modelling methodologies, IDEF0 - a function model, IDEF1 - an information model and IDEF2 - a dynamics model. All three model types allow to describe the environment of the modelled system [Mayer, 1990].

The IDEF0 modelling technique, proposed by the ICAM program, relies mainly on the work on SA Language by Ross [1977]. Ross was also involved in the development of the IDEF0 modelling technique [Mayer, 1990].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

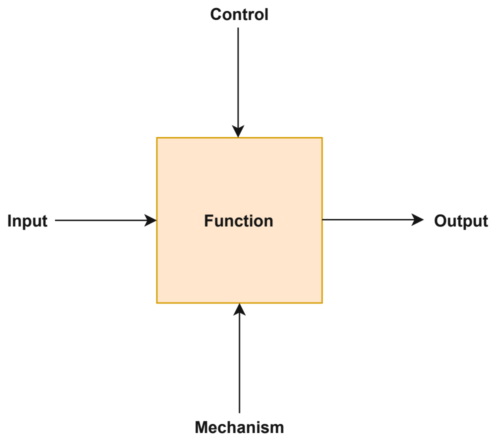


Figure 2.3: Main building block of IDEF0, the *SA box*, based on Ross [1977].

In 1993, effort started to standardize IDEF0 in the ISO/IEC/IEEE Standard. IDEF0 is contained in the ISO/IEEE standard since 1998, when it was introduced by *IEEE Std 1320.1-1998*. The current valid standard can be found in ISO/IEC/IEEE 31320-1:2012.

This thesis builds on IDEF0 for representing the high scope process flows in the use cases in Chapter 4 and Chapter 6. As stated above, to provide a more detailed definition of the use case steps, this thesis uses RUCM to define more fine-grained steps. The following section introduces this approach in more detail.

2.4.3 Restricted Use Case Modeling (RUCM)

Whilst IDEF0 focuses on an abstract, high level view on the system, mostly focused on flows between different steps, Jacobson [1993] proposes a different approach. He proposed, to structure and elicit requirements in a system, a use case driven approach Jacobson [1993]. A use case can be understood as a description of all interactions that a user has with the system.

According to Jacobson [2004], the use case driven modelling approach has been adopted quickly and is widely used in the industry. This is also reported by Dobing and Parsons [2006], where practitioners saw the value of use case diagrams especially in fields, where *"verifying and validating requirements with client representatives on the project team"* [Dobing and Parsons, 2006] is of special importance.

Yue et al. [2009] raised the issue of ambiguity in the use case driven approach due to the predominant textual definition of the use cases. This unstructured textual definition allows a lot of flexibility, but also leads to ambiguity between use cases, since they do not comply to the same structure or rules. To address this issue, Yue et al. [2009] introduce the *Restricted Use Case Modeling (RUCM)* template, together with restrictions, to define use cases. The goal is to reduce ambiguity and provide a more structured approach to model use cases.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	The name of the use case
Brief Description	Summarizes the use case in a short paragraph
Precondition	What should be true before the execution of the use case.
Primary Actor	The actor which initiates the use case.
Secondary Actors	Other actors the system relies on to accomplish the services of the use case.
Dependency	Include and extend relationships to other use cases.
Generalization	Generalization relationships to other use cases.
Basic Flow	Specifies the main successful path, also called "happy path". Also contains numbered steps and the postcondition after the flow.
Specific Alternative Flow	Applies to one specific step of the reference flow. RFS: Reference flow step, where the flow branches from. Also contains numbered steps and the post-condition after the flow.
Global Alternative Flows	Applies to all the steps of the reference flow. Also contains numbered steps and the postcondition after the flow.
Bounded Alternative Flows	Applies to more than one step of the reference flow, but not all of them. RFS: Reference flow step, where the flow branches from. Also contains numbered steps and the postcondition after the flow.

Table 2.2: RUCM use case template, based on Yue et al. [2009]

Table 2.2 illustrates a RUCM template. A filled out template can be found in Table 4.1, defining use case 1 of this thesis. Most steps are self-explanatory and provide the user with a rigid and structured approach of the definition of the use case.

Furthermore, the RUCM template contains four flows that describe the steps needed to conduct the use case. First, the *Basic Flow* contains all steps to define the main successful path of the use case without branching, e.g., a flow to successfully retrieve money from the ATM can be represented by the *Basic Flow*.

Second, the *Alternative Path* has three subcategories that define all steps of alternative scenarios, both success and failure [Yue et al., 2009]. The *Specific Alternative Flow* refers to a single step of the Basic Flow, e.g., representing an Alternative Flow if the PIN to retrieve money at the ATM is wrong. The *Global Alternative Flow*, an Alternative Flow referencing any step in the Basic Flow, e.g., by finding another ATM if the first ATM fails at any step in the Basic Flow. The *Bounded Alternative Flow* refers to multiple steps of the Basic Flow, e.g., defining an Alternative Flow if the credit-card cannot be read by the ATM in the first three steps.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

To reduce ambiguities, RUCM imposes constraints on defining use cases. These restrictions concern either the use of natural language, e.g., using only present tense when defining the Basic Flow, or to enforce the use of special keywords, e.g., using the term “the system” when talking about the system under design. A full listing of these Restrictions can be found in Table 2 of Yue et al. [2009]. The restrictions most relevant for this thesis are:

- R2 - *“‘Describe the flow of events sequentially.”* [Yue et al., 2009]. This aids the readability and traceability for readers of this thesis.
- R4 - *“Describe one action per sentence. (Avoid compound predicates.)‘”* [Yue et al., 2009]. Aims at reducing complexity by separating multiple actions into different sentences.
- R7 - *“Clearly describe the interaction between the system and actors without omitting its sender and receiver.”* [Yue et al., 2009]. This is especially relevant for this thesis, due to the multi-disciplinary context of the use cases. This restriction enforces the explicit definition of interactions between the system and actors.

In summary, the RUCM approach provides a construct to allow a more stringent definition of use cases and aids users in the correctness of their use cases. As outlined and evaluated by Yue et al. [2009], the usage of RUCM leads to a higher correctness when deducting class-diagrams with RUCM.

This thesis uses RUCM for the definition of the flows and steps for each specific use case, as can be found in Chapter 4. RUCM allows the reader to clearly follow each step conducted in the use cases and possible alternative flows. Thus, RUCM aids traceability and verifiably, improving the soundness of the use case representation.

2.5 Stigmergy and Coordination

The multi-disciplinary nature of CPPS Engineering requires approaches for collaboration and coordination, especially in an asynchronous context. Therefore, this section introduces the terms of Stigmergy and CIS. The terms of Stigmergy and CIS are introduced to (1) outline an existing approach that leverages asynchronous communication via traces in the environment and (2) to outline an existing approach that leverages stigmergic concepts in existing software platforms.

This section introduces the term *Stigmergy* by outlining its foundation in the biological field and its relation to the field of engineering. It also introduces resulting methodologies and approaches based on the approach of stigmergic behaviour. This stigmergic approach enforces collaboration, and coordination between engineering disciplines. This section therefore also introduces approaches on coordination via CIS and coordination artefacts in the environment, which will be used in Chapter 5.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2.5.1 Introduction to Stigmergy

To allow and guide the coordination between multiple engineers across multiple engineering disciplines, this thesis requires a method allowing asynchronous communication between engineers. This allows asynchronous working by multiple engineers on MAI tasks. Therefore, the approach provided by Stigmergy is introduced.

Stigmergy was first formally defined by the French biologist and zoologist Grassé [1959]. In his work, he analysed the behaviour of termites as an organized social structure. The goal of his work was to explain how an insect society, like termites, can behave in a coordinated way while each individual seems to work alone and take decisions for itself.

The term Stigmergy, from the Greek word *stigma* (engl. sting) and *ergon* (engl. work), describes in general mechanisms that enable indirect task coordination and regulation [Theraulaz and Bonabeau, 1999]. It was initially introduced in the context of coordination of termites in the work of Grassé [1959]. A simple example for this are pheromones that ants mark in the environment to communicate, e.g., routes to food sources.

As Theraulaz and Bonabeau [1999] outline, the main principle of Stigmergy is very simple. Individuals leave traces in the environment, whilst other agents feed back on them and react to these traces. These traces allow an organized behaviour in between agents, without direct communication. Therefore, each agent seems to behave independently in the system, whilst really feeding off of traces left in the environment. As Theraulaz and Bonabeau [1999] further sketch, this feeding off of traces in the environment allows a feedback loop in between the agents.

This behaviour can be seen in ants that select new food sources near the hive. If the path between the hive and the food source is too long, the traces in the environment, e.g., pheromones, evaporate. This leads to a disappearance of the trail, since no traces in the environment can be found by other ants. It also leads to a positive feedback loop, where ants select closer food sources since they frequent the trail more often and the ants refresh traces in the environment more often. This makes the trail better “visible” to other ants of the colony.

In general, two main types of Stigmergy can be distinguished [Theraulaz and Bonabeau, 1999]:

Quantitative Stigmergy. In quantitative Stigmergy agents react to a quantitative stimulus. This means that the agent reacts to an accumulation of, e.g., soil mixed with pheromones. This is used as an example in Grassé [1959], which outlines the process of building pillars by termites. In the beginning, termites accumulate soil mixed with pheromones in an uncoordinated behaviour. This accumulation of soil and pheromones, triggers a positive feedback loop, in which more and more termites accumulate soil. When enough workers are available, a coordinated effort happens in between the termites, since enough pheromones are available. In this example, the positive feedback loop due to the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

	Quantitative Stigmergy	Qualitative Stigmergy
Marker-based	Path following of ants via pheromones	Combination of pheromones leads to decisions
Sematectonic	Building of pillars by termites	Construction of wasp nests

Table 2.3: Distinctions of different Stigmergy types, based on Parunak [2005]

traces in the environment is visible, since it incentivizes other termites to also accumulate more soil.

Qualitative Stigmergy. In contrast to quantitative Stigmergy, qualitative Stigmergy is based on qualitative stimuli of the environment. For example, a wasp decides what nest structure to build according to the nest form around it, therefore a qualitative aspect [Parunak, 2005].

Theraulaz and Bonabeau [1999] more generally outlines that in qualitative Stigmergy the Stigmergy types consist of a well-defined set. This means that each specific Stigmergy stimulus leads to a specific action by the agent.

Parunak [2005] further differentiates between sematectonic stigmergy and marker-based Stigmergy in a human-human based setting, as two subtypes to quantitative Stigmergy and qualitative Stigmergy. This can be seen in Table 2.3. Sematectonic Stigmergy refers to Stigmergy, where actions base on the current solution state. In comparison, marker-based Stigmergy references Stigmergy where agents communicate indirectly via markers and traces placed in the environment.

This thesis builds on Stigmergy as a building block in Chapter 5. The aim is to improve cooperation, coordination, and the quality of the results of CMAI method by using Stigmergy. This is used in this thesis in Chapter 5. It also will be used to answer RQ2 and RQ3 in Section 3.2 as a possible improvement. The solution approach will focus mainly on marker-based Stigmergy in a qualitative context. This is due to the domain experts involved that want to use indirect communication qualitatively and are looking to analyse and improve models instead of communication via a solution state as is needed in sematectonic Stigmergy. Furthermore, Parunak [2005] outlines that Stigmergy is a good fit in structured environments, e.g., graphs. This makes it a good fit in the context of this thesis, where the goal is to improve Model Analysis and Improvement.

2.5.2 Collective Intelligence Systems (CIS)

In this thesis, the proposed solution approach allows interaction between the engineers via Stigmergy, as outlined above. This results in collected information in the platform that is added by each actor in the system. To handle this aggregation and dissemination of information and to improve the feedback loop between the engineers and the system, the term of Collective Intelligence Systems (CIS) is introduced.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Musil et al. [2015] reviewed and analysed CIS to provide a framework on how to systematically build a CIS architecture. A CIS is a sociotechnical system aimed at collaborative working between humans. Examples for CIS are wikis in the public and corporate context, such as Wikipedia[4] and Confluence[5]. The goal of these platforms is to collaboratively aggregate information on topics and make them available to other users. Musil et al. [2015] also outlines that approaches exploiting stigmergic concepts in CIS are emerging.

This influence by Stigmergy, outlined by Musil et al. [2015], is due to the indirect communication between actors that is enforced by CIS. This stigmergic approach in CIS can be separated in two phases. These are the *aggregation* phase and *dissemination* phase. The aggregation phase is the phase in which actors interact with the artefacts of the CIS. In the dissemination phase, the CIS makes agents aware of new changes and activity in the system. This interaction of aggregation and dissemination leads to a perpetual cycle and a positive feedback loop, as can be found in Stigmergy [Musil et al., 2015].

An example of this is the modification of an existing Wikipedia[6] article. Here, users can collaboratively add changes to an article, to improve its overall quality and include new information. This is the aggregation phase. The homepage then makes these changes visible to other users. This is the dissemination phase. Based on these changes, other users can then review, improve and further modify the article. As can be seen, this leads to a positive feedback loop as present in Stigmergy, since users are aware of changes, improve on them while also creating new changes, leading to a perpetual cycle.

This work builds on the main principles of CIS in which multiple different actors work collaboratively to aggregate and improve knowledge, e.g., engineering models. It also uses the approach of Stigmergy in CIS to improve the overall result quality and cooperation between engineers.

2.5.3 Collaboration in Multi-Disciplinary Engineering

As Musil et al. [2016] outline, collaboration between multiple different engineering disciplines and actors is an important factor in CPPS engineering. Major issues in this context arise from dispersed information between engineering disciplines. Each actor has local engineering knowledge that is difficult to share due to heterogeneous communication channels. This also leads to poor discoverability of information containing critical know-how to the business.

To address these coordination and knowledge sharing issues, Musil et al. [2016] suggests using CIS-approaches in the context of CPPS.

The directions in which CIS intermeshed with CPPS engineering could be used are outlined by Musil et al. [2017]. Musil et al. [2017] sees potential in capability augmentation, machine-to-machine interaction and multi-disciplinary knowledge integration and coordination.

[4]`https://www.wikipedia.org/` Last accessed: 14/05/2021
[5]`https://www.atlassian.com/de/software/confluence` Last accessed: 14/05/2021
[6]`https://www.wikipedia.org/` Last accessed: 14/05/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

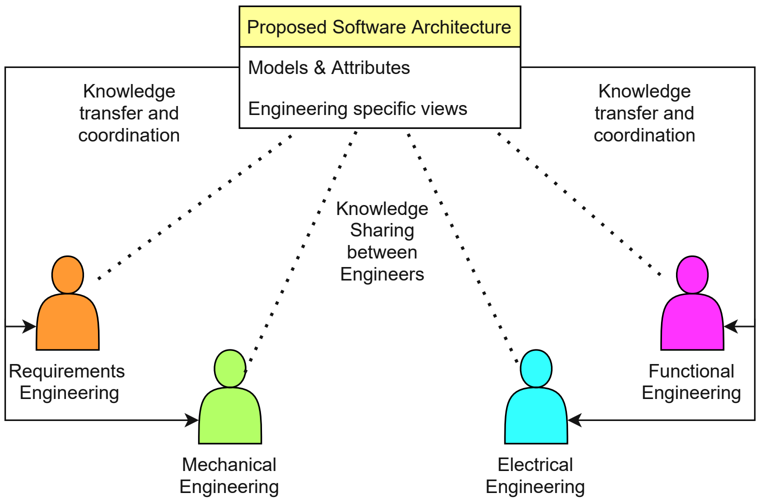

Proposed Software Architecture
Models & Attributes
Engineering specific views
Knowledge transfer and coordination
Knowledge transfer and coordination
Knowledge Sharing between Engineers
Requirements Engineering
Mechanical Engineering
Electrical Engineering
Functional Engineering

Figure 2.4: Knowledge sharing and coordination between engineering disciplines using a software platform, based on Musil et al. [2017].

This thesis makes use of the Coordination aspects of a CIS in the light of improving knowledge sharing and overall quality of artefacts, as outlined in Musil et al. [2017]. A visual representation of this can be seen in Figure 2.4. As can be seen, each engineering discipline adds new knowledge to the platform, which is then transferred and shared back to the engineers via the system.

2.5.4 Environment and Coordination artefacts

Weyns et al. [2007] introduces the environment of multi-agent systems as an important part of the system and as key to provide solid coordination support between agents. Therefore, Weyns et al. [2007] introduced the environment as first-class abstraction in multi-agent systems. The environment has two critical roles. First, it is the space with which agents interact, record their results and evaluate their doings. Secondly, it allows a separation of concerns. It separates the environment strictly from the agents in the systems. This also aids the reduction of complexity when acting with multi-agent systems [Weyns et al., 2007].

Weyns et al. [2007] also outlines that the environment can be used as a flexible coordination medium. Due to the virtual nature of environments in multi-agent systems, it creates further opportunities for agents to communicate, beyond the real world.

This possibility to use the environment as a coordination medium is an idea shared by Omicini et al. [2004]. Omicini et al. [2004] defines coordination artefacts to allow coordination between agents. For Omicini et al. [2004] coordination artefacts must have three main characteristics.

- *Operations.* Actions possible by the agent on the coordination artifact.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- *Operating instructions.* Explicit definition on how to use the coordination artefacts.
- *Coordination behaviour specification.* Explicit definition of the coordinating behaviour of the coordination artifact.

For Omicini et al. [2004] these characteristics can be exploited by intelligent agents and allow even complex coordination scenarios between agents.

This work will build on the state of the art on environments and coordination artefacts to exploit the virtual environment of the proposed software architecture and aid coordination between agents with coordination artefacts. This work will build on stigmergic concepts to aid asynchronous communication between engineers on graphs. Furthermore, the solution design leverages the concepts introduced by CIS to improve knowledge transfer and coordination between engineers. The characteristics outlined for coordination artefacts will be taken into account when designing the solution approach to marker-based stigmergy. This aims at filling the gap in the research regarding insufficient tool support for coordination and collaboration with markers, with a focus on the field of CPPS Engineering.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 3

Research Questions and Approach

The goal of the thesis is to design and evaluate a novel approach for coordination in modelling for Cyber-Physical Production Systems (CPPS) Engineering by adapting established models, methods, and mechanisms from informatics by following the design science approach [Hevner et al., 2004]. The design science approach will be used when designing the Solution Approach proposed in Chapter 5.

3.1 Design Science

Hevner et al. [2004] outlines that engineers develop information systems to improve the workflow and efficiency between users in an organization. To aid the design science research process, Hevner et al. [2004] provide seven guidelines in the research process to assist researchers, reviewers, editors, and readers in the design science research process.

To further aid understanding of the problems and solutions proposed in this thesis, a visual abstract, based on Engström et al. [2020], is introduced by Figure 3.1. The Figure shows the problems, the solutions, the relevance, the rigor, and the novelty of this thesis and provides a quick overview over the thesis and its results.

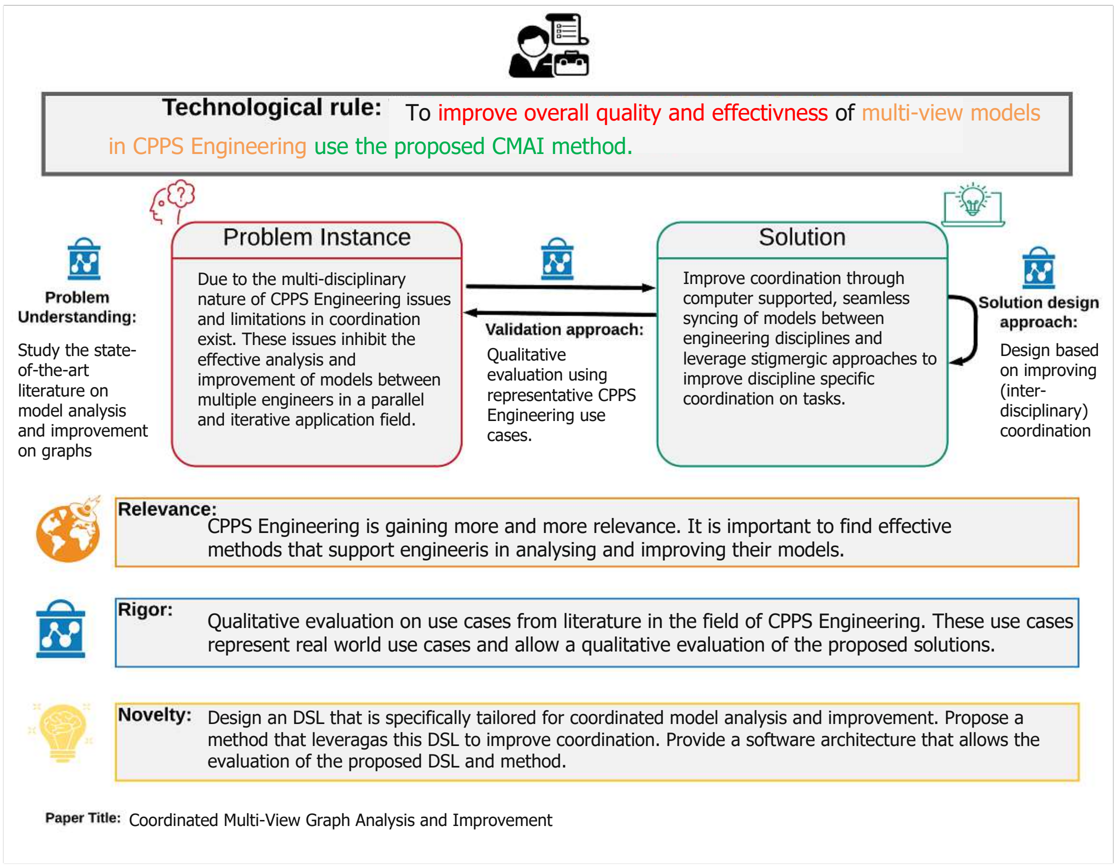
This thesis will build on the guidelines of Hevner et al. [2004] as follows.

Design as an artefact. This guideline will be addressed by the Domain-Specific Language (DSL) and the artefacts needed to answer the research questions defined in Section 3.2.

Problem relevance. The problem relevance arises from the state of the art and the real world problems. Furthermore, this thesis takes challenges from the literature into account to fulfil this guideline. The current state of the art and arising challenges from it has been outlined in Chapter 2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek, Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



Technological rule: To improve overall quality and effectivness of multi-view models in CPPS Engineering use the proposed CMAI method.

Problem Understanding: Study the state-of-the-art literature on model analysis and improvement on graphs

Problem Instance: Due to the multi-disciplinary nature of CPPS Engineering issues and limitations in coordination exist. These issues inhibit the effective analysis and improvement of models between multiple engineers in a parallel and iterative application field.

Validation approach: Qualitative evaluation using representative CPPS Engineering use cases.

Solution: Improve coordination through computer supported, seamless syncing of models between engineering disciplines and leverage stigmergic approaches to improve discipline specific coordination on tasks.

Solution design approach: Design based on improving (inter-disciplinary) coordination

Relevance: CPPS Engineering is gaining more and more relevance. It is important to find effective methods that support engineeris in analysing and improving their models.

Rigor: Qualitative evaluation on use cases from literature in the field of CPPS Engineering. These use cases represent real world use cases and allow a qualitative evaluation of the proposed solutions.

Novelty: Design an DSL that is specifically tailored for coordinated model analysis and improvement. Propose a method that leveragas this DSL to improve coordination. Provide a software architecture that allows the evaluation of the proposed DSL and method.

Paper Title: Coordinated Multi-View Graph Analysis and Improvement

Figure 3.1: Visual Abstract, based on [Engström et al., 2020].
Template used from online repository[1].

Design evaluation. To evaluate the proposed solutions, the use cases introduced in Chapter 4 will be evaluated by the solution approach proposed in Chapter 5. Chapter 6 conducts the evaluation of the proposed design and Chapter 7 discusses the results. This qualitative evaluation approach allows to evaluate the proposed design, which is integral to the design science process as proposed by Hevner et al. [2004].

Research contributions. Since the research questions in Section 3.2 go beyond the state of the art introduced in Chapter 2, this thesis makes new contributions to the research field. This way, this thesis creates new knowledge and contributes to the research field.

Research rigor. This thesis introduces the design and evaluation approach in great detail in this thesis. This allows the readers to easily verify that this work uses sound design and evaluation methods. Due to this, this thesis fulfils the need for research rigor

[1]https://raw.githubusercontent.com/margaretstorey/VASE/master/Template.png Last accessed: 02/09/2021.

in the design science process.

Design as a search process. To allow for a good design and solution, this thesis will analyse the state of the art, elicit relevant use cases to the problem domain in Chapter 4 and propose and appropriate solution design in Chapter 5. This step wise approach allows searching for an optimal solution, relying on the elicited problems and use cases.

Communication of research. This master thesis will be made available to researchers and the public via the library of TU Wien, effectively communicating the research results.

3.2 Research Questions

This thesis aims to answer the following research questions. Chapter 2 introduced the building blocks and research gaps that these research questions base on.

RQ1 is motivated by the existing need in the State of the Art to define a DSL, allowing for coordinated model design and improvement approaches in the field of CPPS Engineering. RQ2 leverages the defined DSL, introduced by RQ1, to define a method that allows users to conduct Coordinated Model Analysis and Improvement. This is founded in the need in the State of the Art to provide a method allowing for coordination between users, while also leveraging stigmergic concepts. RQ3 is concerned with providing a software architecture, allowing the manipulation of models defined in the DSL of RQ1 and aiding the conduction of the Coordinated Model Analysis and Improvement (CMAI) method proposed by RQ2.

RQ1. CMAI-DSL. What domain-specific language (DSL) allows representing and linking of multi-view graphs for coordinated multi-disciplinary engineering? Coordinated engineering requires both the visual and data representation of a multi-view graph of discipline-specific engineering models and markers to represent engineering tasks and their solution states.

To address RQ1, this thesis plans to design and evaluate a DSL for Coordinated Model Analysis and Improvement (CMAI), the CMAI-DSL, which allows flexible representation of engineering models from several engineering disciplines, coordination artefacts - such as markers - and of the links between these models. These links allow for the propagation of multi-view model changes that are the result of a resolution of a specific task, such as improving the design of a model part, modelled in functional, mechanical, and electric views. Further, markers will provide the foundation for defining and using coordination artefacts representing tasks and their solution states, both for humans and computers.

RQ2. CMAI method. What method can domain experts use to define *design and validation loops* to analyse and improve multi-view graphs, with links between models in different engineering disciplines? Domain experts require a method to define a *design and validation loop* within a multi-view model. To achieve

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

this, this thesis proposes a method allowing for coordination and model analysis and improvement, the CMAI method.

As a starting state, the CMAI method starts from related models in a multi-view context, needing design and validation by domain experts. To aid this interdisciplinary design and validation, the CMAI method leverages the links between models, introduced by RQ1, to automatically propagate changes in multi-view models. This allows domain-experts to conduct design and validation loops automatically on the most recent data.

Furthermore, to aid discipline specific coordination in a design and validation loop, the CMAI method leverages the coordination artefacts introduced by RQ1. This allows domain experts to track tasks and resolution states in models, leveraging stigmergic concepts.

The goal of RQ2 and the CMAI method is to provide domain experts with a framework, allowing for design and validation loops in the context of multi-view models.

RQ3. CMAI Editor and Service Architecture. What software architecture and solution design can provide the capabilities to efficiently analyse and improve models? Domain experts require a software architecture to conduct a *design and validation loop* within a multi-view model. For this, this thesis designs the Generic CPPS Modeling (GCM) Editor.

This software architecture needs to allow the storage and visual representation of the resulting CMAI-DSL from RQ1. Furthermore, the architecture needs to provide an efficient way for users to manipulate models represented in the CMAI-DSL.

To provide the capabilities to efficiently analyse and improve models, - using the CMAI method introduced in RQ2 - the GCM Editor must allow the definition of links between models, including the capability of interdisciplinary property propagation. Furthermore, it also needs to allow users to use the coordination artefacts introduced by RQ1 and used by the CMAI Method.

3.3 Work Packages

Figure 3.2, based on Figure 1.2 in Chapter 1, shows a refined summary on the work and research results of this thesis.

WP1. Elicit Use Cases and Survey Literature and Technology. In WP1, this thesis elicits use cases, relevant for researchers and practitioners, resulting in *Use Case Descriptions*. Chapter 4 outlines these use case descriptions.

This thesis surveys literature in the field of CPPS Engineering, Coordination and Multi View model analysis and improvement, resulting in *Scientific gaps that motivate the research.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

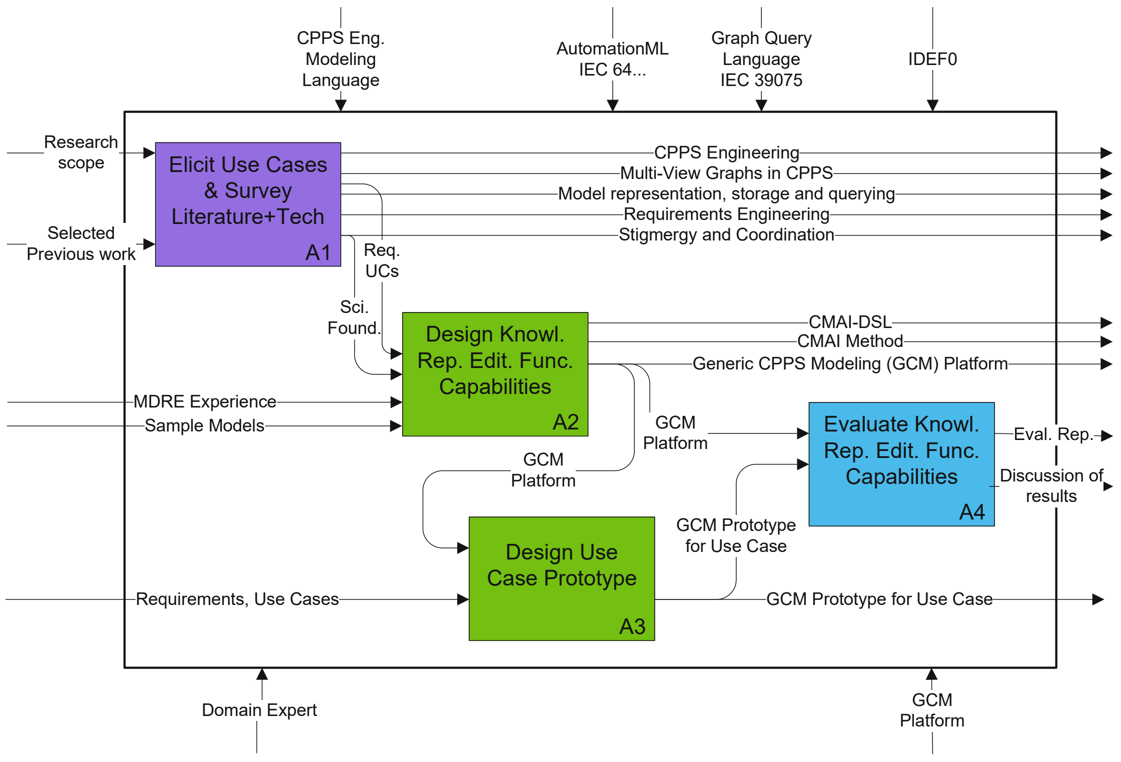

CPPS Eng. Modeling Language
AutomationML IEC 64...
Graph Query Language IEC 39075
IDEF0
Research scope
Selected Previous work
Elicit Use Cases & Survey Literature+Tech
A1
CPPS Engineering
Multi-View Graphs in CPPS
Model representation, storage and querying
Requirements Engineering
Stigmergy and Coordination
Req. UCs
Sci. Found.
Design Knowl. Rep. Edit. Func. Capabilities
A2
CMAI-DSL
CMAI Method
Generic CPPS Modeling (GCM) Platform
MDRE Experience
Sample Models
GCM Platform
GCM Platform
Evaluate Knowl. Rep. Edit. Func. Capabilities
A4
Eval. Rep.
Discussion of results
GCM Prototype for Use Case
Design Use Case Prototype
A3
Requirements, Use Cases
GCM Prototype for Use Case
Domain Expert
GCM Platform

Figure 3.2: Refined work packages of this thesis, based on Figure 1.2.

This thesis surveys technology on model storing, traversal and querying resulting in building blocks for the back end of the Model Analysis and Improvement (MAI) *system architecture*. Chapter 5 introduces these building blocks.

WP2. Design Knowledge Representation and GCM Editor Capabilities. In WP2, this thesis designs the CMAI-DSL to allow for knowledge representation in the field of CPPS Engineering. Furthermore, this thesis designs a *system architecture* based on the requirements for MAI editing capabilities. Section 5.3 outlines this system architecture and its technical aspects. Based on this architecture design, this thesis designs and implement a GCM platform.

WP3. Design Use Case Prototype. In WP3, this thesis designs a *GCM prototype* tailored to the use case, building on the GCM Platform coming from WP2 and the technical architecture outlined in Section 5.3.

WP4. Evaluate Knowledge Representation and GCM Editor Capabilities. WP4 aims at evaluating the results coming from the other work packages and the capabilities of the *GCM platform*, resulting in an evaluation report. Chapter 6 conducts the evaluation and Chapter 7 discusses the results.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.4 Requirements

Due to the current shortcomings in the real world and difficulties motivated in Chapter 1 and outlined in the State of the Art in Chapter 2, the following high scope requirements for the solution approach proposed in Chapter 5 arise. The requirements concern data representation, method design, and model storage and querying. Chapter 4 cross-references these requirements for each use case. Furthermore, Chapter 4 defines more detailed requirements for each conducted use case.

These requirements come from the related literature, outlined in Chapter 2. This thesis elicited these requirements together with researchers and domain experts, also using the use cases outlined in Chapter 4 and using the exemplary use case used in Chapter 1. Overall, the requirements outlined here give a high-scoped overview over the elicited requirements. For more detailed requirements per use case, relate to Chapter 4.

R1: Representation of domain knowledge. *CMAI-Domain-Specific Language (DSL).* The use cases introduced in Chapter 4 require the representation of domain concepts, such as technical assets and links, to model production systems in the CPPS Engineering context. These domain concepts can be described in a DSL. This DSL must allow for *Coordinated Model Analysis and Improvement (CMAI)*, therefore it will be named CMAI-DSL.

CPPS Engineering Network (CEN) representation. The CMAI-DSL must allow for the representation of graphs consisting of nodes and edges and their respective properties, e.g., *name*, *type*, and *description.* These nodes and edges need to allow for a representation of nodes and edges of different types and type-specific properties. Furthermore, the CMAI-DSL must allow the representation of CPPS Engineering Networks (CENs) and multi-view models. This means that models must have the capability to be linked to each other directionally. Via these model-to-model links, a propagation of properties for model elements is possible

Coordination markers. To allow for CMAI it is necessary that the proposed CMAI-DSL allows for the representation of markers as coordination artefacts. These coordination markers aim at leveraging stigmergic methods on models, improving coordination in parallel CPPS Engineering. Furthermore, these coordination markers can be used for model design and model improvement, tracking the solution states of tasks in the model.

The representation of markers should not only be visual, it should also be possible to define a finite set of states that a marker can take on. The markers allow tracking and coordination of the status of nodes and edges in the CMAI method for better coordination between domain experts and the computer. This allows human to human interaction, but also human to machine interaction.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

R2: Method for interactive Model analysis and improvement. *Task-oriented visual representation and easy manipulation of the models.* This capability concerns browsing the model, representation of properties on nodes and edges, querying of models and annotating models and model elements with markers to allow for coordination. This requires that users can intuitively manipulate the existing model and easily augment the model.

Furthermore, providing a method for annotation of the models and model elements, allows users to track design and improvement lifecycle states on models via these coordination markers

Coordination between users in CENs and the CMAI method. This capability concerns notifying users of changes to a CEN to create awareness of ongoing changes and CMAI activities in the team. This can, for example, be achieved by listing updates made to the model by the system or other engineers.

Web architecture. The user interface should be available online without local installation of software. The architecture should also take into consideration further connectivity to external systems, e.g., a ticketing system or a human computation system. This requirement aims at more easily introducing the GCM Editor into an existing (software) ecosystem.

R3: Data storage and querying methods to support Model analysis and improvement. The back-end of the proposed software architecture requires capabilities for storing the graphs and for computational tasks regarding the models.

Model transformation and selection. Sequential collaboration between domain experts capabilities for the import and export of models from the internal data representation to common external formats, e.g., JavaScript Object Notation (JSON), including data transformation to DSLs, e.g., Automation Markup Language. This would allow treating subsets of a bigger model, represented in Automation Markup Language (AML), both in the MAI front-end editor and MAI back-end. This subset may be a set of nodes and edges to review as a CMAI task.

Model configuration and storage. A persistence capability is to store and query the model configurations and the concrete model instances. To allow an effective and responsive representation, manipulation, and querying of big and complex models in the MAI editor, the storage format should be capable of handling the typical use-case.

3.5 Evaluation Concept

This section introduces the evaluation approach as a foundation for the evaluation reported in Chapter 6. First, this section outlines the general approach used for the evaluation. Secondly, this section refers to the online resources used in the evaluation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

This aims at providing the reader with the appropriate background knowledge to verify the evaluation.

3.5.1 Technical feasibility study

The evaluation approach consists of a technical feasibility study. For this, this work conducts the illustrative use cases introduced in Chapter 4 using the software prototype introduced in the solution approach in Chapter 5. After conducting the use cases, this thesis checks the fulfilment of the introduced requirements for each use case. Chapter 6 outlines and discusses the strengths and weaknesses for the proposed solution approach

The use cases used for the evaluation will be defined in the context of multi-disciplinary multi-view models. Due to the nature of multi-disciplinary models, the models are fairly big and complex in a real world CPPS Engineering application. To analyse and improve the models a goal will be defined, e.g., applying changes to an existing production system, such as a solar energy park or improving the production line of a drive-chain. There are two relevant steps on how this can be achieved, represented by the illustrative use cases used for evaluation in this thesis:

Model Analysis. Model analysis aims at marking model elements that are relevant for the defined task. The marking of model elements can either come from an automated system or from human domain experts. Based on these marked elements, a domain expert can systematically define tasks to achieve the goal, and set further markers to define the status of model elements regarding the task goal, e.g., whether a model element requires improvement.

Model Improvement. The results of the model analysis step define goals for model improvement tasks. These model improvement tasks can be conducted by a team of domain experts specialized in the discipline of the assigned task, e.g., improving a design aspect. Depending on the state of the task, the domain experts set new markers for improved coordination in the team. After finishing the model improvement task, changes to the model elements will be iteratively checked in a model analysis step. The goal of this iterative approach is to check whether the applied changes achieved the overall goal or require further improvement.

To allow for a validation of the results of this thesis, the following section introduces online resources containing the software artefacts and tutorials.

3.5.2 Online evaluation resources

To allow for a verification of the results, this thesis points the reader to three online resources.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Software artefact. The GCM Editor, resulting from the solution approach introduced in Chapter 5, is stored in a code repository. It is separated into the GCM Editor Frontend [2] and the GCM Editor Backend [3].

Documentation and DSLs for Software artefact. The documentation for the resulting software artefact and resulting DSLs of the use cases in Chapter 4 can be found also be found online. [4]. This documentation also includes the models used in the Evaluation in this thesis in Chapter 6.

Documentation on external tools. Chapter 6 uses external tools for the evaluation. To get to know the Cypher language used by Neo4J, the reader should refer to the *Get Started* guide of Neo4J [5]. To better understand the Import/Export format used in Chapter 6 the reader can refer to RFC-4180 [6].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[2]GCM Editor Frontend Repository: `https://bitbucket.org/tuw-qse/mdre-frontend`
[3]GCM Editor Backend Repository: `https://bitbucket.org/tuw-qse/mdre-backend`
[4]Model Configuration Repository: `https://bitbucket.org/tuw-qse/mdre-thesis-cengelbrecht`
[5]`https://neo4j.com/developer/get-started/` Last accessed: 30/6/2021
[6]`https://www.ietf.org/rfc/rfc4180.txt` Last accessed: 30/6/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

Illustrative Use Cases for Evaluation

This chapter introduces four use cases to illustrate and evaluate the solution approach described in Chapter 5. Chapter 6 evaluates these conducted use cases and Chapter 7 discusses the results.

This chapter consists of two main sections. The first section introduces the context, setting and engineering setting of the use cases with *I-Star (I*)* and *Integration Definition 0 (IDEF0)* models [Mayer, 1990; Yu, 1997]. This allows to settle the use cases in possible applications in the field of Cyber-Physical Production Systems (CPPS) Engineering.

The second section introduces the use cases that concern (1) connected CPPS-Modelling, (2) Product-Process-Resource (PPR) applications, (3) PPR Asset Network (PAN) applications, and (4) marker-based coordination. For each use case, its motivation, context, and goals are provided. Therefore, this thesis builds on the Restricted Use Case Modeling (RUCM) template, which allows a step by step definition of the use case, as a foundation for defining requirements [Jacobson, 1993]. These requirements influence the solution approach proposed in Chapter 5 and the evaluation of fulfilling these requirements in Chapter 6.

The use cases provide multi-disciplinary engineering process scenarios that are conducted by engineers from multiple domains during CPPS planning. The use cases were elicited from published scientific work in the area of CPPS Engineering [Drath et al., 2020; Biffl et al., 2020; Meixner et al., 2021; Biffl et al., 2021b]. All use cases in the referenced publications were elicited together with CPPS Engineering companies, mainly in the domain of automotive manufacturing. This allows conducting use cases relevant to stakeholders in the domain of multi-disciplinary CPPS Engineering. UC1 to UC3 raise the need for coordination and cooperation concerning CPPS Engineering and in the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

concepts introduced by the State of the Art in the field of Stigmergy, outlined in Section 2.5. This need motivates the fourth use case UC4.

All introduced UCs are used for the evaluation of RQ1, since the representation of any UC requires the usage of the Coordinated Model Analysis and Improvement (CMAI)-Domain-Specific Language (DSL). The CMAI method, planned as contribution to RQ2, is built up using UC2 and UC3, and evaluated using UC4. RQ3 is evaluated by conducting every proposed UC in the software architecture that will be proposed in Chapter 5, thereby showing the fulfilment of the requirements posed onto the software architecture.

4.1 Use Case Context

This section aims to introduce the setting of the use cases in CPPS Engineering. The goal of this section is to outline the dependencies between stakeholders involved in the engineering process and provide a high-level view on the information flows between actors. These generalized views onto the use cases abstract main concepts and ideas. Therefore, the I* and IDEF0 model do not represent each use case specifically, but provide an introduction and overview over the challenges and resulting requirements in multi-disciplinary engineering. These generalized I* and IDEF0 views concern all introduced use cases.

For the description of the models, selected stakeholders were chosen that take part in a CPPS Engineering Process. This subset was taken from the State of the Art in Chapter 2, e.g., Biffl et al. [2017]. A minimum subset of stakeholders has been chosen for these use cases, since a CPPS Engineering Process consists of at least 8 to 10 engineering views [Biffl et al., 2020]. Since the research questions introduced in Section 3.2 aim at coordinated analysis and improvement of models in CPPS Engineering, representative engineering disciplines, like Requirements Engineering or Electrical Engineering, from this field were chosen.

4.1.1 Strategic stakeholder goal modelling I*

To conduct a high scope elicitation of the dependencies between actors in the conducted use cases, this thesis uses the I* modelling approach. This allows to gain insights on internal goals and requirements of actors in the context of CPPS Engineering.

Figure 4.1 represents the modelled actors and their dependencies. In the identified use cases, four types of engineers are involved. These engineers are the Requirements Engineer, the Functional Engineer, the Mechanical Engineer and the Electrical Engineer. Due to this multi-disciplinary aspect, engineers need to exchange data of their models in a cross-disciplinary manner, to work on the same underlying data. Figure 4.1 models this required data exchange by a softgoal.

Furthermore, the engineers need to be provided with data representing the building blocks for CPPS Engineering models. The Data Curator takes on this task in providing this data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

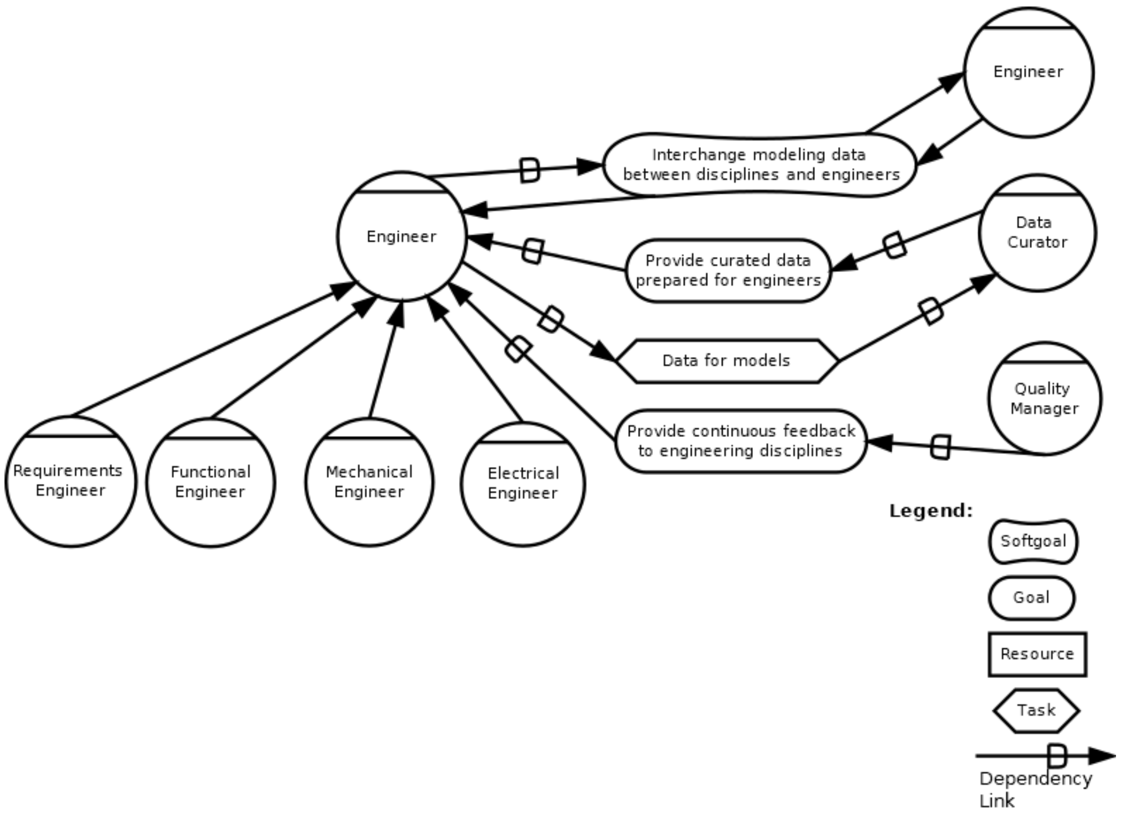


Figure 4.1: Dependencies between actors in the conducted use cases, modelled using the I* framework [Yu, 1997].

Lastly, a Quality Manager is interested in providing feedback to the engineers to improve the overall quality of the CPPS Engineering models. Figure 4.1 also depicts this aspect.

The I* approach provides a good overview over each actor's goals in the context of multi-disciplinary, multi-view graphs. However, I* has its limitation when depicting the flow of information and steps taken by each actor. Therefore, Figure 4.2 shows the data flows of the use cases modelled in IDEF0 notation.

4.1.2 Process overview with IDEF0 diagram

Figure 4.2 shows an overview over the data flows in the conducted use cases. They different parts of the model are explained in the following paragraphs.

The process step *A0*, see top left part of the diagram, serves as entry point into the IDEF0 diagram. In this process step, the requirements for the domain-specific CPPS engineering model (of a particular stakeholder) and its unstructured data are provided. As output, this process step provides the *curated data*, usable for CPPS modelling by each engineer.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

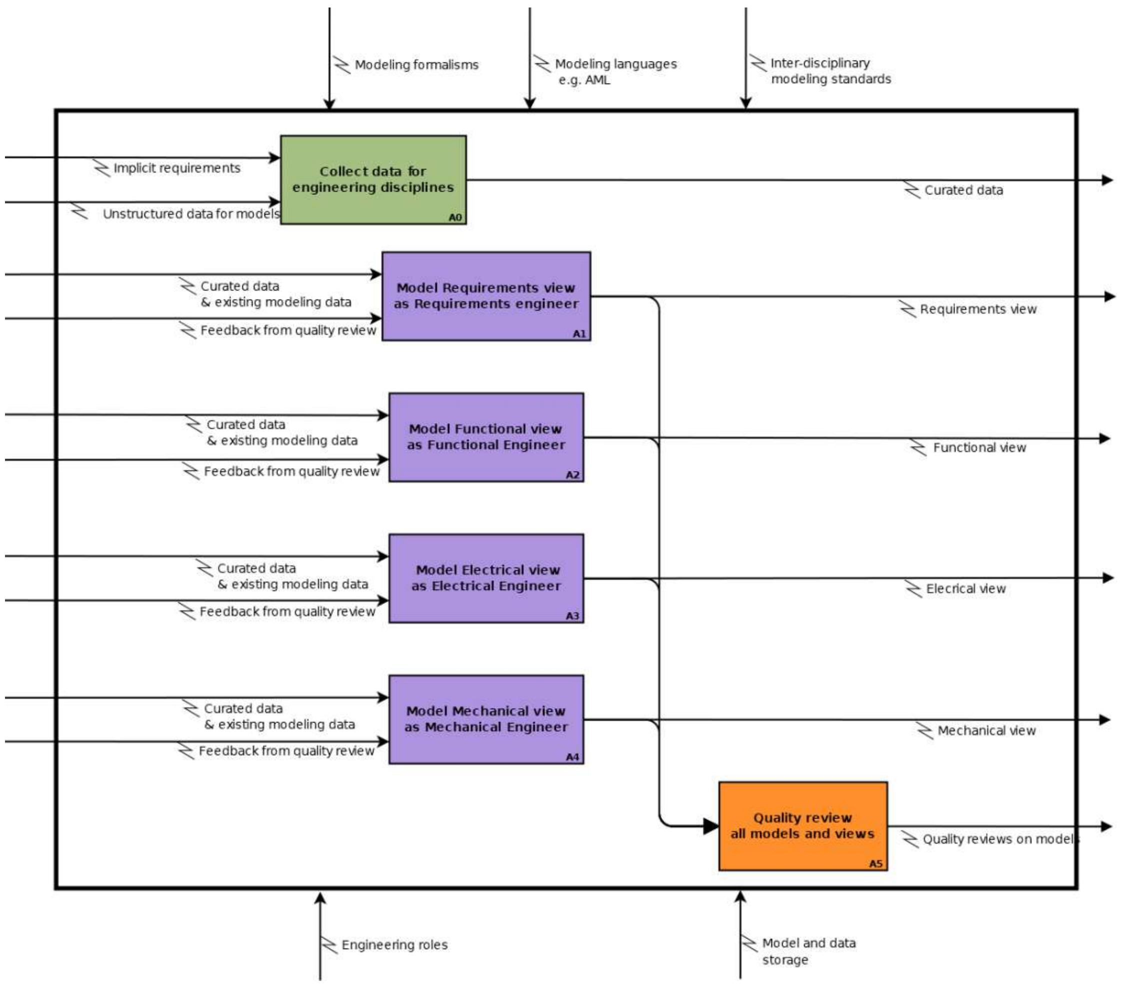

Modeling formalisms
Modeling languages e.g. AML
Inter-disciplinary modeling standards
Implicit requirements
Unstructured data for models
Collect data for engineering disciplines
A0
Curated data
Curated data & existing modeling data
Feedback from quality review
Model Requirements view as Requirements engineer
A1
Requirements view
Model Functional view as Functional Engineer
A2
Functional view
Model Electrical view as Electrical Engineer
A3
Elecrical view
Model Mechanical view as Mechanical Engineer
A4
Mechanical view
Quality review all models and views
A5
Quality reviews on models
Engineering roles
Model and data storage

Figure 4.2: Flow of the conducted use cases, modelled using the IDEF0 framework.

For each engineering discipline introduced in Figure 4.1, a process step, *A1* - *A4* exists. Here, the curated data works as an input. Furthermore, existing modelling data from other engineering disciplines can be used as input. Finally, feedback from quality reviews is incorporated as input. As a result from these inputs, a discipline-specific view on the CPPS is created.

Process step *A5* uses these specific views as input. This allows the quality manager to provide feedback on each discipline-specific view. This feedback is the output of process step *A5*. Figure 2.2 (cf. Chapter 2) depicts such a round-trip and feedback loop.

As represented in Figure 4.2, the use cases depend on the controls of modelling languages in general, their formalisms and related modelling standards. Furthermore, the use cases depend on the mechanisms and engineering tools of each engineering role. The use cases also depend on possibilities for model storage, to allow a persistence of model instances.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The arrows at the top and the bottom of the IDEF0 model represent these dependencies.

4.2 UC 1: Data Propagation between Model Views

This use case results from the analysis of the whitepaper for Automation Markup Language (AML) by Drath et al. [2020]. The whitepaper outlines a use case for AML, namely, a powertrain driving a linear drive. This use case represents the multi-disciplinary nature of such a CPPS engineering and modelling process and the relation to CPPS Engineering. This use case supports *RQ1* (see Section 3.2): which DSL allows linking of different engineering models.

4.2.1 Context and Motivation

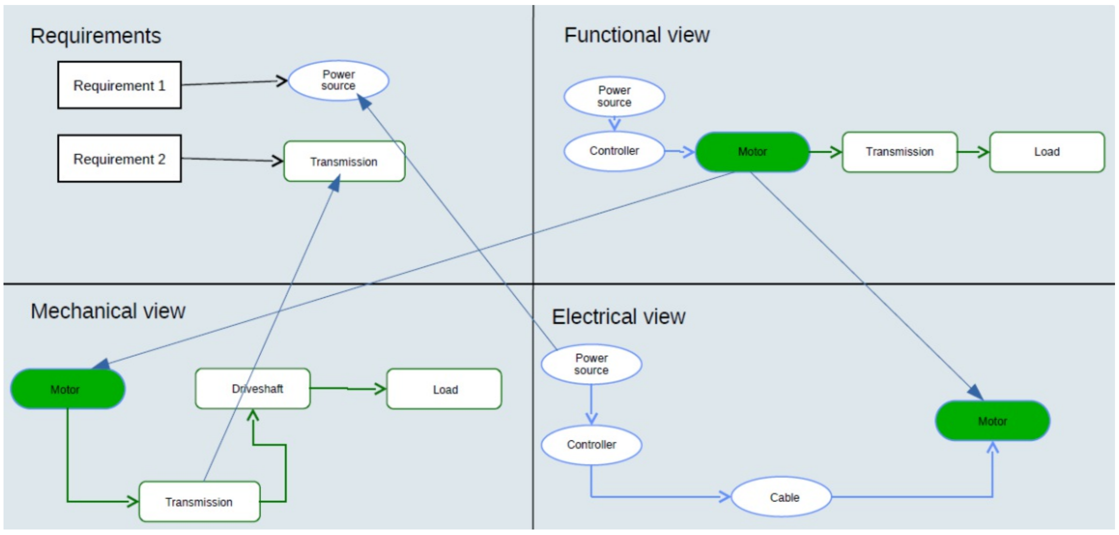


Figure 4.3: CPPS Powertrain Design: Multi-view model elements and dependencies.

This use case concerns the field of multi-disciplinary CPPS Engineering. It is a typical modelling use case where multiple engineering disciplines take part in designing, analysing, and improving a "real world" artefact, in this case a powertrain powering a linear drive [Drath et al., 2020]. To better understand the need for coordination, Figure 4.3 shows a mockup of the goal state of this use case with views from four different disciplines, i.e., the requirements, functional, mechanical, and electrical view. Model elements, like the motor or the transmission, are present in multiple, discipline-specific views, while still referring to the same artefact. To allow each engineer to work on the same underlying data this use case outlines the requirement to link these shared artefacts and propagate changes to element properties between the views.

The goal of this use case, therefore, is to aid such a coordination via an automated process, avoiding errors that result from outdated or not shared data instances between

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

different engineering disciplines. This use case concerns fulfilling requirements *R1* and *R2*, outlined in Section 3.4. Special focus is put on fulfilling the requirements for CPPS Engineering Network (CEN), defined by Requirement *R1*, since in this use case multiple different engineering disciplines work in their own view and local DSL. Section 4.2.3 outlines more fine-grained requirements resulting from this use case.

4.2.2 Use Case variants represented in RUCM

Table 4.1 outlines the steps to execute the described use case of connected CPPS modelling. Each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

4.2.3 Elicited Requirements

Elicited requirements from this use case are concerned with cooperation across discipline-specific boundaries. These requirements represent the need for different engineering disciplines to exchange data with other engineers. A difficulty is that each engineering discipline works in their own local toolset, like spreadsheets or CAD drawing, in the context of this work the DSL.

RQ.UC1.1: Each engineering discipline uses a DSL. This allows the engineer to work in their known environment, e.g., known representation of nodes.

RQ.UC1.2: Allow import and export of model elements from different engineering disciplines. This allows to take over existing model elements across disciplines that refer to the same real world element.

RQ.UC1.3: Linking of models to represent a CEN. This allows linking multiple views that reference the same modelled domain.

RQ.UC1.4: Propagation of changes to model element properties across discipline boundaries. This results in each engineer having the most up-to-date information for each model element automatically. This propagation must not only copy values, but also consider different representations of nodes and properties across discipline boundaries.

4.3 UC 2: Linking of Model Elements to Engineering Artefacts

The concept of a Product-Process-Resource (PPR) Asset describes a modelling approach, where the model consists of three types of nodes, namely the product, the production process, and the resource node that automates the production process. This allows the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Use Case Name	UC-1 Connected CPPS modelling
Brief Description	Goal of this UC is to facilitate efficient cooperation of engineers across discipline boundaries. This is achieved via data propagation and import/export of elements to other engineering disciplines. The goal is to design a drive chain with contributions coming from four engineering disciplines.
Precondition	A local DSL for each engineering discipline exists.
Primary Actor	1. Functional Engineer 2. Mechanical Engineer 3. Electrical Engineer 4. Requirements Engineer
Secondary Actors	Further engineers and actors that work with one of the models. Mostly in a passive role, e.g., viewing and extracting information
Dependency	No related use cases.
Generalization	This use case can be generalized by viewing and conducting it independently of specific engineering disciplines.
Basic Flow	1. Requirements Engineer starts modelling requirements for the drive chain in its local DSL. 2. Functional engineer starts modelling the drive chain and its components in local DSL. 3. Mechanical Engineer imports nodes and edges that are relevant to its blank canvas in mechanical engineering specific DSL. 4. Mechanical Engineer continues modelling the drive chain in specific view using the imported models. 5. Electrical Engineer imports nodes and edges that are relevant to its blank canvas in electrical engineering specific DSL. 6. Electrical Engineer continues modelling the drive chain in specific view using the imported models. 7. Requirements Engineer imports nodes and edges that are relevant to its blank canvas in electrical engineering specific DSL. 8. Requirements Engineer associates imported elements to existing requirements. 9. Due to the linking of the models, the system propagates the changes in shared elements and makes them visible to all engineers.
Alternative Flows	This use case can be augmented easily. Further engineering disciplines can take part in this process, increasing the amount of dependencies between models increases.

Table 4.1: RUCM table, representing UC1 - Connected CPPS modelling.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

modelling of production lines. The work of Schleipen et al. [2015] introduces this concept. This modelling concept allows representing a production process together with the CPPS production line that automates the process as required in the following use cases.

4.3.1 UC2a: PPR Asset Directory

This use case relies on the concept of a PPR Asset Directory, introduced by Biffl et al. [2021c]. The paper outlines the representation of a PPR asset directory, a use case from CPPS Engineering. This work selects this use case due to its relation to CPPS Engineering and the foundation it provides for the following use cases.

4.3.2 Context and Motivation

The context of this use case is a PPR Asset Directory [Biffl et al., 2021c]. In this Asset Directory, Biffl et al. [2021c] link the PPR Assets to Engineering artefacts. This link establishes a relation of each PPR Asset to an Engineering artefact, representing valid CPPS configurations and the Engineering artefacts that define a CPPS configuration. This allows efficiently querying related Engineering artefact and Assets. This establishes the name PPR Asset Directory, since a lookup like in a directory is possible to the user.

This concept aims at improving the traceability and transparency between Engineering artefacts and the assets representing them in the model. Figure 4.4 represents a mock-up representation of a PPR Asset Directory. This model bases on [Biffl et al., 2021b] and models the *"Mounting dashboard to car body"* use case. This task represents a typical task when assembling a car in a CPPS production line. An integral part of the car, in this case the dashboard, is screwed onto the car frame. For this, different drills, screws, and robots are necessary, which are modelled in Figure 4.4. Figure 4.4 models a mock up the *"Mounting dashboard to car body"* use case, introduced by [Biffl et al., 2021b].

This use case aims at fulfilling requirements *R1* and *R2* (see Section 3.4). Here, the main focus lies on *R2*, especially allowing model analysis since the links between Engineering artefacts and PPR Assets will be analysed by engineers. More fine-grained requirements, based on the requirements from Section 3.4, are outlined in Section 4.3.4.

4.3.3 Use Case variants represented in RUCM

Table 4.2 outlines all steps conducted to execute the described use case of a PPR Asset Directory. Each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

4.3.4 Elicited Requirements

The requirements elicited from this use case mainly concern the representation of links between different nodes. This is needed to allow an association of real world artefacts to nodes in the model.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

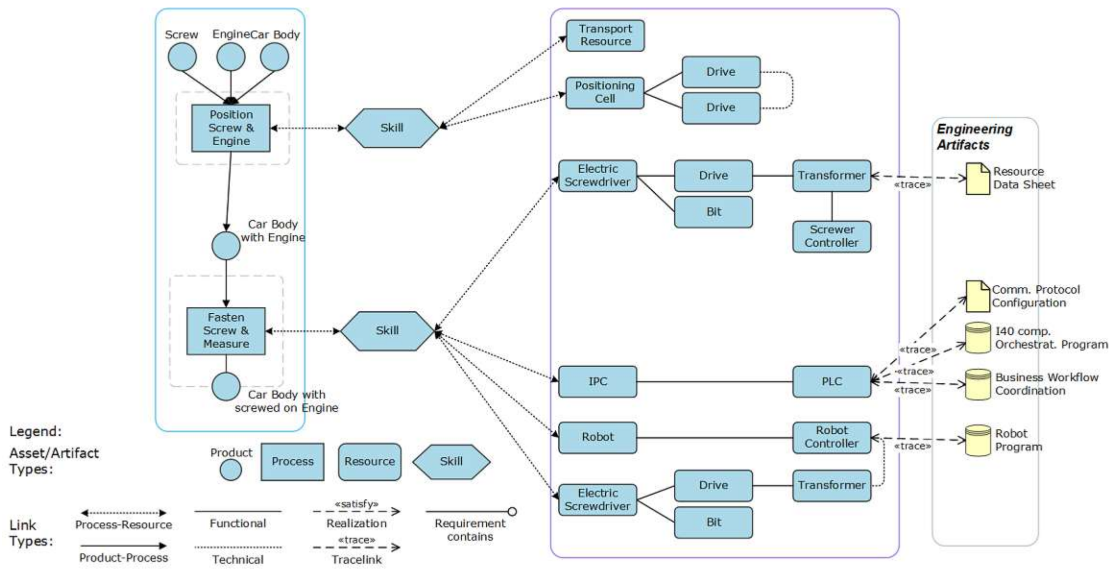


Figure 4.4: Paper mockup of the models to be modelled in UC2a and UC2b. The Figure shows the assets and links between the assets. Based on [Biffl et al., 2021b].

RQ.UC2a.1: Flexible representation of nodes and links, defined by a DSL. This capability allows the actors to choose their visual representation of nodes and links, e.g., using PPR Assets.

RQ.UC2a.2: Linking of nodes. This capability allows the actors to visually represent links between the nodes.

RQ.UC2a.3: Placement of nodes and links on canvas. This capability allows the actors to freely place nodes on the modelling canvas, which aids the representation of real world concepts in a modelling world.

4.3.5 UC2b: PPR Asset Network (PAN)

An augmentation of UC2a in Section 4.3, is based on [Meixner et al., 2021], introducing the term *PPR Asset Network (PAN)*. A PAN consists of nodes and links, in particular PPR Assets. In contrast to UC2a, nodes and links in a PAN can contain properties, which allows more complex data structures for model elements.

4.3.6 Context and Motivation

As outlined, the context of this use case is a PAN. In a PAN, edges link multiple nodes to each other, while also containing properties to each model element. This allows, like in UC2a, the linking and traceability between nodes while having the added benefit of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Use Case Name	UC-2a PPR Asset Directory
Brief Description	Goal of this UC is to represent a PPR Asset Directory. This means that PPR Assets can be linked to Engineering artefacts.
Precondition	A DSL containing PPR Assets and Engineering artefacts exists.
Primary Actor	Modelling Engineer
Secondary Actors	Further engineers and actors that work with one of the models in a passive role, e.g., viewing and extracting information.
Dependency	UC2b, Section 4.3.5.
Generalization	No generalization.
Basic Flow	1. Modelling engineer starts modelling a model consisting of the PPR Assets. 2. Modelling engineer introduces nodes, representing the Engineering artefacts to the model. 3. Modelling engineer links Engineering artefacts to PPR Assets. 4. Modelling engineer finishes complete model with links to all Engineering artefacts.
Alternative Flows	An alternative approach would be to model the PPR Assets after the Engineering artefacts have been added to the model.

Table 4.2: RUCM table, representing UC2a - PPR Asset Directory

storing properties to selected elements, which may be relevant to the modelled real world artefact.

The goal of PANs is to improve the representation of scattered knowledge in the CPPS Engineering domain, by allowing the representation of nodes and links and collecting properties to each model element, including dependencies between PPR Assets.

This use case aims at fulfilling requirements *R1* and *R2* (see Section 3.4). Here, this use case splits the focus between Requirement *R1* and Requirement *R2*, aiming at a representation of different links and node types, while still allowing the representation of properties for each model element. Section 4.3.8 outlines more fine-grained requirements, based on the requirements from Section 3.4.

To better visualize the relation of UC2a and UC2b, the same underlying model, represented in Figure 4.4 will be used.

4.3.7 Use Case variants represented in RUCM

Due to the relation to UC2a, some steps outlined in Table 4.3 are similar to Table 4.2. UC2b can be seen as evolution of UC2a. As in UC2a, each step will be executed using

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	UC-2b PPR Asset Network (PAN)
Brief Description	Goal of this UC is to represent a PAN.
Precondition	A DSL containing PPR Assets and Engineering artefacts exists. Furthermore, the DSL must allow the definition of properties for each node.
Primary Actor	Modelling Engineer
Secondary Actors	Further engineers and actors that work with one of the models in a passive role, e.g., viewing and extracting information.
Dependency	UC2a, Section 4.3.
Generalization	No generalization.
Basic Flow	1. Modelling engineer starts modelling a model consisting of PPR Assets and other node types. 2. Modelling engineer introduces nodes representing the Engineering artefacts to the model. 3. Modelling engineer links Engineering artefacts to PPR Assets. 4. Modelling engineer links remaining nodes in between each other where applicable. 5. Modelling engineer adds properties to relevant model elements. 6. Modelling engineer finishes complete model with links to all Engineering artefacts.
Alternative Flows	An alternative approach would be to model the PPR Assets and other node types after the Engineering artefacts have been added to the model.

Table 4.3: RUCM table, representing UC2b - PAN.

the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

4.3.8 Elicited Requirements

Requirements elicited in Section 4.3.4, are also relevant to this use case. This is why only extending requirements, compared to Section 4.3.4, are listed here. The requirements from UC2a are still relevant to this use case.

RQ.UC2b.1: Flexible definition of properties on model elements by a DSL. This capability allows the actors to define a set of properties that can be set for each model element.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

RQ.UC2b.2: Manipulation of model element properties. This capability allows the actors to manipulate the model element properties, e.g., changing the description contained in a node.

4.4 UC 3: Linking Requirements and Design Networks

Based on the context of PANs, Biffl et al. [2020, 2021b] introduces different real world applications based on such model networks. This work choses this use case due to the different representation of modelling nodes and layering in the models. The goal is to provide a use case leveraging requirements *R1* and *R3* of Section 3.4, improving the state of the art in the proposed solution approach in Chapter 5. To aid traceability, this chapter is divided into two sub-use cases. First, UC3a presents a simpler approach of modelling risk mitigation. Secondly, UC3b outlines a more complex approach of modelling requirements networks.

This use case relies on UC2, and also aims at answering *RQ2* from Section 3.2. As UC2, it focuses on *design and validation* in a more complex application scope.

4.4.1 UC3a: PAN Application Risk Mitigation

The work of Biffl et al. [2020] introduces the concept of Multi-Aspect Risk Exploration (MARE). This proposed method aims at representing risk exploration and risk mitigation in a multi aspect environment. Due to this nature of the method, it is chosen as a representative use case for this thesis.

4.4.2 Context and Motivation

This use case builds on the concept of Multi-Aspect Risk Exploration (MARE). This concept aims at exploring and mitigating risks in the field of CPPS Engineering, e.g., when designing a drive chain. This is achieved by designing an overlay application model as extension to the underlying technical PAN model, derived from data in CPPS engineering artefacts. This overlay application model does not affect the existing model, it aims at providing additional information that is not needed for the production process.

Such an overlaying MARE network can be seen in Figure 4.5. The Figure visualizes that the underlying PAN exists independently of the overlying Cause, Detection, and Mitigation nodes. Such a second layer of nodes allows analysing possible risks in the network, their causes and their possible mitigations, increasing the overall quality of the underlying network [Biffl et al., 2020].

Figure 4.5 represents a *4-color printer*. The aim of a 4-color printer is to allow printing of pages consisting of (max.) four different colours. To allow printing, multiple mechanical steps, e.g., a printing plate or a motor, are necessary. Risks that can arise in such a use case span from missing colours in the print to bad overall print quality.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

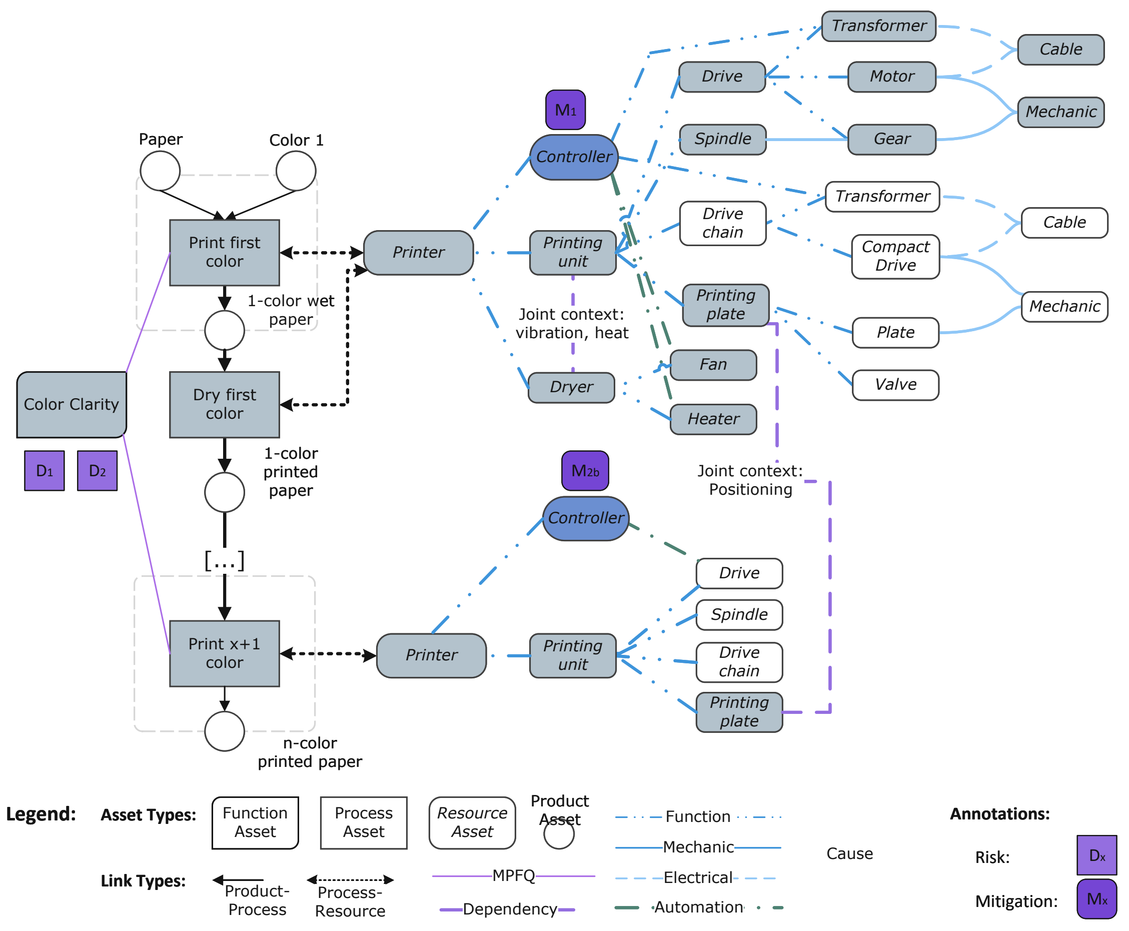


Figure 4.5: Paper mockup of the use case presented in UC3a and UC3b. Represents a 4-color printer, based on [Biffl et al., 2020].

The goal of this use case is therefore to allow a representation of an overlaying network, improving the quality of PANs. This provides quality engineers with an additional method in representing risks on models.

This use case aims at fulfilling the requirements *R1* and *R2* from Section 3.4. It relates to the representation of domain knowledge (*R1*) and also to the task-oriented visual representation of models (*R2*). Section 4.4.4 outlines further requirements resulting from this use case.

4.4.3 Use Case variants represented in RUCM

Table 4.4 outlines all steps conducted to execute the described use case of a PAN Application in the field of Risk Mitigation. Each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	UC-3a PAN application - Risk mitigation
Brief Description	Goal of this UC is to represent an application of PANs, with the focus on risk mitigation.
Precondition	A DSL containing PPR Assets and Engineering artefacts exists. Furthermore, the DSL must allow the definition of properties for each node. Also, the DSL must contain nodes representing risk artefacts, e.g., causes and mitigations.
Primary Actor	Quality Engineer
Secondary Actors	Further engineers and actors that work with one of the models in a passive role, e.g., viewing and extracting information.
Dependency	No related use cases.
Generalization	Can be generalized and used in other applications, as outlined in Section 4.4.5.
Basic Flow	1. Different CPPS Engineers model a multi-view model, e.g., a CPPS production chain. 2. CPPS Engineers inform Quality Engineer of a pending review. 3. Quality Engineer adds a new layer to the model. 4. Quality Engineer models risks, causes, mitigations etc. to existing nodes of model. 5. Quality Engineer informs that he finished the review. 6. CPPS Engineers can toggle risk mitigation layer as needed.
Alternative Flows	An alternative approach would be to model the risks directly to each node, instead of building a second layer to the model. This approach will be outlined in Section 4.5, leveraging the concept of markers.

Table 4.4: RUCM table, representing UC3a - PAN application, risk mitigation

4.4.4 Elicited Requirements

Requirements elicited from this use case concern mainly the representation of different layers to the models, allowing to build a “sub-model” to the existing model.

RQ.UC3a.1: Flexible addition of layers via the DSL. This capability allows the quality engineer to place its risk artefacts on a separate layer.

RQ.UC3a.2: Toggling of layers. This capability allows stakeholders, especially apart from the quality engineer, to toggle the layer containing risk artefacts. This allows a more clutter-free visual representation in cases where the risk layer is not relevant to the model.

4.4.5 UC3b: PAN Application Requirements Network

A different application approach for PANs is the usage in a requirements network, as outlined in Biffl et al. [2021b]. This augments the UC3a, presented in Section4.4.1. In contrast to UC3a, in this use case actors do not model risks but a network of requirements concerning the specific model elements.

4.4.6 Context, motivation and goals

As outlined, this use case aims at modelling a requirements network, relating to existing assets in a PAN. This requirements network aims at providing a better traceability for engineers, leading to an overall better model quality. This is achieved by having the requirements related to modelled assets in the PAN and allowing traceability between requirements, e.g., tracking changes to other affected parts of the models via links.

Since this use case is related to UC3a, UC3b uses the same underlying model for this demonstration. A mock-up of this model can be seen in Figure 4.5.

This use case is related mainly to requirements *R2* and *R3*. It relates to the visual representation, aiding model analysis and improvement (*R2*) and the data storage, concerned with the capability of storing information to assets in models. Also, requirements resulting of UC3a in Section 4.4.1 are relevant to this use case, together with the requirements outlined in Section 4.4.8.

4.4.7 Use Case variants represented in RUCM

Table 4.5 outlines the RUCM steps. As in UC3a, each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	UC-3b PAN application – Requirements Network.
Brief Description	Goal of this UC is to represent an application of PANs, with the focus on building a requirements network, linking assets to requirements and requirements between each other.
Precondition	A DSL containing PPR Assets and Engineering artefacts exists. Furthermore, the DSL must allow the definition of properties for each node. Also, the DSL must contain requirements nodes, with properties for each requirement asset, to allow the storage of requirement specific properties.
Primary Actor	Requirements Engineer
Secondary Actors	Further engineers and actors that work with one of the models in a passive role, e.g., viewing and extracting information.
Dependency	Relates to UC3a.
Generalization	Can be generalized and used in other PAN applications.
Basic Flow	1. Requirements Engineer models requirements and the relation between requirements. These requirements are modelled on a separate layer 2. Different CPPS Engineers model a multi-view model, e.g., a CPPS production chain on a separate layer. 3. CPPS Engineers inform Requirements Engineer of finished model. 4. Requirements Engineer links CPPS Assets to requirements between layers. 5. Requirements Engineer informs that he finished the modelling of requirements. 6. CPPS Engineers can toggle the Requirements layer as needed.
Alternative Flows	As in UC3a, an alternative approach would be to model the requirements directly to each node instead of building a second layer. This approach will be outlined in Section 4.5, leveraging the concept of markers.

Table 4.5: RUCM table, representing UC3b - PAN application – requirements network.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

4.4.8 Elicited Requirements

The elicited requirements in this use case relate strongly to the requirements elicited in UC3a, Section 4.4.1. Therefore, the requirements outlined in this use case can be seen as an addition to the already outlined requirements of UC3a.

RQ.UC3b.1: Linking of nodes existing in different layer. This capability allows the requirements engineer to model trace links between assets and requirements existing on different layers.

RQ.UC3b.2: Storing of properties on nodes not existing in the main layer. To aid the information storage and consolidation, it must be possible to store properties on nodes not existing in the main layer, e.g., the requirements layer of a model.

RQ.UC3b.3: Import/Export of the whole model, independently of layers. To allow an external manipulation and analysis of the model, all information of the model, independently of layers, must have the possibility to be exported.

4.5 UC 4: Coordination with Markers

This use case aims at improving the precision and efficiency of communication and coordination between engineers working on their engineering models by browsing, querying, and manipulating makers on model elements. As a foundation to achieve this capability, Chapter 2 introduces the concept of *coordination markers*. To better leverage the markers as coordination artefacts, the use case is divided into two sub - use cases. First, UC4a represents a guided task with markers, which aims at augmenting the existing model using markers as an asynchronous coordination tool between engineers. As a second use case, UC4b uses markers as a feedback tool on models. These two use cases show the possibilities that markers provide as an asynchronous coordination and cooperation tool.

As the final use case, this use case aims at answering *RQ2* and *RQ3* from Section 3.2. It relies on external tools to allow a more efficient analysis and improvement of models, whilst fully integrating the *design and validation loop* of *RQ2*.

4.5.1 UC4a: Guided Task with markers

This use case leverages the concept of coordination artefacts, introduced in Section 2.5. Furthermore, it uses the concept of Stigmergy, in this case of qualitative Stigmergy, since the actors taking part in this use case take decisions based on qualitative traces in the environment.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.5.2 Context and Motivation

As outlined, this use case leverages the concept of Stigmergy by placing coordination artefacts, so-called markers, in the environment. This means that CPPS Engineers can use markers to leave asynchronous traces for other engineers on the model.

Figure 4.6 outlines such a model containing markers. As can be seen, different markers in different states allow the asynchronous transportation of information between engineers. To better define the relations between the illustrated use cases, the same underlying use case as in UC3a and UC3b is chosen. The model in Figure 4.6 represent a 4-color printer, as introduced by UC3. Figure 4.6 again shows a printer that is able to (max.) print in 4 colours.

This use case aims to improve the asynchronous coordination and communication between different engineering disciplines in the CPPS Engineering field. This use case focuses on the model improvement part of the CMAI process.

Requirements *R2* and *R3* of Section 3.4 mainly motivate this use case. It specifically focuses on fulfilling the requirement of coordination between users in the CMAI process. Section 4.5.4 outlines more detailed requirements resulting from this use case.

4.5.3 Use Case variants represented in RUCM

Table 4.6 outlines all steps conducted to execute the described use case of guided tasks with markers. Each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

4.5.4 Elicited Requirements

The elicited use cases from this use case focus on the concept of markers and their handling. This thesis elicits the following requirements:

RQ.UC4a.1: DSL must allow definition of a marker with different states. This capability allows the lead engineer to place a marker for each node. Further, it allows base engineers to cycle through states while conducting their work. This allows the tracking of design and improvement lifecycle states, as is commonly used in agile software engineering.

RQ.UC4a.2: Manipulation of marker states. It must be possible for the users to cycle between the states of a marker defined in the DSL.

RQ.UC4a.3: Export/Import of markers. It must be possible to import and export markers from the Generic CPPS Modeling (GCM) Editor. This capability allows to leverage the data logistics approach introduced in Chapter 2. This thesis needs an export

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	UC-4a Guided task with markers.
Brief Description	Goal of this UC is to guide further engineers in improving and augmenting the existing model. This is achieved by setting corresponding markers and using Stigmergy in the coordination of tasks.
Precondition	A model exists that should be improved and augmented.
Primary Actor	Lead Engineer. This engineer sets the corresponding markers to initiate the guided tasks.
Secondary Actors	Base Engineer. Engineer that conducts the guided tasks and works on the improvement and augmentation of models.
Dependency	Builds upon and is applicable to all previous use cases.
Generalization	This UC can be adapted easily to fit any type of model that needs extension.
Basic Flow	1. Lead engineer develops base models that will be extended in the further steps. 2. Lead engineer sets markers on nodes that should be extended. These markers signal "to-do" to the base engineers. Each marker provides a short explanation of the task. 3. Base engineers see a "To-Do" marker, reads the task explanation and takes on the task. To signal to its peers that the basic engineer started the task, the engineer switches the marker state from "To-Do" to "In-Progress". 4. Base engineer fulfils the task. 5. After fulfilment, the base engineer sets the marker state to "Done" and notifies the Lead engineer. 6. The lead engineer can then send the extension to a review, or rerun the whole use case from the beginning if the models needs further extensions.
Alternative Flows	This use case can be augmented to fit multiple base engineers, taking up work as available and improving the model.

Table 4.6: RUCM table, representing UC4a - Guided task with markers

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

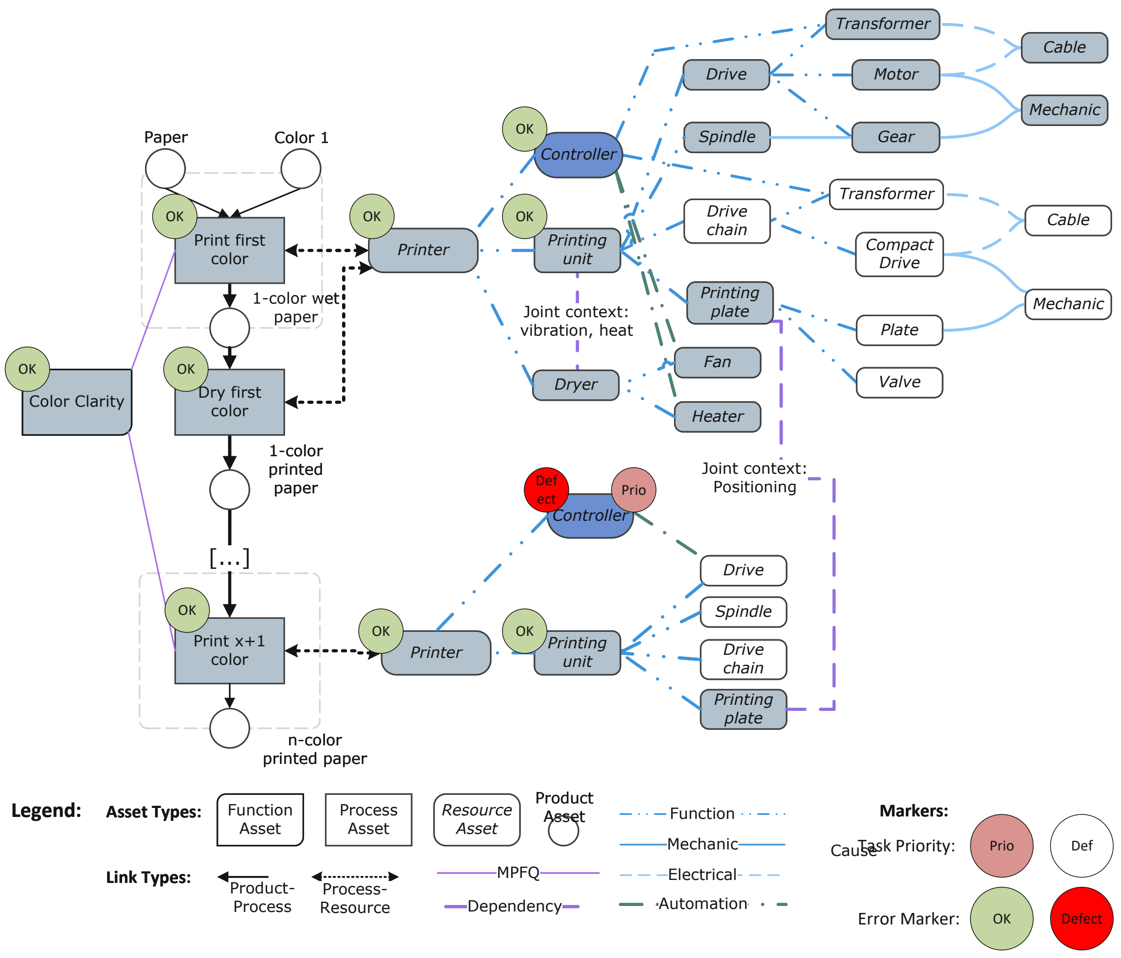


Figure 4.6: Mock-up of a model containing markers as coordination artefacts, based on Biffl et al. [2020].

and import to a Goal Question Metric (GQM) Editor, to allow the semi-automatic manipulation of markers via an external tool.

4.5.5 UC4b: Feedback with markers

As outlined in UC4a, this use case leverages the concept of Stigmergy and coordination artefacts. The use case relates strongly to UC4a, but shows a different field of application for coordination artefacts, leveraging the usage of semi-automatic marker placement.

4.5.6 Context and Motivation

To improve the overall quality of models, feedback is important for CPPS Engineers. To achieve this, this use case introduces an approach to provide feedback via markers.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

This use case is related to UC4a in Section 4.5. Therefore, the sketch of markers on model elements in Figure 4.6 is also relevant to this use case.

Goal of this use case is to provide feedback and improving the overall quality of CPPS models, by leveraging the stigmergic approaches outlined in Section 2.5.

Requirement *R2* and *R3* of Section 3.4 mainly motivates this use case, like UC4a. It focuses on allowing markers to be set semi-automatically by external tools, e.g., Neo4J. Section 4.5.8 outlines further requirements.

4.5.7 Use Case variants represented in RUCM

Table 4.7 outlines the RUCM definition of this use case. As in UC4b, each step will be executed using the solution approach described in Chapter 5. The fulfilment of the requirements of this use case by the proposed prototype will be evaluated in Chapter 6.

4.5.8 Elicited Requirements

This use case is related to UC4a in Section 4.5.1. Therefore, the requirements outlined here are an augmentation to the requirements outlined in Section 4.5.4.

RQ.UC4b.1: Properties in markers. This capability aids the asynchronous transportation of information between engineers. Furthermore, it helps in providing more accurate feedback that can not be implemented by markers themselves.

RQ.UC4b.2: Set markers semi-automatically, e.g., with a database query language. This capability allows users to leverage graph queries like provided in Neo4J to semi-automatically set a larger amount of markers to model elements.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Use Case Name	UC-4b Feedback with markers.
Brief Description	Goal of this UC is to provide another engineer with feedback. To achieve this, model querying, e.g., via Neo4J is leveraged to semi-automatically set markers for feedback in the environment.
Precondition	Engineer finishes with modelling the model and requires feedback.
Primary Actor	Modelling Engineer, that modelled the model that is reviewed.
Secondary Actors	Quality engineer, reviewing the model provided by the primary actor.
Dependency	Builds upon and is applicable to all previous use cases.
Generalization	This UC can be adapted easily to fit any type of model that needs review.
Basic Flow	1. Engineer provides the model in the local DSL to the quality engineer. 2. Quality Engineer begins visual reviewing in the GCM Editor. 3. Quality Engineer transforms the existing model into a different DSL, in this case into the Cypher language of Neo4J. 4. Quality Engineer imports transformed model into Neo4J. 5. Quality Engineer writes Cypher queries to better analyse and review the model. 6. Quality Engineer sets markers in Neo4J on the results if feedback is appropriate. 7. Quality Engineer exports results from Neo4J and imports them to the GCM Editor. The system transforms the model on import into the local DSL. 8. GCM Editor shows set markers on model. 9. Quality Engineer notifies Engineer on completion of review. 10. The engineer can now improve model based on feedback.
Alternative Flows	Alternatively, any other engineering disciplines capable of providing feedback can form part of this use case.

Table 4.7: RUCM table, representing UC4b - Feedback with markers

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

Coordinated Model Analysis and Improvement Approach

This chapter aims at providing concepts that allow answering the requirements and research questions outlined in Chapter 3 and Chapter 4. The goal is to provide concepts that allow *coordinated model analysis and improvement*, with special focus on applications in the Cyber-Physical Production Systems (CPPS) Engineering context.

The first section aims at answering RQ1, by providing a Domain-Specific Language (DSL) allowing for Model Analysis and Improvement (MAI), the Coordinated Model Analysis and Improvement (CMAI)-DSL. The second section introduces, based on the CMAI-DSL, a method to conduct coordinated model analysis and improvement, the CMAI method. The second section aims at answering RQ2. The last section introduces the software architecture and design, aimed at answering RQ3. This software architecture applies the results for RQ1 and RQ2 and designs a suitable software architecture around the research questions.

5.1 CMAI-DSL Metamodel (RQ1)

Table 5.1 represents the key concepts needed to fulfil the approach of *coordinated model analysis and improvements.*

To address these requirements this thesis introduces the Coordinated Model Analysis and Improvement (CMAI) concepts, starting with a *Model*, consisting of *Nodes*, *Markers* and *Links*. The paragraphs below introduce each concept in more detail, relating to the research questions and requirements:

Node. A *node* represents a real world artefact, e.g., a motor in a CPPS chain. Each node is of a certain type t. For each node, properties can be defined to store data, e.g.,

TU Bibliothek Your knowledge hub WIEN

Concepts	Concept Description
N; N_t	A *Node* of type t.
L; $L_t(N_1, N_2)$	A *Link* of type t between two Nodes N_1 and N_2.
M; M_t	A *Marker* of type t.
MO; MO(N, L, M)	A *Model*, consisting of Nodes, Links and Markers.
ML; ML(MO_1, MO_2)	A model link existing between models MO_1 and MO_2.
p(N), p(L), p(M)	Property of a Node, Link or Marker.
s(M)	A state of a Marker.
ma(N), ma(L)	A Marker related to a Node or a Link.

Table 5.1: Key concepts to address the requirements introduced in Chapter 3 and Chapter 4, based on Biffl et al. [2020].

storing a name and a description. Figure 5.1 represents a node with tag number 1.

Link. Based on nodes, the concept of *links* can be introduced. Like nodes, each link is of a certain type t. A directional link connects two Nodes, N_1 and N_2. To aid data storage, properties can also be defined on links. Figure 5.1 represent a link with tag number 2.

Marker. As introduced in Chapter 3 and in UC 4 in Section 4.5, coordination via artefacts in the environment plays an important part in asynchronous communication and coordination. Therefore, this thesis introduces the concept of markers. Each marker can consist of multiple states, but each marker can only be in exactly one state at each time. Markers can be set on Nodes and Links. To aid coordination, properties can also be set on markers to store important information for coordination between engineers. Figure 5.1 represents a Marker with tag number 3.

This concept relies on the foundations laid by Grassé [1959] regarding Stigmergy by leaving markers in the environment. This work introduces markers, allowing usage directly on graphs and aiding coordination, while leveraging stigmergic concepts.

Model. A Model describes the sum of *Nodes*. *Links* and *Markers* describing a real world artefact. Relating to the requirements introduced in Section 3.4, a model is the smallest subset of the concepts that must provide the capability to be imported and exported. This allows the approach of data logistics and round-trip engineering, as introduced in Chapter 2.

Figure 5.1 introduces the metamodel of model elements forming part of a model. As is outlined in Figure 5.1, a Model consists of *Nodes*, *Links* and *Markers*. Each Model can consist of multiple *Nodes*. Each Node can have multiple incoming and outgoing *Links*, that connect the *Nodes* to each other.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

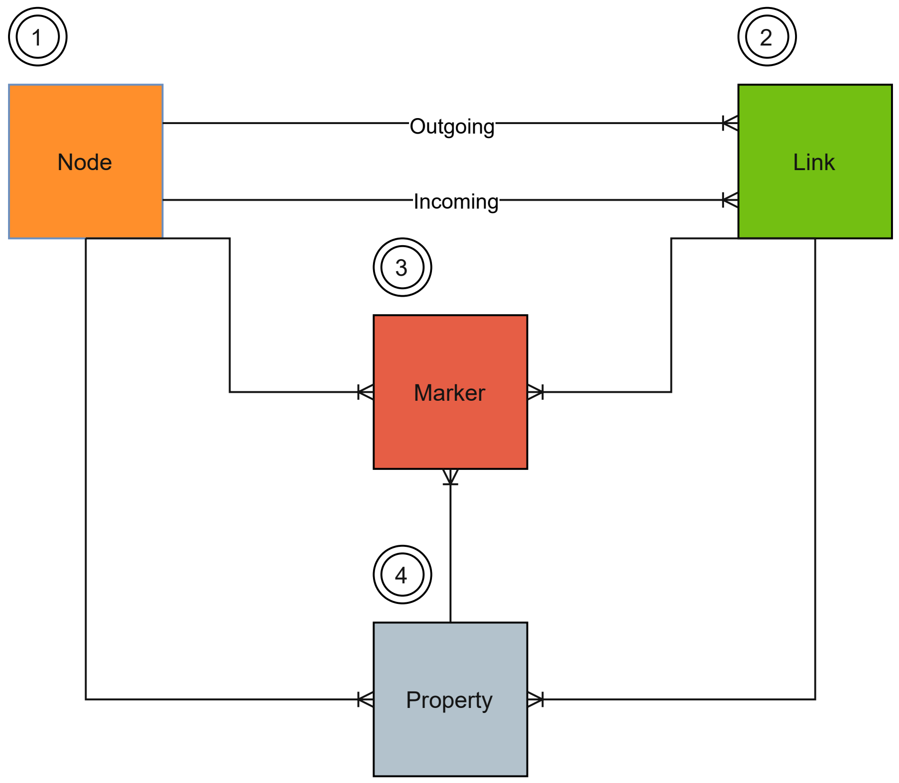


Figure 5.1: CMAI metamodel, based on Biffl et al. [2021c]. following the Entity-Relationship Diagram standard; Legend: cardinality between entities: an open arrow end means *Can have zero to many* entities.

Furthermore, Figure 5.1 outlines that each *Node* and each *Link* can have multiple *Markers*. Finally, *Nodes*, *Links* and *Markers* can have multiple properties.

This metamodel can be used to elaborate a configuration format to allow a definition of DSLs. It shows that a DSL that fulfils the research questions and requirements illustrated in Chapter 3 and Chapter 4, must allow the definition of at least these three model elements, each with their own respective properties. This DSL is founded by *R1 - Representation of domain knowledge*. This requirement outlines that typical CPPS Engineering Networks and coordination markers must be supported by the designed DSL. A technical introduction on the implementation of the metamodel can be found in Section 5.3.

5.2 CMAI Method (RQ2)

This section introduces the solution approach for the CMAI method. For this, the CMAI method uses concepts for change propagation and round trip engineering in a parallel and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

iterative engineering approach. This allows to lay out the theoretical concepts needed for conducting the CMAI method, which are then leveraged and used in Section 5.3, introducing the technical architecture. This section aims at answering RQ2 of Chapter 3.

5.2.1 Change propagation in multi-view model

To allow for *coordinated model analysis and improvements*, links between models play an important part. This concept of links between models aims at answering RQ2 of Section 3.2.

This concept goes beyond the current state of the art, by allowing a seamless and automatic transformation and propagation of properties in a cross-cutting, multi-disciplinary engineering field. These model links are directional links, allowing propagation of properties in between different models.

To achieve this, the CMAI-DSL holds a list of other CMAI-DSLs, each DSL is identified by a unique UUID. This represents a directional link between these DSLs. These links allow propagation, since the origin DSL can iterate over all linked DSLs and find matching model elements. The systems then compares the shared properties and updates them if needed. Furthermore, the system writes a change log, so that information reaches the engineers if changes to their models happen resulting from automatic change propagation.

5.2.2 Round-trip engineering

Resulting from *RQ2* and *R3* from Chapter 3, the need for an import and export of models arises. This is founded in the need to allow external manipulation and querying, apart from the DSL for the technical solution approach proposed in Section 5.3. Furthermore, it allows for data logistics approaches and round-trip engineering as introduced in Chapter 2. Parallel and iterative engineering especially needs round-trip engineering, where engineering involves multiple users.

Following the ideas put forward by Biffl et al. [2019b] this means that artefacts modelled in the context of a local DSL must be transformed to an external format and vice versa when importing. This capability allows handling models and model elements in external tools, allowing external enrichment of the model elements. This capability furthermore allows each engineering discipline to work in their own DSL and not relying on a single format across all disciplines.

This approach aims at allowing cooperation between different engineering tools and disciplines. Furthermore, it aims at avoiding divergence of data between different formats, if easy import and export is possible. Section 5.3 leverages this concept of round-trip engineering, to provide a semi-automatic way of setting markers on big models. This approach can be used in the CMAI method flow, proposed in the following subsection.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

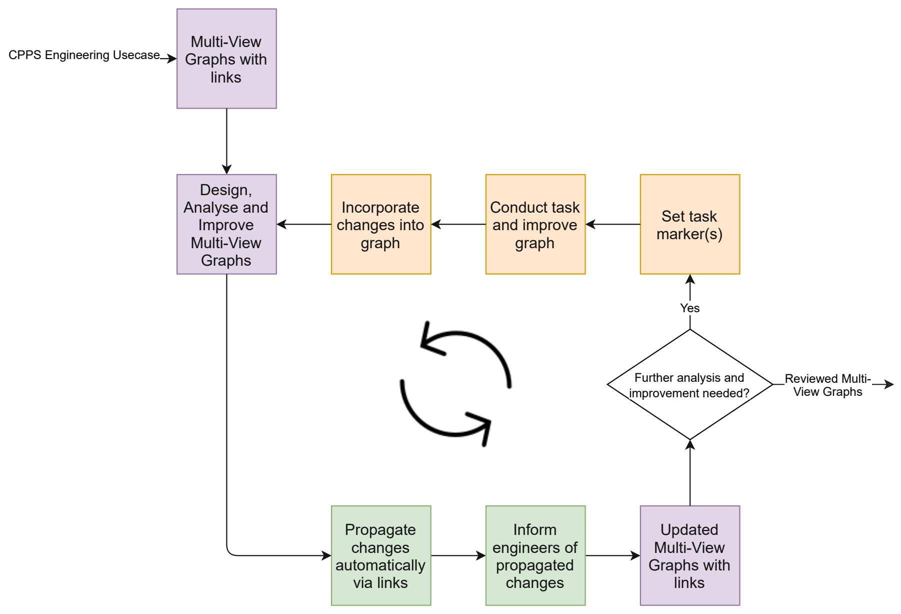

CPPS Engineering Usecase
Multi-View Graphs with links
Design, Analyse and Improve Multi-View Graphs
Incorporate changes into graph
Conduct task and improve graph
Set task marker(s)
Yes
Further analysis and improvement needed?
Reviewed Multi-View Graphs
Propagate changes automatically via links
Inform engineers of propagated changes
Updated Multi-View Graphs with links

Figure 5.2: CMAI method flow. Modeled using a flowchart based on Hering [1984].

5.2.3 CMAI method flow

5.2 represents the proposed CMAI method flow. It provides the capabilities for CPPS Engineers to conduct a *design and validation loop* in a Multi-View Graph Environment. It unites the proposed concepts of RQ1 and uses them in the CMAI method flow.

Graph Representation and Manipulation (CMAI-DSL). To represent typical graphs and models, the CMAI-DSL is used. This is shown by the purple quadrants in Figure 5.2. It furthermore allows CPPS Engineers to update and manipulate models in the CMAI-DSL.

Property Propagation between Graphs. Using the links between multiple CMAI-DSLs, proposed in Section 5.1, updates to graphs can be propagated to models of other engineers. This is shown in green in Figure 5.2. This makes property changes accessible to all engineers, improving the graphs by avoiding working on stale data.

Integrated Markers on Graphs. Propagated changes — or models in general — need to be reviewed. For this, this thesis proposes the concept of markers as integrated coordination tool, providing a stable way of describing tasks on graphs. This is shown in the orange rectangles in Figure 5.2. Since these tasks themselves may result in changes, the CMAI method flow results in a loop. The resulting changes are propagated, may need further review, which themselves may create new changes. This results in a *design and validation loop*, increasing overall quality and reducing risks in Multi-View Graphs.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.3 Software Architecture and Design (RQ3)

This section maps the theoretical concepts outlined in Section 5.1 and Section 5.2 onto specific technical solutions. The technical solution provides an implementation that allows the evaluation of the use cases introduced in Chapter 4. Based on the evaluation, the research questions of this thesis can be answered.

This section structures the topics as following: First, this section introduces the proposed technical architecture. Secondly, this section discusses the definition of a DSL. Finally, this section provides an outline for the import and export from and to the technical prototype.

5.3.1 Technical architecture

R2 of Section 3.4 strongly influences the technical architecture of the proposed *Generic CPPS Modeling (GCM) editor*. The need for a web architecture, avoiding a local installation, is founded in the ease of use for users of the Generic CPPS Modeling (GCM) editor. Furthermore, the architecture has to take into account the possibility of connectivity to external systems.

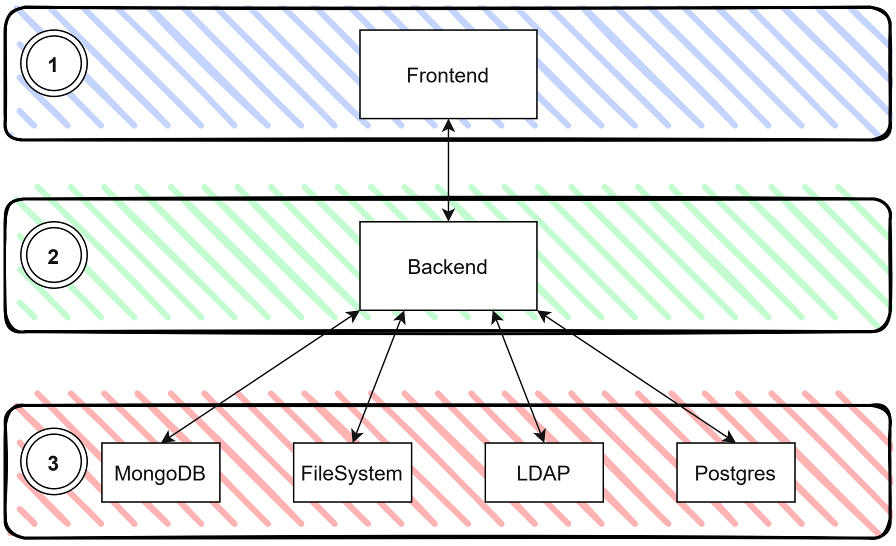


Figure 5.3: Technical architecture of the GCM editor.

To allow a reusage of the resulting technical architecture, an important concern was to choose an architecture that clearly separates the technical aspects by their concerns. Furthermore, state of the art software engineering approaches influenced the chosen architecture. To access and work on the resulting GCM Editor, readers can refer to a repository for the Frontend[1] and the Backend[2].

[1]GCM Editor Frontend Repository: `https://bitbucket.org/tuw-qse/mdre-frontend`
[2]GCM Editor Backend Repository: `https://bitbucket.org/tuw-qse/mdre-backend`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Figure 5.3 introduces the technical architecture of the GCM editor. It relies on the concept of separation of concerns, where each part of a system focuses on a specific concern. This is why the architecture separates itself into the following three layers: Frontend, Backend, and Persistence Layer. This layered architecture represents the state of the art in Software Engineering. Its clear separation of concerns aids in reducing the dependencies between each layer and increases overall comprehensibility for other engineers [Schatten et al., 2010]. This is why this thesis chooses this architectural approach.

Frontend. The frontend solely focuses on the representation and manipulation of models. It needs to represent the nodes, links, and markers to the users and needs to allow easy manipulation (*R2*). Furthermore, it needs to run in the web browser of the user, without requiring additional installation (*R2*).

Due to this requirement profile, *VueJS* [3] was chosen. This is due to the fact that *VueJS* is currently becoming more popular and works as a *single page web application*. This leads to a very small performance footprint, which is especially important when running the frontend in the browser of the user. Furthermore, *VueJS* integrates well with different Scalabale Vector Graphic (SVG) libraries, an important aspect in the representation of model elements in the editor, as will be outlined in Section 5.3.2. Figure 5.3 depicts the frontend layer with Tag *1*.

Backend. The backend of the GCM editor validates the models and relays them to the persistence layer. Furthermore, it relays additional metadata, e.g., the creation date of a model, to the persistence layer. It also provides the capability of import and export, as outlined in *R3*, since models can be transformed easily in the backend layer. External systems can be integrated in this layer.

To fulfil the requirements for the backend, *Kotlin* [4] was chosen as programming language. Strengths of this language are that it integrates with the Java Runtime whilst providing functional programming approaches to the developer. To aid the development, the *Spring Boot* [5] framework was chosen. Amongst other things, it provides a simpler integration of the Persistence Layer and aids development by abstracting boilerplate code. Figure 5.3 depicts the backend layer with Tag *2*.

Persistence Layer. The persistence layer stores the CMAI-DSLs and further information created by the GCM editor. The need for storing models is defined in Requirement *R3*.

[3]`vuejs.org` Last accessed: 05/07/2021
[4]`kotlinlang.org` Last accessed: 05/07/2021
[5]`spring.io` Last accessed: 05/07/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Due to the multitude of different data formats that need to be stored, this solution approach choses a *polyglot persistence* [6] layer. The persistence layer provides a *MongoDB* [7] database, a Not only SQL (NoSQL) database, to store the models. Additionally, the solution approach uses the *File System* to store images related to models. *LDAP* [8] is used to store user data and permissions for the GCM editor. Finally, this solution approach uses the *Postgres* [9] database to store metadata to the models and further information needed for the GCM editor. Existing approaches in the field partly influence the chosen persistence technologies, e.g., LDAP, on the other hand the need of storing non-relational data influences the chosen solution, as Chapter 2 introduces. Figure 5.3 depicts the persistence layer with Tag *3*.

5.3.2 Definition of DSLs

The flexible definition of DSLs is an integral part to answer the research questions of this thesis. R1 of Section 3.4 strongly influences the proposed DSL design. The idea of Automation Markup Language (AML), a neutral, XML based format to represent engineering data, influences the proposed solution [IEC 62714-1, 2018]. This thesis picks up this concept for a neutral data representation format, but uses JSON as underlying language, as will be outlined in the following paragraph. The metamodel for these DSLs has already been discussed. Therefore, this section focuses on the concrete implementation of this metamodel.

To allow a flexible definition of DSLs, the solution needs a data format that is human-readable, but also machine-readable. Due to this, the proposed solution choses JavaScript Object Notation (JSON) as data format to store a DSL. This provides a rigid data format, whilst still allowing human manipulation.

The following paragraphs provide an introduction into the data format of a DSL in the GCM editor:

Basic model elements (Abstract). All Nodes and Links in the GCM editor share basic properties, since they need to conform to similar requirements. Listing 5.1 outlines such an abstract basic model element.

The concept of a basic model element is abstract and does not form part of the DSL in the GCM editor. It is provided to the user as an abstract concept to aid the understanding of shared properties between Nodes and Links.

The shared properties are:

- `typeName`: The field `typeName` defines the type of the defined model element, e.g., a node of type `Product`. This property has to be unique across the DSL.

[6] `https://martinfowler.com/bliki/PolyglotPersistence.html` Last accessed: 05/07/2021
[7] `https://www.mongodb.com/` Last accessed: 05/07/2021
[8] `https://ldap.com/` Last accessed: 05/07/2021
[9] `https://www.postgresql.org/` Last accessed: 05/07/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

```
{
    "typeName":"Product",
    "attributeTypes":[
        {
            "variableName":"id",
            "displayName":"ID",
            "type":"string"
        }
    ],
    "layer": "ProductLayer"
    "display":"<svg width='100' height='100'><circle cx='50%' cy='50%' r='48' stroke='black' stroke-width
       ='4' fill-opacity='1' fill='white'/><text x='50%' y='50%' text-anchor='middle' fill='black' font-size='
       12'>{{attribute_id}}: {{name}}</text></svg>"
}
```

Listing 5.1: Properties shared by Nodes and Links in the DSL.

- `attributeTypes`: Defines attributes for the given model element. Consists (1) of a `variableName`, that can be referenced in the `display` of the model element. (2) of a `displayName`, which will be shown to the user (3) of a `type`, which allows “string”, “text” or “boolean” as types.
- `layer`: Allows the representation of a model element on a specific layer. All model elements will be placed on the defined layer. Layers can be toggled visually by the user in the GCM frontend. If the user omits this property, the system places the model element on a default general layer.
- `display`: To allow a flexible representation of nodes and links, the GCM editor leverages the concept of SVGs. Any SVG can be provided and will be displayed by the GCM editor. This allows different engineering disciplines to represent their domain specific model elements flexibly. Furthermore, the `variableName` can be referenced in the `display` string, to allow the representation of dynamic properties. Such reference onto a variable name needs to be defined by using double leading and trailing double curly brackets, e.g., `{{attributes_variableName}}`. Furthermore, the prefix `attributes_` needs to be set.

Furthermore, by default, each model element has a `name` and `description`. These are provided by the GCM Editor and do not need to be defined in the DSL. Based on this basic model element, this section introduces the concepts of Nodes and Links. To aid variability, this section abbreviates shared properties like the `attributeTypes` to maintain readability.

Nodes. To represent nodes in the GCM editor, the properties defined in Listing 5.2 can optionally be defined.

The additional property serves the following purpose:

- `connectPoints`: To aid a consistent representation of links on nodes, `connect-Points` can be defined in the DSL. Links only will connect to these `connectPoints`.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

```
{
   "typeName":"Product",
   "attributeTypes":[],
   "connectPoints":[
      {
         "x":50,
         "y":105
      },
      {
         "x":50,
         "y":−10
      }
   ],
   "layer":"ProductLayer",
   "display":"PLACEHOLDER"
}
```

Listing 5.2: Properties optionally defined on Nodes in the DSL.

The coordinates are relative to the middle of the SVG. If the user defines no `connectPoints` on a node, links find the closest path between the linked nodes.

To connect nodes, the following paragraph introduces the concept of link.

Links. Links allow the connection of two nodes in the GCM editor. To achieve this, Listing 5.3 outlines the additional properties on links in the DSL.

The additional properties on links serve the following purpose:

- `connects`: Defines rules on which connections the link can be used. If the user omits `connects`, links can connect any node. To further aid the visual representation `to/fromAllowedConnectPoints` allows binding specific connect points to each link.
- `bidirectional`: Allows the definition if this link is uni- or bidirectional. By default, all links in the GCM editor are unidirectional.

Based on the nodes and links, a DSL allowing the modelling of a graph can be defined. To answer the research questions aimed at asynchronous cooperation between engineering disciplines, the following paragraph introduces the concept of markers.

Markers. Markers can be defined additionally for each model element. Listing 5.4 outlines the properties needed to define a marker.

Each marker consists of the following properties:

- `typeName`: As with other model elements, it defines the type of marker.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
{
    "typeName":"Directional Dashed Arrow",
    "attributeTypes":[],
    "connects":[
        {
            "from":"Resource",
            "fromAllowedConnectionPoints":[
                {
                    "x":−5,
                    "y":50
                },
                {
                    "x":105,
                    "y":50
                }
            ],
            "to":"Process",
            "toAllowedConnectionPoints":[
                {
                    "x":−10,
                    "y":50
                },
                {
                    "x":105,
                    "y":50
                }
            ]
        }
    ],
    " bidirectional ": false ,
    "display":"PLACEHOLDER"
}
```

Listing 5.3: Properties optionally defined on Links in the DSL.

```
{
    "typeName":"ReviewMarker",
    "markerStates":[
        {
            "stateName":"In Review",
            "display":"<svg width='40' height='40'><rect width='40' height='40' stroke='black' stroke−width='
  2' fill='white' /><text x='50%' y='50%' text−anchor='middle' font−weight='bold' fill='black'>IR</
  text></svg>"
        },
        {
            "stateName":"Approved",
            "display":"<svg width='40' height='40'><circle cx='50%' cy='50%' r='19' stroke='black' stroke−
  width='2' fill='green' /><text x='50%' y='50%' text−anchor='middle' font−weight='bold' fill='black'>
  A</text></svg>"
        },
        {
            "stateName":"Discussion",
            "display":"<svg width='40' height='40'><circle cx='50%' cy='50%' r='19' stroke='black' stroke−
  width='2' fill='white' /><text x='50%' y='50%' text−anchor='middle' font−weight='bold' fill='black'>
  D</text></svg>"
        }
    ]
}
```

Listing 5.4: Properties defined on Markers in the DSL.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
{
  "modelType": "Complete−Model",
  "version": "1.0.0",
  "nodeTypes": [
    /∗ NODE TYPE PLACEHOLDER ∗/
  ],
  "edgeTypes": [
    /∗ EDGE TYPE PLACEHOLDER ∗/
  ],
  "markerTypes": [
    /∗ MARKER TYPE PLACEHOLDER ∗/
  ],
  layers: []
}
```

Listing 5.5: Shortened example of a model DSL

- `markerStates`: Contains a list of marker states. Each marker can only be in one state at one time. A marker state consists of the following properties:
  - `stateName`: Defines the name of the marker that is shown to the user in the GCM editor frontend.
  - `display`: As with model elements, any given SVG can be used to represent a marker.

To further aid coordination via markers between different engineers, markers, by default, have a name and description property. Via the description, additional information can be transported between users. Furthermore, to avoid cluttering of the GCM editor frontend, the GCM Editor automatically places each marker type on a distinct layer, so each marker type can be toggled individually.

In the following paragraph, a complete DSL is outlined, containing all mentioned concepts.

Models. Finally, all model elements can be summarized into the model DSL. Listing 5.5 depicts such a configuration.

As is visible in Listing 5.5, the DSL contains further metadata such as a `modelType`, to define a name for the DSL and a `version`, to allow a versioning of DSLs. In the field `layers`, all referenced layers in model elements have to be outlined.

Based on the configured DSL, concrete models can be modelled. To allow the concept of round trip engineering, as outlined in Chapter 2, import and export of models is crucial and is introduced in the following section.

5.3.3 Import and Export from and to different data formats

As introduced in Chapter 2, import and export of data into the Generic CPPS Modeling (GCM) editor plays an important part in settling the proposed software architecture in the state of the art. A known approach in this field is the *round trip engineering*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

```
MERGE (`Magnet`:`Product Asset`{`mdreId`:"4ead4476-b553-48c0-8392-50
    ab2e1b45b1",`mdreDescription`:"Magnet",`mdreXPosStart`:"296",`
    mdreYPosStart`:"323",`mdreXPosEnd`:"149",`mdreYPosEnd`:"-4",`id`:"x.ppr.
    PD2",`name`:"Magnet",`type`:"x.ppr.pd01",`note`:"Product type"})

MERGE (`Grundaufbau`)-[:`Process-Resource`{`mdreId`:"d2218ce1-dd10-40ae-a410-
    d872bb28edde",`mdreEdgeStartNode`:"c558d97b-62a8-409d-bda4-cf6a71af179e
    ",`mdreEdgeEndNode`:"2114af0b-d982-4d64-abaf-6f71e75fd183"}]->(`Modul 4`)
```

Listing 5.6: Example export from the GCM editor to cypher. Some properties omitted for readability.

approach Lüder et al. [2018]. In this approach, multiple engineering tools and disciplines work sequentially on the artefact, while transforming the artefact into their own data model between each step. To integrate the GCM editor into the existing approaches and tools, it must provide its own import and export interface. This approach strongly relates to R3 in Section 3.4.

This section introduces import and export via cypher and comma-separated values (CSV). Due to the architecture of the GCM editor, the chosen solution allows an easy addition of further import and export formats. The two introduced formats serve as examples on how such a behaviour can be achieved.

Import/Export via Cypher

The import and export to the cypher format, used by Neo4J, aims at answering RQ3 of Section 3.2. This interface allows the modification of properties via an external tool in a semi-automatic manner. To achieve this, this thesis leverages Neo4J as an external tool, due to its leading role in the field of Graph Query Languages (GQLs).

Export from the GCM editor. To achieve this, the data format introduced in the preceding section needs to be transformed into the cypher language, which is used by Neo4J.

Listing 5.6 outlines two exemplary Cypher statements, exported from the GCM Editor.

The first statement shows a `MERGE` statement for a node. `MERGE` allows the creation of a new element. If the specified node already exists, the new data gets added when using `MERGE`. The statement consists (1) of the name of the node, (2) of the type, and (3) of the properties from the node. In the properties, the properties defined in the DSL get stored.

The second statement represents the creation of a link. `(Startnode)-[Link]-(Endnode)` specifies a link. As with nodes, the definition of a link consists of (1) a name, (2), the link type, and (3) the properties of the link.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

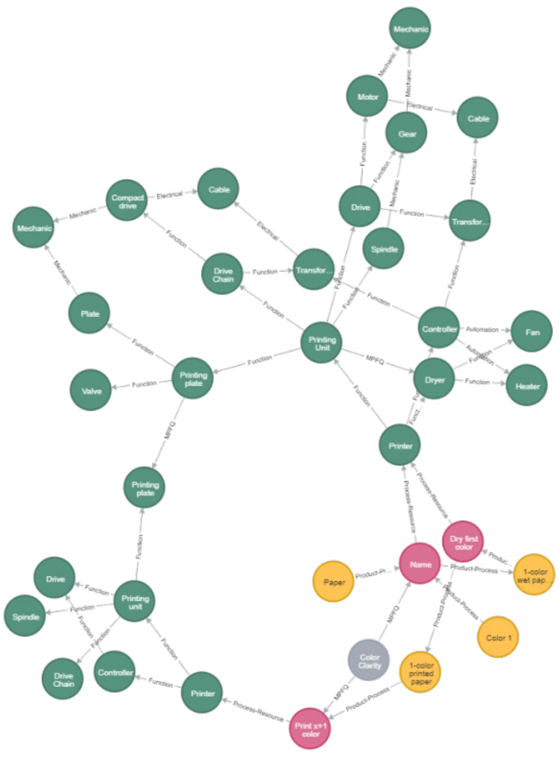


Figure 5.4: Exemplary graph from the GCM Editor, represented in Neo4J. This graph will be reused in Chapter 6

When exporting from the GCM editor, the GCM Editor sets these properties according to the stored data automatically. For each stored node and link, the GCM Editor creates the cypher `MERGE` statements and can be used to transfer an existing model to Neo4J. To further understand the possibilities of cypher, the Neo4J homepage [10] provides a further introduction. Figure 5.4 shows a graph represented in Neo4J. Chapter 6 goes into more detail showing use cases conducted using Neo4J.

Modifications via Cypher. The aim of using Neo4J as an external tool is to allow semi-automatic modification of the models, e.g., setting new markers or changing properties. Using a GQL may be easier than using the GCM frontend when handling bigger and more complex models.

After exporting a model from the GCM editor, cypher queries can be written that modify the model. Listing 5.7 outlines such a query.

The query matches all nodes, where the `Change State` is set to `changed`. For all matching nodes, the query modifies the related nodes. On the related nodes, the `ValidationLifeCycleState` is set to `To Validate`. Such a query may prove useful

[10] https://neo4j.com/developer/get-started/ Last accessed: 09/07/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
MATCH (startnode)<-[edge*1..1]-(endnode)
WHERE startnode.`Change State` = "changed"
SET endnode.ValidationLifeCycleState="To Validate"
```

Listing 5.7: Example query to modify a model via cypher.

when tracking changes to a model, that require further validation. In this example, the new value of `ValidationLifeCycleState` is visible in the GCM editor after reimporting the model.

Cypher further allows path queries or other path traversal queries. This allows many potential use cases of such modifications on models.

Import to the GCM editor. To allow a stable import to the GCM editor from cypher as an external data source, some defined steps have to be followed. To match the expected data format of the GCM editor, the query in Listing 5.8 needs to be used.

```
MATCH (startnode) OPTIONAL MATCH (startnode)-[edge]->(endnode)
WITH startnode, edge, endnode
RETURN startnode, edge, endnode
```

Listing 5.8: Export query for Neo4J.

Neo4J allows the export of the result of the query as JSON. The GCM Editor allows the import of this exported JSON file. Any modifications that were undertaken while using Neo4J will be imported to the GCM editor and will be visible in the GCM frontend.

This approach focuses on models that were already modelled in the GCM editor. To improve the usability when creating a model from scratch, the following section introduces an import and export via CSV.

Import/Export via CSV

To better establish the GCM editor in the State of the Art, a CSV import and export is possible. This allows large quantities of nodes to be imported, which may prove helpful when setting up new models. Furthermore, it helps to connect existing tools and approaches, since CSV is a well established approach that has been formalized in RFC-4180 [11].

Import via CSV. The import via CSV needs a header line. After the header line, each record has to be placed on a separate line.

Listing 5.9 outlines such a CSV import. Before defining the nodes, `Node Data` must be part of the imported file, to aid the GCM editor which part of the file is relevant for the import. After this block, the first line represents the header. It contains all properties

[11] `https://www.ietf.org/rfc/rfc4180.txt` Last accessed: 09/07/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
Node Data:

"NodeType","name","Description","ValidationLifeCycleState","Change State"
"Asset","M.Torque","Screw & Measure","designed","no_change"
```

Listing 5.9: Example for a CSV import into the GCM editor.

```
"EdgeType","Name","ConnectFromId","ConnectToId","Description"
"Function","Name","b365f599-95e2-444d-8e9e","c1829e49-f079-40dd-900e",""
```

Listing 5.10: Example for a link CSV export.

that were defined in the DSL. For each node type, a block with a header and the concrete instances below it needs to be created, to allow an import into the GCM editor.

This approach allows a fast and easy import, since it allows copying of instances in bulk, without having to use the GCM frontend.

Export via CSV. Similar to the import, the export exports all nodes and edges - and their properties - of a model in the CSV format. To achieve this, the GCM Editor uses the same data format as for the import. This allows to incorporate the GCM editor in an engineering tool chain, when relying on the CSV format.

Listing 5.10 outlines a link, contrary to the input. Providing a link during import in the shown format is possible, but isn't intuitively to use in more complex models. Therefore, it is suggested to use the GCM frontend to place links between nodes.

The following chapter evaluates the research questions using the introduced technical solution. To guide the evaluation, the following chapter uses the use cases introduced in Chapter 4.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 6

Evaluation

Goal of this chapter is to evaluate the research questions and requirements outlined in Chapter 3. To achieve this, this chapter uses the technical solutions proposed in Section 5.3. Using these technical solutions, this thesis conducts the illustrative use cases of Chapter 4. These use cases then allow an evaluation and discussion of research question and requirements in Chapter 7.

The remainder of this chapter treats the relevant topics as follows: This chapter deals with each use case outlined in Chapter 4 in a separate section. Each section introduces the test cases used for the evaluation and documents the execution of the test cases with screenshots and explanations. Lastly, each section gives an outlook on variants of the use case. To define the goals of each use case, this chapter uses the Goal Question Metric (GQM) approach as introduced by Caldiera and Rombach [1994]. The GQM provides a structured approach on how to measure (software) metrics, by dividing the measurement on three levels, the goal level, the questions level, and the metric level. Due to this simple template, it will be used in the evaluation of each use case, to define the goals and metrics. Chapter 7 outlines the discussion of the evaluation results.

6.1 Capability 1: Change Propagation

This use case analyses data propagation between engineering disciplines while using different local Domain-Specific Languages (DSLs). This is done for the purpose of improving data flows between engineering disciplines in fields of multi-disciplinary engineering, with respect to reducing effort of data dissemination and aggregation between disciplines and improving overall model quality. This thesis conducts this use case from the view point of engineers in their respective disciplines in the application context of Cyber-Physical Production Systems (CPPS) engineering.

Nr.	Name	Relates to
UC1.1	Import of model elements	RQ.UC1.1, RQ.UC1.2
UC1.2	Propagation of data	RQ.UC1.3, RQ.UC1.4

Table 6.1: Test cases used for evaluation of UC1

6.1.1 Test cases

Table 6.1 introduces the two test cases used to evaluate the use case.

Test case number 1 focuses on the possibility of allowing engineers to import shared model elements from other CPPS models. These model elements can possibly be defined in a different DSL, which proves as a challenge in existing approaches.

Test case number 2 works on an existing network of related CPPS models. To improve overall quality and reduce errors, shared properties on model elements in related model elements get automatically propagated, and the Generic CPPS Modeling (GCM) Editor informs users of these external changes.

6.1.2 Use case execution

To document each use case this section uses screenshots and explanations for each step. The underlying models used for this use case can be found, together with other materials of this thesis, in an external repository [1].

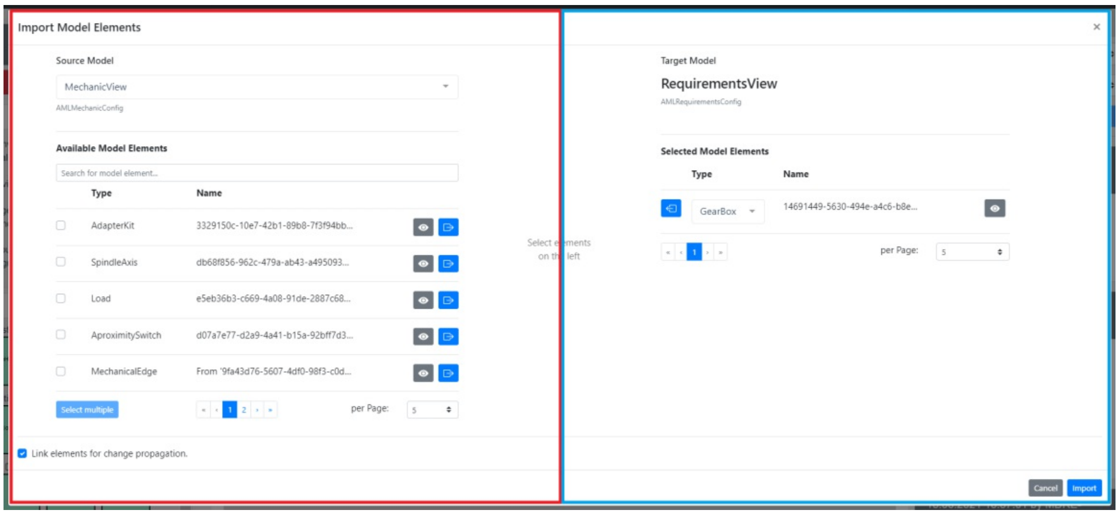
Test case 1 Relating to the Restricted Use Case Modeling (RUCM) steps for this use case outlined in Table 4.1, this test case relates to all steps, where an engineering discipline imports existing model elements into its own view. In this specific case, the evaluation represents the import of a *Gear box* model element into the requirements view. This allows a requirement engineer to trace existing requirements onto a shared model element, in which the system updates properties if they change in other engineering disciplines.

In Figure 6.1 shows the import of a shared model element from the Mechanic view. The left-hand side (red), represents all model elements present in the Mechanic view. Here, model elements can be shown and selected for import. The Figure shows all selected model elements for import on the right-hand side (blue). The GCM Editor automatically translates the representation of the model element into the local DSL. Furthermore, the two models get linked, so that the GCM Editor can keep each shared property on the imported node up to date automatically between the two views.

Figure 6.2 shows the resulting model of the import. The imported model element *Gear box* is selected and its properties shown (blue rectangle). The left-hand side (red) shows

[1]Model Configuration Repository: `https://bitbucket.org/tuw-qse/mdre-thesis-cengelbrecht`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Import Model Elements

Source Model

MechanicView

AMLMechanicConfig

Available Model Elements

Search for model element...

	Type	Name
☐	AdapterKit	3329150c-10e7-42b1-89b8-7f3f94bb...
☐	SpindleAxis	db68f856-962c-479a-ab43-a495093...
☐	Load	e5eb36b3-c669-4a08-91de-2887c68...
☐	AproximitySwitch	d07a7e77-d2a9-4a41-b15a-92bff7d3...
☐	MechanicalEdge	From '9fa43d76-5607-4df0-98f3-c0d...

Select multiple « ‹ 1 2 › » per Page: 5

☑ Link elements for change propagation.

Select elements on the left

Target Model

RequirementsView

AMLRequirementsConfig

Selected Model Elements

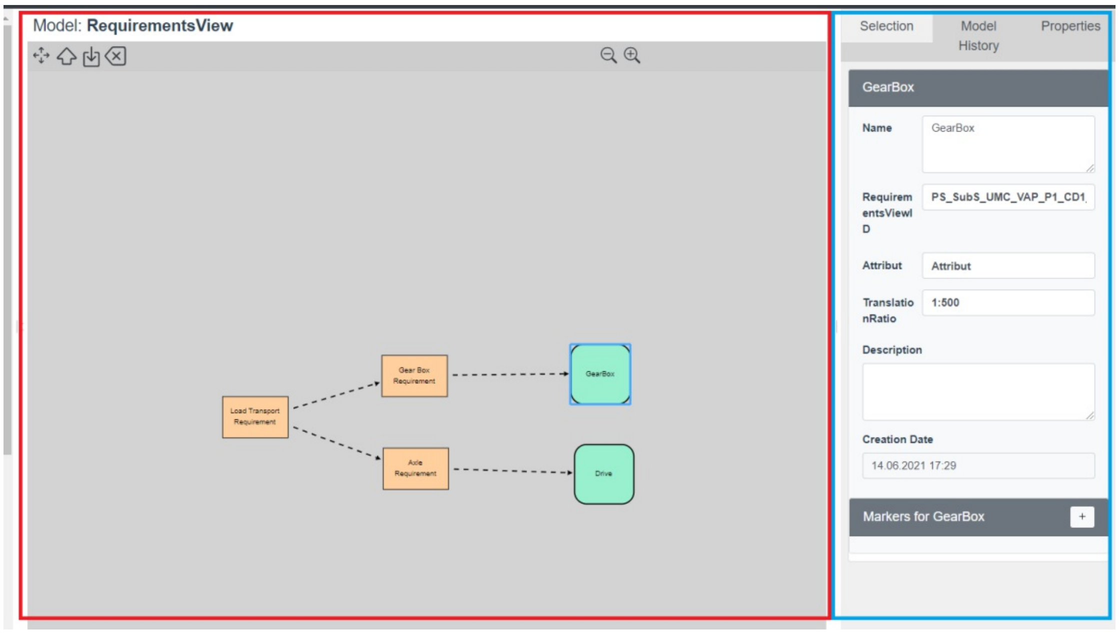
Type	Name
GearBox	14691449-5630-494e-a4c6-b8e...

« ‹ 1 › » per Page: 5

Cancel Import

Figure 6.1: Import of model elements in the GCM Editor.

the editor canvas of the GCM editor. This canvas allows the requirements engineer to link the imported model element to existing requirements. The properties shown on the right-hand side (blue) are shared (partly) with the mechanic view, so the GCM Editor propagates any change, which may lead to a violation of the requirement, in this case the *translation ratio*.



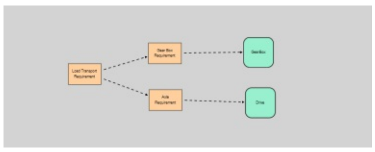
GearBox

Name	GearBox
RequirementsViewID	PS_SubS_UMC_VAP_P1_CD1,
Attribut	Attribut
TranslationRatio	1:500
Description	
Creation Date	14.06.2021 17:29

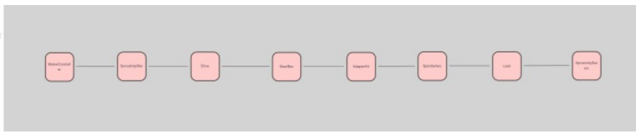
Markers for GearBox +

Figure 6.2: Result of import of model element into requirements view.

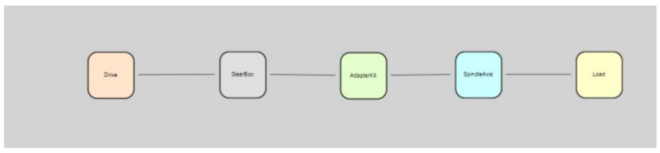
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



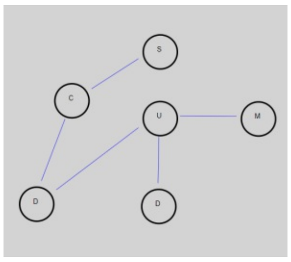
(a) Requirements View



(b) Functional View



(c) Mechanic View



(d) Electric View

Figure 6.3: The four engineering disciplines specific views represented in the GCM editor. They are for the evaluation of UC1 and relate to the paper mock-up in Figure 4.3.

Test case 2 Test case 2 focuses on the automatic propagation of information between engineering views on shared model elements.

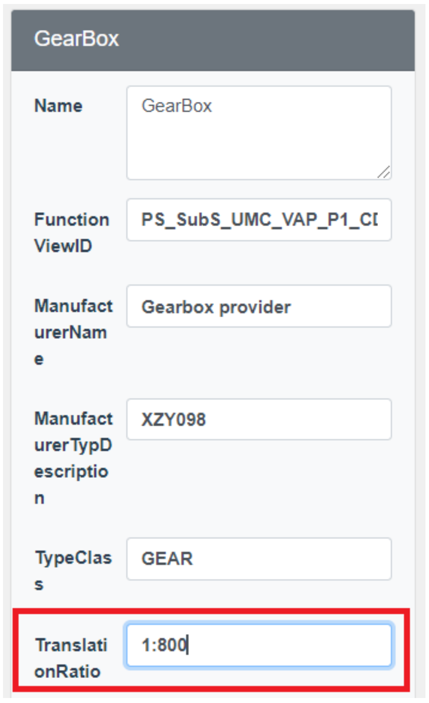
Figure 6.3 shows the four engineering views, introduced in the mock-up in Figure 4.3. As can be seen, the model element *Gear box* exists in three independent views. To keep shared information up-to-date between these models, an automated propagation is relevant to improve overall quality and consistency of the models.

Figure 6.4 shows this automatic propagation. First, the model element *GearBox* is modified in the Functional View and its property *TranslationRatio* changed to 1:800. Since the Requirements View and the Functional View share this model element, the GCM Editor propagates this change on save. Albeit the Requirements View using a different DSL, the GCM editor can propagate this information across engineering disciplines. The GCM Editor adds an entry into the *Model History* to inform the user of external changes. The entry contains all changed properties, in this case the model history informs the user that the *GearBox* was changed: `GearBox.TranslationRatio: 1:500 -> 1:800`. This entry in the model history is visible in Figure 6.4b.

6.1.3 Advanced Application Options

The possibility of linking models between them is easily scalable. The GCM editor theoretically allows an endless amount of different linked models. This method allows possible different applications, e.g., when designing different versions of the same model. By linking these models, the GCM Editor propagates all property changes to all different versions of the same model, keeping the models consistent in-between them automatically.

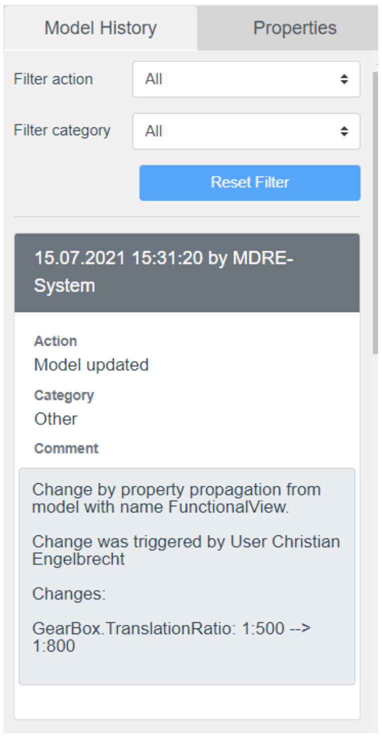
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



GearBox

Name	GearBox
Function ViewID	PS_SubS_UMC_VAP_P1_CI
ManufacturerName	Gearbox provider
ManufacturerTypDescription	XZY098
TypeClass	GEAR
TranslationRatio	1:800

(a) Gear Box properties in Functional View. The property *TranslationRatio* (red) was changed in this use case.



Filter action: All
Filter category: All

15.07.2021 15:31:20 by MDRE-System

Action
Model updated
Category
Other
Comment

Change by property propagation from model with name FunctionalView.

Change was triggered by User Christian Engelbrecht

Changes:

GearBox.TranslationRatio: 1:500 --> 1:800

(b) Model History in the requirements view showing external changes

Figure 6.4: Screenshots showing property propagation between related models in the GCM editor.

6.2 Capability 2a: Linking of Model Elements to Engineering Artefacts

Using the GQM approach, the goals of this use case can be outlined as following.

This use case analyses linking of nodes, building a Product-Process-Resource (PPR) Asset Directory. This is done for the purpose of improving traceability and transparency between the modelled Engineering Artefacts. This section conducts the use case from the view point of engineers that aim to improve the traceability and linking of artefacts in their model.

6.2.1 Test cases

Table 6.2 outlines one test case used for the evaluation of this use case.

Test case number 1 Test case number 1 focuses on linking nodes to build a PPR Asset Directory. For this, Engineering Artefacts get linked to an existing model, building

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Nr.	Name	Relates to
UC2.1	Link Nodes for traceability	RQ.UC2a.1, RQ.UC2a.2, RQ.UC2a.3

Table 6.2: Test cases used for evaluation of UC2a

links to the existing model. This allows tracing model elements to Engineering Artefacts via the added links.

6.2.2 Use case execution

Relating to the RUCM steps outlined in Table 4.2, this use case portrays the step of linking Engineering Artefacts to PPR Assets in the model.

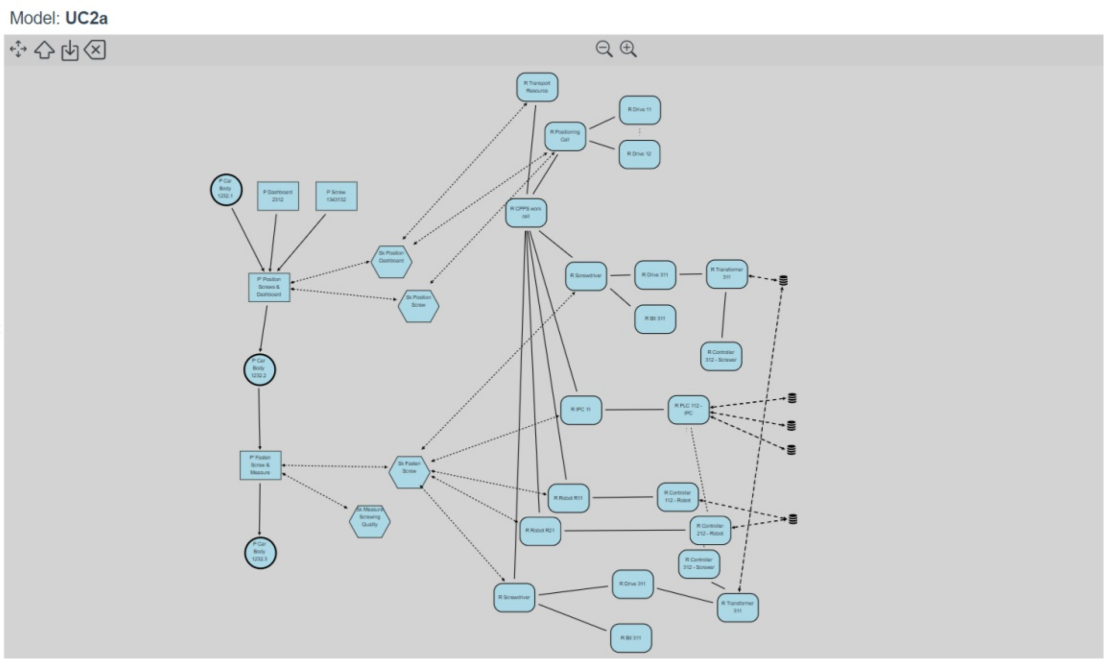


Figure 6.5: PPR Asset Directory modelled in the GCM Editor. Based on mock-up in Figure 4.4.

This is shown in Figure 6.5. Figure 6.5 derives from the paper mock-up in Figure 4.4, by leveraging the import/export function of the GCM Editor via CSV. This allowed to easily import all nodes. To model the links between nodes, this use case uses the GCM Editor Frontend. On the left-hand side of the model, the GCM Editor Frontend shows existing linked PPR Assets. On the right-hand side, Figure 6.5 introduces Engineering Artefacts, depicted by a black icon. As is visible, these Engineering Artefacts are then linked to the existing PPR Asset model via links, modelled via dashed bidirectional arrows.

These links now allow stakeholders to relate existing PPR Assets to Engineering Artefacts,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Nr.	Name	Relates to
UC2b.1	Manipulate properties on model elements	RQ.UC2b.1, RQ.UC2b.2

Table 6.3: Test cases used for evaluation of UC2b

building a PPR Asset Directory in the process.

6.2.3 Advanced Application Options

The basic concept of this use case is the linking of nodes to each other, to convey information. This concept can be applied to different use cases, and is not restricted to PPR Asset Directories. The following section proposes an augmentation of this use case that stores properties with each node, to convey even more information and to allow modelling of more complex real world scenarios.

6.3 Capability 2b: Model Element Properties

Using the GQM approach, the goals of this use case can be outlined as following. These goals relate closely to UC2a, since UC2b is an augmentation of the existing approach in UC2a.

This use case analyses the linking of nodes, each containing properties and information, to build a PPR Asset Network (PAN). This is done for the purpose of improving traceability and transparency between the modelled Engineering Artefacts and storing additional information. This section conducts the use case from the view point of engineers that aim to improve the traceability and linking of artefacts in their model.

6.3.1 Test cases

Table 6.3 outlines one test case to evaluate UC2b.

Test case number 1 Since UC2a outlines the linking of model elements, this use case focuses on storing additional data with each model element, resulting in a PPR Asset Network (PAN). This test case evaluates the flexible definition and manipulation of properties.

6.3.2 Use case execution

To allow for a definition of properties on model elements, these properties have to be defined in the DSL, as outlined in Chapter 5.

Listing 6.1 outlines such a property definition. The DSL defines each required property explicitly. In the provided snippet, this is done exemplary for the model element *Engineering artefacts*, but can be applied to any other model element in the GCM Editor.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
{
      "typeName": "Engineering artefacts",
      "attributeTypes": [
        {
          "variableName": "type",
          "displayName": "Type"
        },
        {
          "variableName": "ea_type",
          "displayName": "ea_type"
        },
        {
          "variableName": "storage_location",
          "displayName": "storage_location"
        },
        {
          "variableName": "note",
          "displayName": "Note"
        }
      ]
    }
```

Listing 6.1: Definition of properties for UC2b in the DSL. Some properties omitted for brevity.

Figure 6.6 shows the defined properties in the GCM Editor Frontend. The left-hand side shows the editor canvas and show the selection of an Engineering Artefact. The right-hand side shows each defined property and can be manipulated. The GCM Editor stores any change to these property values and shows them to other engineers working on the same model. As introduced in Chapter 4, this is the main difference between Figure 6.6 and Figure 6.7. UC3b in Figure 6.7 allows the definition of properties on nodes and links, which is not possible in UC3a. This allows to build a PAN and overall improve traceability in models, while being able to model more complex real-world scenarios by using properties on nodes.

6.3.3 Advanced Application Options

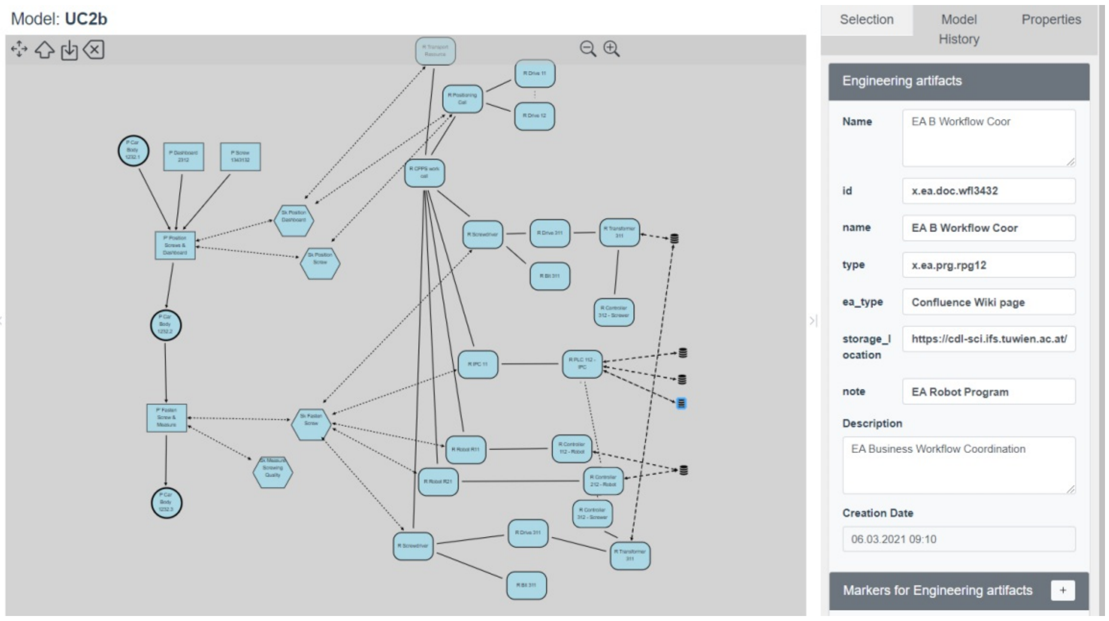
The same outlooks, that are outlined for UC2a, are also applicable to this use case. In general, any real-world use case that would profit of modelling of nodes and links and storing information to each of these model elements, are applicable to the use case. This is also a result of the architecture chosen for the GCM Editor that allows a flexible definition of properties via a DSL as configuration language.

6.4 Capability 3a: Model Layers by Model Element Type

Using the GQM approach, the goals of this use case can be outlined as following.

This use case analyses the representation of nodes on different layers. This is done for the purpose of improving coordination and cooperation, while adding additional knowledge

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



Name: EA B Workflow Coor
id: x.ea.doc.wfl3432
name: EA B Workflow Coor
type: x.ea.prg.rpg12
ea_type: Confluence Wiki page
storage_location: https://cdl-sci.ifs.tuwien.ac.at/
note: EA Robot Program
Description: EA Business Workflow Coordination
Creation Date: 06.03.2021 09:10

Figure 6.6: PPR Asset Network (PAN) modelled in the GCM Editor. Based on mock-up in Figure 4.4 and Figure 6.5.

Nr.	Name	Relates to
UC3a.1	Adding layers to DSL	RQ.UC3a.1
UC3a.2	Toggle layers to declutter editor	RQ.UC3a.2

Table 6.4: Test cases used for evaluation of UC3a

to the models. This section conducts the use case from the view point of CPPS engineers that aim to represent additional information in their models.

6.4.1 Test cases

Table 6.4 shows the two test cases used for the evaluation of this use case.

Test case number 1 As is outlined in Chapter 5, the proposed approach allows the flexible definition of additional layers via the DSL. To outline this, test case 1 aims at representing this possibility.

Test case number 2 Based on these layers, the GCM Editor must allow the toggling of additional layers, to avoid cluttering of the GCM Frontend to the user. This is outlined by test case 2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

6.4.2 Use case execution

To use additional layers in the GCM Frontend, new layers need to be defined in the DSL.

To achieve this, the corresponding layer needs to be added to the concerned model elements. This is shown in Listing 6.2.

```
{
  "modelType": "4—Color—Printer",
  "version": "1.0.0",
  "nodeTypes": [
    {
      "connectPoints": [],
      "insidePlacementRules": [],
      "typeName": "Risk",
      "attributeTypes": [],
      "display": "<svg width='50' height='50'></svg>",
      "layer": "Risk"
    }
  ],
  "edgeTypes": [],
  "markerTypes": [],
  "active": true,
  "layers": [
    "General",
    "Risk"
  ]
}
```

Listing 6.2: Defined layer in the DSL for the *Risk* node. Some properties omitted for brevity.

When the user defines a new layer in the underlying DSL, it is then shown in the GCM Editor Frontend, as is visible in Figure 6.7a. All existing layers in the DSL are shown on the left-hand side of the editor, see the red rectangle with Tag 1.

As outlined by test case number 2, these layers need to provide the possibility of being toggled in the frontend. Figure 6.7 shows this possibility. When a layer gets toggled in the GCM Editor Frontend, the defined nodes are not visible to the user. This is outlined by the blue rectangles and Tag 2 in Figure 6.7b.

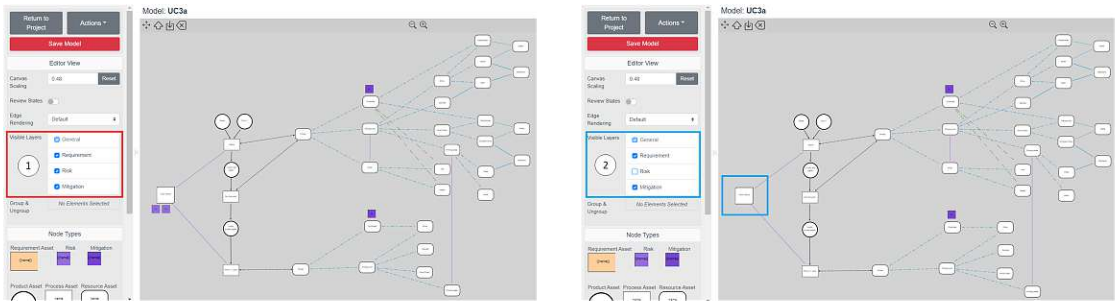
6.4.3 Advanced Application Options

The possibility to define layers and make parts of the model invisible to the users can be applied freely to many use cases. Due to the flexible definition of layers for each model element type in the DSL, even each node type could be placed on its own layer, without limiting the user in the GCM Editor Frontend.

6.5 Capability 3b: Connections between layers

Using the GQM approach, the goals of this use case can be outlined as following. UC3b is closely related to UC3a, which is why some goals are similar.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) All defined layers visible in the GCM Editor Frontend and toggleable.

(b) Risk layer toggled, resulting in visibility changes in the GCM Editor.

Figure 6.7: Figures showing the toggling of layers in the GCM editor.

Nr.	Name	Relates to
UC3b.1	Link nodes across layers	RQ.UC3b.1, RQ.UC3b.2

Table 6.5: Test cases used for evaluation of UC3b

This use case analyses the representation of nodes on different layers and linking nodes across different layers. This is done for the purpose of improving coordination and cooperation, while adding additional knowledge to the models. It further aids traceability by linking nodes to additional information, represented on additional layers. This section conducts the use case from the view point of CPPS engineers that aim to represent additional information in their models and improve the resulting quality of models.

6.5.1 Test cases

Table 6.5 shows the test case used for the evaluation of this use case. Requirement *RQ.UC3b.3*, outlined in Section 4.4.5, will be covered in the evaluation of UC4a and UC4b.

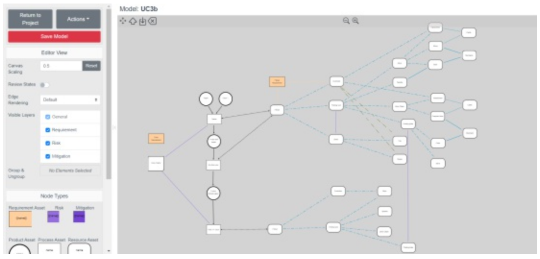
Test case number 1 This test case is closely related to the test cases used for UC3a. This test case augments the approach, by linking nodes across different layers. This allows to improve traceability, e.g., by linking requirements to model parts on an additional layer.

6.5.2 Use case execution

As with UC3a, layers have been modelled on different nodes. But this time, links have been modelled across layer boundaries, e.g., linking the *Printer Requirement* node to the *Printer* node on the "General" layer.

As is visible in Figures 6.8, these links are visible in the GCM Editor, but can also be toggled if needed, which also makes the links invisible. This allows users to model a

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub


(a) All layers visible in the GCM Editor, with linked nodes on different layers.

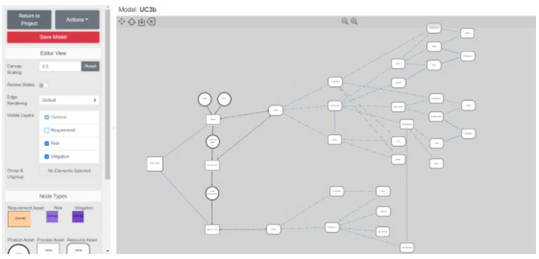

(b) Toggled Requirement layer, resulting in visibility changes.

Figure 6.8: Screenshots showing toggling of layers in models in the GCM editor.

Nr.	Name	Relates to
UC4a.1	Define DSL with Markers and initiate Task.	RQ.UC4a.1
UC4a.2	Cycle through marker states.	RQ.UC4a.2

Table 6.6: Test cases used for evaluation of UC4a

second, overlaying model, consisting of additional information that allows improving the underlying model. The disabling of layers also allows decluttering the GCM Editor Frontend, which allows users to focus better on their layer.

6.5.3 Advanced Application Options

As with UC3a, many applications are possible, consisting of designing additional layers and linking these to an existing model. The possibility to link nodes across layer boundaries allows to build a more closely intertwined overlaying model, which aids traceability and allows modelling of relationship between models on different layers.

6.6 Capability 4a: Coordination with Markers – Manual Setting of Markers

Using the GQM approach, the goals of this use case can be outlined as follows.

This use case analyses the coordination and cooperation for guided tasks using stigmergic approaches via markers. This is done for the purpose of improving coordination and cooperation between engineers, by using the environment as an asynchronous coordination tool. This section conducts the use case from the view point of CPPS engineers that aim to augment their existing model, by creating tasks that can be conducted by other CPPS engineers.

6.6.1 Test cases

Table 6.6 shows the two test cases used for the evaluation of this use case.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Test case number 1 Test case number 1 focuses on the definition of corresponding markers in the DSL, and the initial setup of the use case. For this, the DSL defines three marker types, a *Priority Marker*, a *Task Marker*, and a *Review Marker*. For this test case, the *Priority Marker* and *Task Marker* will be used.

Test case number 2 To guide the tasks via stigmergic coordination, this test case uses different marker states. These marker states convey information CPPS Engineers, that work on resolving the tasks.

6.6.2 Use case execution

To set new markers on model elements in the GCM Editor Frontend they have to be defined in the DSL. This has been introduced in Chapter 5. Listing 6.3 shows a representative definition of a marker. This defined marker has three states, namely *TODO*, *In Progress* and *Done* and represents the states of a task. This marker can be used to coordinate work between multiple CPPS Engineers that pick up tasks in the model.

```
{
  "typeName": "TaskMarker",
  "markerStates": [
    {
      "stateName": "TODO",
      "display": "<svg width='40' height='40'></svg>"
    },
    {
      "stateName": "In Progress",
      "display": "<svg width='40' height='40'></svg>"
    },
    {
      "stateName": "Done",
      "display": "<svg width='40' height='40'></svg>"
    }
  ]
}
```

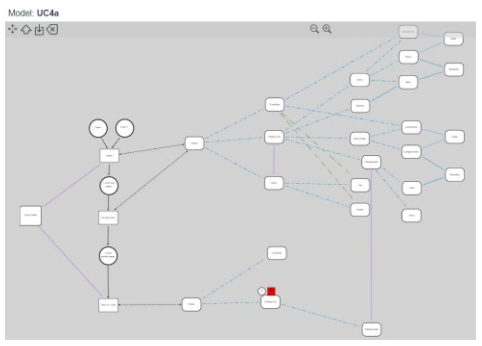
Listing 6.3: Representative defintion of one marker in the DSL.

Based on this marker definition, a CPPS Engineer sets a TODO Marker on a node in the model. This is shown in Figure 6.9a. Furthermore, due to the architecture of the GCM Editor, the CPPS Engineer can add a short description to the marker, e.g., *Task: Design the remaining parts of the printing plate.*

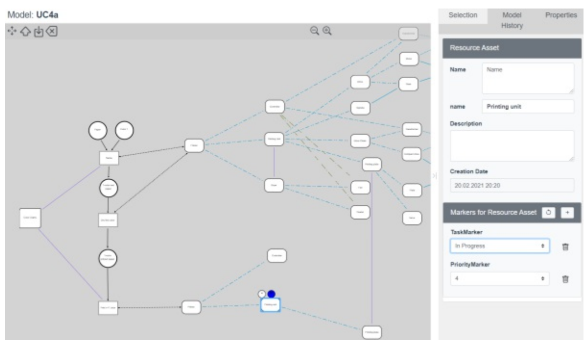
This is visible in Figure 6.9b. Here, a CPPS Engineer picked up a task, and set the state to *In Progress*. This is visible to all other engineers working on the model, which leads to them not picking up the task and focusing on other work in the model. This leverages the approach of stigmergic coordination using the modelling canvas of the GCM Editor as environment.

When the CPPS Engineer finishes a task he sets the marker state to *Done*, as shown in Figure 6.9c. This signals to other engineers the conclusion of that the task, and other

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

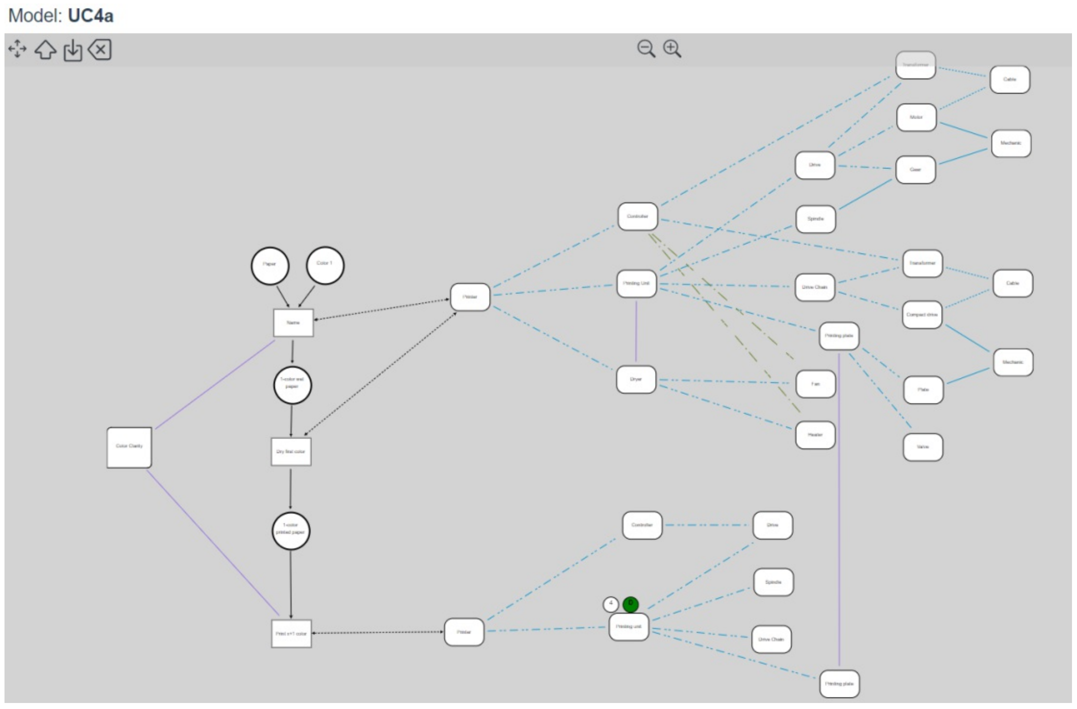


(a) TODO Marker with Priority 4 set on *Printing unit* node.



Name: Printing unit
Description:
Creation Date: 20.02.2021 20:20
TaskMarker: In Progress
PriorityMarker: 4

(b) CPPS Engineer picked up task, state change to In Progress.



(c) Task finished, Done marker set.

Figure 6.9: Usage of markers and marker states on models.

work can be started, e.g., reviewing these new model parts. UC4b evaluates this follow-up task.

6.6.3 Advanced Application Options

The approach of markers is very flexible in the GCM Editor. Markers can be used in any use case where the need for asynchronous coordination via the environment exists.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Nr.	Name	Relates to
UC4b.1	Export model to Neo4J.	RQ.UC3b.3, RQ.UC4a.3, RQ.UC4b.1
UC4b.2	Set new markers semi-automatically.	RQ.UC4b.1, RQ.UC4b.2

Table 6.7: Test cases used for evaluation of UC4b

Furthermore, it allows storing information specifically to a node, which can be leveraged in other, non CPPS specific, approaches.

6.7 Capability 4b: Coordination with Markers – Program-Based Marker Setting

Using the GQM approach, the goals of this use case can be outlined as following. UC4b is closely related to UC4a, also leveraging the marker approach. Therefore, some goals are similar between the two use cases.

This use case analyses the coordination of feedback using stigmergic approaches via markers. This is done for the purpose of improving coordination between engineers, aiming at improving the overall model quality. This section conducts the use case from the view point of CPPS engineers that aim to collect feedback on their model.

6.7.1 Test cases

Table 6.7 shows the two test cases used for the evaluation of this use case.

Test case number 1 Test case number 1 outlines the export of models from the GCM Editor and the import to Neo4J. This approach allows round-trip engineering as outlined as State of the Art in Chapter 2. Furthermore, it allows external manipulation of the model.

Test case number 2 Test case number 2 focuses on using a Graph Query Language (GQL), specifically Cypher in Neo4J, to set markers semi-automatically. This result has then to be imported from Neo4J back to the GCM Editor, to allow for a common representation.

6.7.2 Use case execution

First, the model used in UC4a, containing the *Done* marker, is exported from the GCM Editor and imported to Neo4J. The resulting export can be found in the documenting repository [3]. The representation of the model in Neo4J is documented in Figure 6.10. The Figure shows the representation of the model by Neo4J. Neo4J assigns every node

[3]Model Configuration Repository: `https://bitbucket.org/tuw-qse/mdre-thesis-cengelbrecht`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

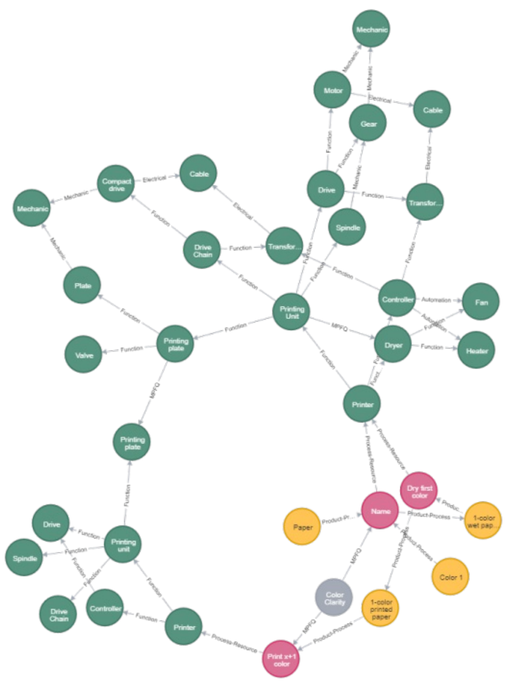


Figure 6.10: Result of C4b represented as a Neo4J graph[2]

type a specific colour automatically and stores properties set in the GCM Editor to each node. This now allows the GCM Engineers, to use Cypher to semi-automatically set new markers on the model. For this use case the query in Listing 6.4 is used. It finds the node with the marker *Done*, and sets an *In Review* Marker on every node, that is one hop away via a link. This is useful, since UC4a manipulated or affected these nodes.

```
MATCH (n)-[*1]-(m) WHERE n.mdreMarkerStateTaskMarker = "Done" SET m.
    mdreMarkerStateReviewMarker="In Review" RETURN n,m
```

Listing 6.4: Setting of markers via Neo4J Cypher Query.

The GCM Editor allows a reimport of these newly set markers. Figure 6.11 shows this. Using these markers, other CPPS Engineers can now review the nodes that are marked via the *In Review* marker.

6.7.3 Advanced Application Options

Readers can leverage the proposed approach in different directions. Since Cypher as GCM allows for many graph-based queries, e.g., path queries, different applications scenarios are possible. Together with the GCM Editor as model representation and manipulation tool, it allows for different application scenarios.

[2]`https://neo4j.com/developer/get-started/` Last accessed: 15/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

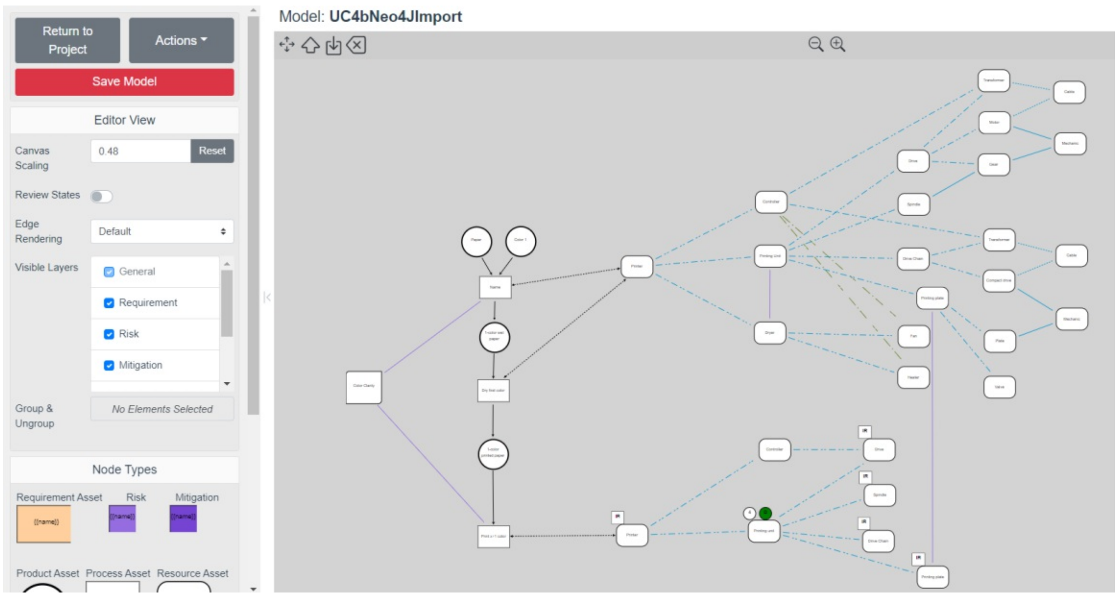


Figure 6.11: UC4b with *In Review* Markers set after import from Neo4J.

6.8 Overall Evaluation Results

The conclusion of this chapter summarizes the overall evaluation. To aid this, Table 6.8 shows which UC addressed which research question and overall requirement in Chapter 3. Table 6.8 further shows that each Requirement and each Research Question in Chapter 3 is addressed in the evaluation of this thesis.

Table 6.8 shows that UC1 and UC4 focus strongly on the evaluation of coordination in a parallel and iterative engineering in the field of CPPS Engineering. In comparison, UC2 and UC3 focused strongly on the graph and data representation and manipulation part in the GCM Editor. Overall, by nature of the research questions and global requirements, all UCs are involved in some way with RQ1 and RQ3, since the Coordinated Model Analysis and Improvement (CMAI)-DSL is used for definition of the models and the resulting software architecture of RQ3 - the GCM Editor - is used as evaluation platform. Furthermore, UC2 and UC3 provide the foundation for UC4. Therefore, the two UCs relate partly to Requirement R2 and RQ2, by laying the groundwork needed for conducting and evaluating UC4.

6.9 Reproducibility of Evaluation Results

To allow for reproducibility of the conducted evaluation, three repositories provide online resources to the interested reader.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

	R1	R2	R3	RQ1	RQ2	RQ3
UC1	X	X		X	X	X
UC2	X	(X)		X	(X)	X
UC3	X	(X)		X	(X)	X
UC4	X	X	X	X	X	X

Table 6.8: Table showing the addressed requirements and research questions, as outlined in Chapter 3, by each UC.

Frontend Repository Code defining the GCM Editor Frontend, can be found in an external repository[4]. It contains all necessary information for a local installation.

Persistence and Backend Repository Code defining the GCM Editor Backend, can be found in an external repository[5]. It also contains all necessary information for a local installation and is necessary when using the GCM Editor Frontend, to allow for storage of the models.

DSLs and models To allow a reproduction of each UC, a repository [6], containing the DSLs and models for each UC, is provided. This allows the interested reader to use the DSLs and models, to reproduce and reevaluate the results of this thesis.

[4]GCM Editor Frontend Repository: `https://bitbucket.org/tuw-qse/mdre-frontend`

[5]GCM Editor Backend Repository: `https://bitbucket.org/tuw-qse/mdre-backend`

[6]Model Configuration Repository: `https://bitbucket.org/tuw-qse/mdre-thesis-cengelbrecht`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 7

Discussion

The goal of this chapter is to tie together the results of this thesis and discuss the relation to the research questions. Furthermore, it outlines limitations of the proposed solutions. For this, this chapter connects each research question, outlined in Chapter 3, to the evaluation and discusses the fulfilment of the research question.

This chapter also treats the *threats to validity* [Höst et al., 2000] that may arise from the solution approach of this thesis.

7.1 RQ1: Domain Specific Lanuage for Coordination

RQ1. CMAI-DSL. What domain-specific language (DSL) allows representing and linking of multi-view graphs for coordinated multi-disciplinary engineering? RQ1 aims at the visual and data representations of models in a multi-disciplinary domain. For this, a Domain-Specific Language (DSL) is necessary, that allows not only the modelling of graphs, but also the linking of models in between them.

This thesis proposes the Coordinated Model Analysis and Improvement (CMAI)-DSL, as introduced in Chapter 5. The CMAI-DSL allows defining the visual representation of nodes and links, resulting in a graph. It also allows the definition of data properties, that can be stored and manipulated. Therefore, the CMAI-DSL allows the representation of engineering models.

To answer the coordinated, multi-disciplinary aspect of RQ1, the CMAI-DSL allows linking of models. A unique ID identifies every model, which then allows building links between models. These links are especially relevant in the method proposed by RQ2.

The CMAI-DSL has been used in all evaluated use cases in Chapter 6. Furthermore, the resulting artefacts can be found in the documentation repository of this thesis [1].

[1]Model Configuration Repository: https://bitbucket.org/tuw-qse/

TU WIEN Bibliothek Your knowledge hub

The proposed solution using the CMAI-DSL leverages existing approaches for model representation, manipulation and storing, as outlined in the state of the art in Chapter 2 [Biffl et al., 2017, 2019b; Lüder et al., 2019; Lüder et al., 2020]. Furthermore, the proposed solutions separate itself from Model Driven Systems Engineering (MDSE) and the Eclipse Modeling Framework (EMF), by providing a more lightweight and modern experience. A further difference is the focus on Cyber-Physical Production Systems (CPPS) Engineering, while MDSE strongly focuses on Software Engineering, while providing models that can be directly translated to code [Brambilla et al., 2012].

This research question goes beyond the state of the art introduced in Chapter 2 as it focuses on an approach, that allows the linking and propagation of data properties in models, while still providing a separate visual representation for each discipline in the multi-disciplinary context. It also provides a way of synchronously keeping models consistent between models, allowing each stakeholder to work on the most recent data. For the data propagation, the links between CMAI-DSLs are leveraged.

This adds new knowledge to the state of research by providing a way of synchronously keeping models up to date in a parallel, iterative, multi-disciplinary engineering approach. Overall, this reduces risks in the production process and improves the quality of graphs in Cyber-Physical Production Systems (CPPS) Engineering by providing up-to-date knowledge and data for all engineering disciplines.

Limitations. The proposed solutions in Chapter 6 focus solely on real-world use cases that can be represented in a graph using nodes and links. This means, that the CMAI-DSL can only model in two dimensions. This limits the application scenarios, compared to more sophisticated modelling tools used in the CPPS context, e.g., AutoCAD [2].

Furthermore, the CMAI-DSL introduces one more data structure in the field of CPPS Engineering, that is already very heterogeneous due to its multi-disciplinary nature. To alleviate this limitation, the proposed Generic CPPS Modeling (GCM) Editor allows import and export to different data formats, as outlined in Chapter 5. This interface allows to transform the stored models and integrate the GCM Editor into the multi-disciplinary CPPS Engineering field.

7.2 RQ2: Method for Coordinated Analysis and Improvement

RQ2. CMAI method. What method can domain experts use to define *design and validation loops* to analyse and improve multi-view graphs, with links between models in different engineering disciplines? RQ2 focuses on providing domain experts with a method to conduct *design and validation loops* on engineering models. In the current state of the art, the possibility to conduct *design and validation*

mdre-thesis-cengelbrecht

[2] `https://www.autodesk.de/products/autocad/overview` Last accessed: 05/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

loops is limited by the existing tools, as outlined in Chapter 2 [Omicini et al., 2004; Biffl et al., 2016]. To improve on the state of the art, this thesis provides two solution approaches, that can be integrated with each other into the CMAI method.

First, the concept of links, as already outlined in RQ1, improves on the state of the art by providing domain experts with an automated approach to keep models up-to-date seamlessly, regardless of engineering discipline and underlying DSL. This helps domain experts in defining *design and validation loops*, by providing each domain expert with the current underlying model element properties and avoiding working on stale information. This propagation of properties continues through the whole design cycle of a CPPS Engineering model, which leads to a loop that repeats itself. Section 6.1 evaluates this propagation of properties and the resulting loops.

As a second approach to analyse and improve engineering models, this thesis proposes an approach using coordination markers by leveraging Stigmergy. For this, the concept of markers has been introduced in Chapter 5. Using markers and their states, a *design and validation loop* can be defined. This approach has been evaluated in Section 6.6 and Section 6.7. As is evaluated, engineers can asynchronously analyse and improve models using markers for coordination. This approach goes beyond the state of the art by allowing engineers to use the environment, e.g., the GCM Editor canvas, as coordination space. This improves on using external tools, e.g., Jira [3], and allows a more fine-grained approach, setting markers on model elements directly.

Combining these two approaches allows to easily conduct a design and validation loop, leading to the CMAI method. This method improves the state of the art by providing a method that can be applied on existing multi-view graphs in the field of CPPS Engineering, aids engineers in conducting design and validation loops and overall improves the quality of the analysed and improved graphs.

Limitations. The proposed approaches work with a wide variety of use cases, but do have some limitations.

The concept of links between CMAI-DSLs, allow a very flexible linking of models. The limitation of using these proposed links, is that model elements representing the same real world artifact need to have the same unique identifier in their respective model. To overcome this, the proposed solution approach allows importing of model elements from different models, which already have the same unique identifier set. Chapter 6 evaluates this use case.

Using markers, they allow the definition of a set of states that can be represented. This is enough for most basic use cases, but is limited by not allowing domain experts to define stricter rules on how marker and their states can be used. In state of the art issue tracking tools, such as Jira [4], more complex task states and rules can be defined.

[3]`https://www.atlassian.com/de/software/jira` Last accessed: 05/09/2021
[4]`https://www.atlassian.com/de/software/jira` Last accessed: 05/09/2021

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

7.3 RQ3: Editor and Service Architecture for Coordinated Model Analysis and Improvement

RQ3. CMAI Editor and Service Architecture. What software architecture and solution design can provide the capabilities to efficiently analyse and improve models? To further improve the possibilities for domain experts to analyse and improve models, software tool support is desirable. In the current state of the art, this software tool support only exists to a very limited amount.

First, this thesis proposes a software architecture, allowing the usage of the CMAI-DSL and the visual representation and manipulation of graphs defined by the CMAI-DSL. As outlined in Chapter 2, state of the art approaches in the field of CPPS Engineering introduce DSLs such as Automation Markup Language (AML). Furthermore, approaches for round-trip engineering exist in the state of the art. To pick up on this, this thesis introduced model transformation using import and export from the GCM Editor in the solution approach, allowing the transformation of models defined by the CMAI-DSL.

This thesis then leveraged the import and export of the usage of Neo4J and Cypher as Graph Query Language (GQL), to aid the analysis and improvement of (big) models. This is achieved by allowing domain experts to export models from the GCM Editor, using the CMAI-DSL, to Neo4J. Using Cypher, domain experts can then efficiently query big models. Furthermore, using the coordination markers introduced in this thesis, domain experts can mark results of their queries, e.g., for further human-based analysis, using Neo4J. The GCM Editor allows an import of the results from using the software tool support, e.g., Neo4J. Modification of properties, new markers, etc. are then shown in the GCM Editor Frontend. This has been evaluated in Section 6.7.

The improvements introduced by this software tool based approach are diverse. Due to the strengths of query languages in general, a big amount of data can be handled in very short time. This allows domain experts, to analyse and improve models, regardless of their size. From a certain point, humans alone are unable to analyse and review models efficiently, if at all. This allows domain experts to handle bigger models, but also reduce their time needed for the task. This freed up personnel can then conduct more human-based tasks that require human input and cannot be conducted by software tools.

Overall, this leads to an improvement in the state of the art, by providing a software architecture that can be combined with existing models via an import/export to the GCM Editor. It furthermore improves the effectivity of reviews conducted by domain experts, by reducing the needed time for querying and marking big models, by providing a software architecture that allows a semi-automatic approach using GQLs.

Limitations. Limitations of this solution approach mainly concern the integration of GQLs with the GCM Editor. Although the import and export approach to Neo4J has proved sufficiently robust for the evaluated use cases, it does not work seamlessly, which might increase the difficulty of usage for domain experts. Furthermore, the data format

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

of the CMAI-DSL has to be mapped onto the data format of Neo4J, which requires model transformation and adds more complexity to the already very heterogeneous field of CPPS Engineering.

To overcome this, future work can be done on integrating Cypher as GQL into the GCM Editor. This would allow conducting analyses directly from the GCM Editor, increasing ease of use for domain experts.

7.4 Limitations

The following limitations apply to the results of this thesis:

Property propagation on model element level. The proposed property propagation between models defined in the CMAI-DSL only allows propagation on model element level. This means that the system only recognizes model elements that have the same unique identifier as representing the same artefact, and only propagates properties if they have the same name. This means that properties that the system should propagate need to have the same name, which has its limitations in real-world scenarios, if different engineering disciplines use different names for properties.

No rule set for marker states in coordination markers. Marker states defined for coordination markers can be cycled through in any arbitrary order by engineers. In a well-defined process between engineers of one engineering discipline, this may not lead to problems, but it would be preferable to limit the allowed state transitions between marker states. This would allow the system to enforce following certain review or task steps instead of allowing arbitrary state changes.

Use cases focus only on CPPS Engineering as application field. The selected use cases and the application field of this thesis is strictly limited to CPPS Engineering. This focuses the results of this thesis to CPPS Engineering, which may introduce a bias into the finding of this thesis.

To address these limitations, this thesis proposes future work in Section 8.2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 8

Conclusion and Future Work

The last chapter of this thesis summarizes the conducted work and outlines future work, to establish further research that can be conducted based on the results of this thesis.

8.1 Conclusion

This thesis investigated *coordinated multi-view graph analysis and improvement*. The thesis addressed new solution approaches in the field of Cyber-Physical Production Systems (CPPS) Engineering, an application field in need of coordinated behaviour, due to its multi-disciplinary nature.

To achieve this, Chapter 4 outlines illustrative use cases. These use cases focused on different aspects of *coordinated multi-view graph analysis and improvement* in the field of CPPS Engineering. These use cases are a result of the issues investigated by the state of the art in Chapter 2 of this thesis.

Based on the state of the art and resulting illustrative use cases, this thesis identifies two main solution approaches. First, the *model element propagation*, and second, the *coordination markers*.

The *model element propagation* focuses on allowing CPPS Engineers to propagate information to colleagues automatically, regardless of boundaries between models, resulting from different engineering disciplines and modelling approaches.

Second, *coordination markers* allow leveraging stigmergic coordination between engineers, by using traces in the environment for asynchronous coordination. To further simplify the handling of coordination markers, this thesis investigated the semi-automatic placement approaches of markers.

To allow the definition of graphs, consisting of nodes and edges, the Coordinated Model Analysis and Improvement (CMAI)-Domain-Specific Language (DSL) was introduced.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Apart from allowing the configuration of nodes and edges, it also allows defining the *model element propagation* and the *coordination markers*.

To evaluate the proposed solutions, this thesis proposes a technical prototype. This is the Generic CPPS Modeling (GCM) Editor. The GCM Editor, using the CMAI-DSL, was used to evaluate the proposed solution approaches, using the outlined illustrative use cases. This thesis used a qualitative evaluation to investigate the results of the proposed solution approaches.

In total, the *model element propagation* and the *coordination markers* showed promising application fields in the domain of CPPS Engineering. The conducted use cases show that coordination between engineers can be enhanced and aided by computers, allowing each engineer to work on the same underlying data, being synced automatically by the system. Furthermore, analysis and improvement of graphs in multi-disciplinary domains can benefit of using the proposed *coordination markers*. It allows and aids the conduction of analysis and improvements tasks, leveraging stigmergic approaches.

Overall, these approaches allow improving the efficiency of CPPS Engineering. Furthermore, these approaches reduce the effort used for coordination in CPPS Engineering while reducing overall risks, since better coordination between engineering disciplines can take place. In the future, this could be used as foundation for agile CPPS Engineering.

The following results of this thesis address the challenges introduced in Chapter 1.

C1. *Limited analysis and improvement resources.* The novel CMAI method improves the efficiency of *design and validation flows* in Multi-View Graphs. This leads to a better quality in CPPS Multi-View models and reduces the time domain experts require to conduct analysis and improvement tasks on graphs. This addresses the challenge by reducing the resources required for analysis and improvement.

C2. *Large models and expensive modelling tools.* The novel software architecture and tool support introduced in Chapter 5 enable CPPS Engineers to use Graph Query Language (GQL) to semi-automatically set markers on large models. This capability facilitates handling large models, since CPPS Engineers can quickly and efficiently set coordination markers on large graphs. Furthermore, the proposed technical architecture is easily accessible and free of cost, tackling the challenge of expensive modeling tools.

C3. *Limited approaches for coordinating analysis and improvement processes.* This thesis introduced the CMAI-DSL. The CMAI-DSL facilitates the propagation of properties between models. Furthermore, the CMAI-DSL enables the definition of coordination markers. These coordination markers provide capabilities for coordinating analysis and improvement processes by allowing the tracking of task states on graphs in an integrated manner.

Following up on the stakeholders and readers introduced in Section 1.4, each group of readers can pick up different results of this thesis.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

(1) **Practitioners** in the field of CPPS Engineering can use the GCM Editor, to model their models. This could include change propagation and coordination markers, leading to risk reduction and an overall more efficient CPPS Engineering progress. To pick up the results of this work, this would mean that a new role of, e.g., a model curator would need to be introduced, to define models using the CMAI-DSL.

(2) Readers working in **management positions** could pick up the results of this work to track and manage progress of the CPPS Engineering progress. For this, the introduced coordination markers could prove especially useful.

(3) **Researchers** can pick up the results of this work and base their further research in the field on it. For this, Section 8.2 proposes future work, to point to further research fields based on this thesis.

(4) **Software Engineers** can pick up the software architecture proposed for the GCM Editor. The reader can see the proposed architecture as detached from a concrete implementation, and a new implementation on a different technological stack might be interesting to Software Engineers. Furthermore, the resulting codebase, documented in a Repository for the GCM Editor Frontend [1] and the GCM Editor Backend [2], can be used for further development and improvement of the GCM Editor.

8.2 Future Work

This section introduces possibilities for future work, based on the results of this thesis.

Enhance property propagation. For the prototype in this thesis, the solution approach restricts property propagation to model elements that are identified by the same unique UUID. Furthermore, only model element properties that have the same name in both CMAI-DSLs are propagated. This may be a limitation in some use cases. To evaluate if a more sophisticated propagation approach also improves the coordination between engineers, this area needs further research.

Improve rule sets of *coordination markers*. The current GCM Editor prototype allows every user to cycle between all state of a *coordination marker*. This allows a great flexibility for users, but might not be intended in every use case.

For this, further research should be conducted, analysing if defining rule sets for *coordination markers* improves coordination and broadens the field of conducted use cases, leveraging this approach.

Integrate CMAI-DSL in state of the art. As introduced in Chapter 2, data formats like Automation Markup Language (AML) are existing approaches in the domain of

[1] GCM Editor Frontend Repository: `https://bitbucket.org/tuw-qse/mdre-frontend`
[2] GCM Editor Backend Repository: `https://bitbucket.org/tuw-qse/mdre-backend`

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CPPS Engineering. To make the results of this thesis more accessible in the domain, a better integration of the CMAI-DSL with existing standards should be strived for. This would allow using proposed solutions by this thesis on already existing models in the domain.

Improve representation for CPPS Engineers. Lastly, as already outlined in Chapter 7, the current prototype of the GCM Editor, only allows the representation of two-dimensional, nodes and edges graphs. This is enough for most use cases, but limits the applications of the GCM Editor.

Future research should investigate if the proposed solutions can be leveraged in existing, three-dimensional modelling tools, or if the CMAI-DSL can be adapted to allow three-dimensional modelling approaches to CPPS Engineers.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

Colin Atkinson, Christian Tunjic, and Torben Möller. Fundamental realization strategies for multi-view specification environments. In *2015 IEEE 19th International Enterprise Distributed Object Computing Conference*, pages 40–49. IEEE, 2015.

Stefan Biffl, Arndt Lüder, and Dietmar Winkler. Multi-disciplinary engineering for industrie 4.0: Semantic challenges and needs. In Stefan Biffl and Marta Sabou, editors, *Semantic Web Technologies for Intelligent Engineering Applications*, pages 17–51. Springer, 2016. doi: 10.1007/978-3-319-41490-4_2. URL `https://doi.org/10.1007/978-3-319-41490-4_2`.

Stefan Biffl, Detlef Gerhard, and Arndt Lüder. Introduction to the multi-disciplinary engineering for cyber-physical production systems. In Stefan Biffl, Arndt Lüder, and Detlef Gerhard, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems, Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 1–24. Springer, 2017. doi: 10.1007/978-3-319-56345-9_1. URL `https://doi.org/10.1007/978-3-319-56345-9_1`.

Stefan Biffl, Arndt Lüder, Felix Rinker, and Laura Waltersdorfer. Efficient engineering data exchange in multi-disciplinary systems engineering. In *International Conference on Advanced Information Systems Engineering*, pages 17–31. Springer, 2019a.

Stefan Biffl, Arndt Lüder, Felix Rinker, Laura Waltersdorfer, and Dietmar Winkler. Engineering data logistics for agile automation systems engineering. In *Security and Quality in Cyber-Physical Sys. Eng.*, pages 187–225. Springer, 2019b.

Stefan Biffl, Arndt Lüder, Elmar Kiesling, Kristof Meixner, Felix Rinker, Christian Engelbrecht, Matthias Eckhart, and Dietmar Winkler. Multi-Aspect Risk Exploration in Models for Positioning and Joining Simulation(Case Study) Part II. Technical report, Technische Universität Wien, 11 2020.

Stefan Biffl, Arndt Lüder, Kristof Meixner, Felix Rinker, Matthias Eckhart, and Dietmar Winkler. Multi-View-Model Risk Assessment in Cyber-Physical Production Systems Engineering. In *8th Int. Conf. on Model-Driven Eng. and Softw. Dev., MODELSWARD 2021*. SciTePress, 2021a.

Stefan Biffl, Kristof Meixner, Jan Herzog, Arndt Lüder, Felix Rinker, Christian Engelbrecht, and Dietmar Winkler. Requirements Tracing in Cyber-Physical Robot Cells for Positioning and Joining(Case Study). Technical Report CDL-SQI-2021-03, CDL-SQI, Institute for ISE, TU Wien, March 2021b.

Stefan Biffl, Kristof Meixner, Dietmar Winkler, and Arndt Lüder and. Towards Efficient Asset-Based Configuration Management with a PPR Asset Directory. In *26th IEEE ETFA 2021, Västerås, Sweden, September 07-10, 2021*, page 4 pages. IEEE, 2021c.

Bundesministerium für Bildung und Forschung BMBF. Industrie 4.0, Dec 2018. URL `https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html`. (Accessed on 1.11.2020).

Marco Brambilla, Jordi Cabot, and Manuel Wimmer. Model-driven software engineering in practice. *Synthesis Lectures on Software Engineering*, 1(1):1–182, 2012.

Victor R Basili1 Gianluigi Caldiera and H Dieter Rombach. The goal question metric approach. *Encyclopedia of software engineering*, pages 528–532, 1994.

Olivier Cardin. Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Computers in Industry*, 104:11–21, 2019.

Brian Dobing and Jeffrey Parsons. How uml is used. *Communications of the ACM*, 49 (5):109–113, 2006.

Rainer Drath, Arndt Luder, Jorn Peschke, and Lorenz Hundt. AutomationML - the glue for seamless automation engineering. Part 6: AutomationML Component. 10 2020.

Emelie Engström, Margaret-Anne Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. How software engineering research aligns with design science: a review. *Empirical Software Engineering*, 25(4):2630–2660, 2020.

Plerre-P Grassé. La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80, 1959.

Ekbert Hering. Der datenflußplan nach din 66001. In *Software-Engineering*, pages 56–62. Springer, 1984.

Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

Hubert F Hofmann and Franz Lehner. Requirements engineering as a success factor in software projects. *IEEE software*, 18(4):58–66, 2001.

Martin Höst, Björn Regnell, and Claes Wohlin. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5(3):201–214, 2000.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

IEC 62714-1. Engineering data exchange format for use in industrial automation systems engineering – automation markup language – part 1: Architecture and general requirements. *Int. Standard, Second Edition, Int. Electrotechnical Commission, Geneva*, 2, 2018.

ISO/IEC 9075. ISO/IEC 9075:2016 Information technology-Database languages. *ISO/IEC*, 2016(E):9075–1, 2016.

ISO/IEC WD 39075. ISO/IEC WD 39075, 2020. URL `https://www.iso.org/standard/76120.html`. (Accessed on 13.11.2020).

ISO/IEC/IEEE 31320-1:2012. Information technology — Modeling Languages — Part 1: Syntax and Semantics for IDEF0. Standard, International Organization for Standardization, Geneva, CH, September 2012.

Ivar Jacobson. *Object-oriented software engineering: a use case driven approach*. Pearson Education India, 1993.

Ivar Jacobson. Use cases–yesterday, today, and tomorrow. *Software & Systems Modeling*, 3(3):210–220, 2004.

Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industrie 4.0. *Wirtschaftsinformatik*, 56(4):261–264, 2014.

Edward A Lee. Cyber-physical systems-are computing foundations adequate. In *Position paper for NSF workshop on cyber-physical systems: research motivation, techniques and roadmap*, volume 2, pages 1–9. Citeseer, 2006.

Ziqi Li. NoSQL Databases. *OSF Preprints*, Jul 2019. doi: 10.31219/osf.io/u5q3y.

Arndt Lüder, Johanna-Lisa Pauly, Konstantin Kirchheim, Felix Rinker, and Stefan Biffl. Migration to automationml based tool chains–incrementally overcoming engineering network challenges. In *4th AutomationML user conference*, 2018.

Arndt Lüder, Johanna-Lisa Pauly, Felix Rinker, and Stefan Biffl. Data exchange logistics in engineering networks exploiting automated data integration. In *2019 24th IEEE Int. Conf. on Emerging Tech. and Factory Automation (ETFA)*, pages 657–664. IEEE, 2019.

Arndt Lüder, Anna-Kristin Behnert, Felix Rinker, and Stefan Biffl. Generating Industry 4.0 Asset Administration Shells with Data from Engineering Data Logistics. In *25th IEEE Int. Conf. on Emerging Tech. and Factory Automation, ETFA, Austria, 2020*. IEEE, 2020.

Richard J Mayer. Idefø function modeling: A reconstruction of the original air force report. *College Station, TX: Knowledge Based Systems*, 1990.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Kristof Meixner, Arndt Lüder, Jan Herzog, Dietmar Winkler, and Stefan Biffl. Patterns for reuse in production systems engineering. In *Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering*, pages 37–44. World Scientific, 2021. doi: 10.18293/SEKE2021-150.

László Monostori. Cyber-physical production systems: Roots, expectations and r&d challenges. *Procedia Cirp*, 17:9–13, 2014.

Angelika Musil, Juergen Musil, and Stefan Biffl. Towards collective intelligence system architectures for supporting multi-disciplinary engineering of cyber-physical production systems. In *2016 1st International Workshop on Cyber-Physical Production Systems (CPPS)*, pages 1–4. IEEE, 2016.

Angelika Musil, Juergen Musil, Danny Weyns, Tomas Bures, Henry Muccini, and Mohammad Sharaf. Patterns for self-adaptation in cyber-physical systems. In *Multi-disciplinary engineering for cyber-physical production systems*, pages 331–368. Springer, 2017.

Juergen Musil, Angelika Musil, Danny Weyns, and Stefan Biffl. An architecture framework for collective intelligence systems. In *2015 12th Working IEEE/IFIP Conference on Software Architecture*, pages 21–30. IEEE, 2015.

Neo4J. Cypher query language, 2020. URL `https://neo4j.com/developer/cypher/`. (Accessed on 7.11.2020).

Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 286–293, 2004.

H Van Dyke Parunak. A survey of environments and mechanisms for human-human stigmergy. In *International workshop on environments for multi-agent systems*, pages 163–186. Springer, 2005.

Douglas T Ross. Structured analysis (sa): A language for communicating ideas. *IEEE Transactions on software engineering*, (1):16–34, 1977.

Thomas Schäffler, Matthias Foehr, Arndt Lüder, and Kristin Supke. Engineering process evaluation: Evaluation of the impact of internationalisation decisions on the efficiency and quality of engineering processes. In *2013 IEEE International Symposium on Industrial Electronics*, pages 1–6. IEEE, 2013.

Alexander Schatten, Stefan Biffl, Markus Demolsky, Erik Gostischa-Franta, Thomas Östreicher, and Dietmar Winkler. *Best Practice Software-Engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen.* Springer-Verlag, 2010.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Miriam Schleipen, Arndt Lüder, Olaf Sauer, Holger Flatt, and Jürgen Jasperneite. Requirements and concept for plug-and-work. *at-Automatisierungstechnik*, 63(10): 801–820, 2015.

Dave Steinberg, Frank Budinsky, Ed Merks, and Marcelo Paternostro. *EMF: eclipse modeling framework*. Pearson Education, 2008.

Guy Theraulaz and Eric Bonabeau. A brief history of stigmergy. *Artificial life*, 5(2): 97–116, 1999.

Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database: a data provenance perspective. In *Proceedings of the 48th annual Southeast regional conference*, pages 1–6, 2010.

Birgit Vogel-Heuser, Thomas Bauernhansl, and Michael Ten Hompel. Handbuch industrie 4.0 bd. 4. *Allgemeine Grundlagen*, 2, 2020.

Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

Danny Weyns, Andrea Omicini, and James Odell. Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems*, 14(1):5–30, 2007.

Dietmar Winkler, Manuel Wimmer, Luca Berardinelli, and Stefan Biffl. Towards model quality assurance for multi-disciplinary engineering. In *Multi-disciplinary engineering for cyber-physical production systems*, pages 433–457. Springer, 2017.

Eric Yu. Agent orientation as a modelling paradigm. *Wirtschaftsinformatik*, 43(2): 123–132, 2001.

Eric Siu-Kwong Yu. *MODELLING STRATEGIC RELATIONSHIPS FOR PROCESS REENGINEERING*. PhD thesis, University of Toronto, 1995.

Eric SK Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering*, pages 226–235. IEEE, 1997.

Eric SK Yu and John Mylopoulos. Understanding "why" in software process modelling, analysis, and design. In *Proceedings of 16th international conference on software engineering*, pages 159–168. IEEE, 1994.

Tao Yue, Lionel C Briand, and Yvan Labiche. A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation. In *International conference on model driven engineering languages and systems*, pages 484–498. Springer, 2009.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Figures

TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Tables

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Listings

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acronyms

AML Automation Markup Language. 15, 16, 35, 43, 68, 98, 103

CEN CPPS Engineering Network. 11, 13, 14, 34, 35, 44, 110

CIS Collective Intelligence Systems. 9, 22, 24–27

CMAI Coordinated Model Analysis and Improvement. xi, xiii, 4–6, 12, 24, 31–35, 40, 56, 61, 63–65, 67, 93, 95–99, 101–104, 110

CPPS Cyber-Physical Production Systems. xi, xiii, xv, 1, 2, 4, 6, 9–13, 15, 17, 22, 25, 27, 29, 31–34, 36, 39–44, 46, 48, 50, 52, 54, 56, 58, 59, 61, 63, 65, 77, 78, 85, 87–93, 96–99, 101–104

CPS Cyber Physical Systems. 9–11, 110

CSV comma-separated values. 15, 73, 75, 76, 115

DSL Domain-Specific Language. xi, xiii, 2, 5, 13, 14, 16, 29, 31–35, 37, 40, 43–45, 47–49, 52, 53, 56, 60, 61, 63–73, 76–78, 80, 83–86, 88, 89, 93–99, 101–104, 110, 115

EMF Eclipse Modeling Framework. 15, 96

GCM Generic CPPS Modeling. xi, xiii, 6, 7, 16, 32, 33, 35, 37, 56, 60, 66–70, 72–76, 78–94, 96–99, 102–104, 111, 115

GDBMS Graph Database Management System. 16

GQL Graph Query Language. 16, 73, 74, 91, 98, 99, 102

GQM Goal Question Metric. 58, 77, 81, 83, 84, 86, 88, 91

I* I-Star. 17, 18, 39–41, 110

ICAM Integrated Computer Aided Manufacturing. 19

IDEF0 Integration Definition 0. 17, 19, 20, 39–43, 110

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

JSON JavaScript Object Notation. 35, 68, 75

MAI Model Analysis and Improvement. 2, 4, 11, 14, 23, 33, 35, 61, 110

MARE Multi-Aspect Risk Exploration. 50

MDSE Model Driven Systems Engineering. 15, 96

NoSQL Not only SQL. 16, 68

PAN PPR Asset Network. 39, 47–54, 83–85, 111, 113

PPR Product-Process-Resource. 39, 44, 46–49, 52, 81–83, 111, 113

RUCM Restricted Use Case Modeling. 17, 19–22, 39, 44–46, 48, 49, 51–54, 56, 57, 59, 60, 78, 82, 113

SA Structured Analysis. 19, 20, 110

SVG Scalabale Vector Graphic. 67, 69, 70, 72

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.