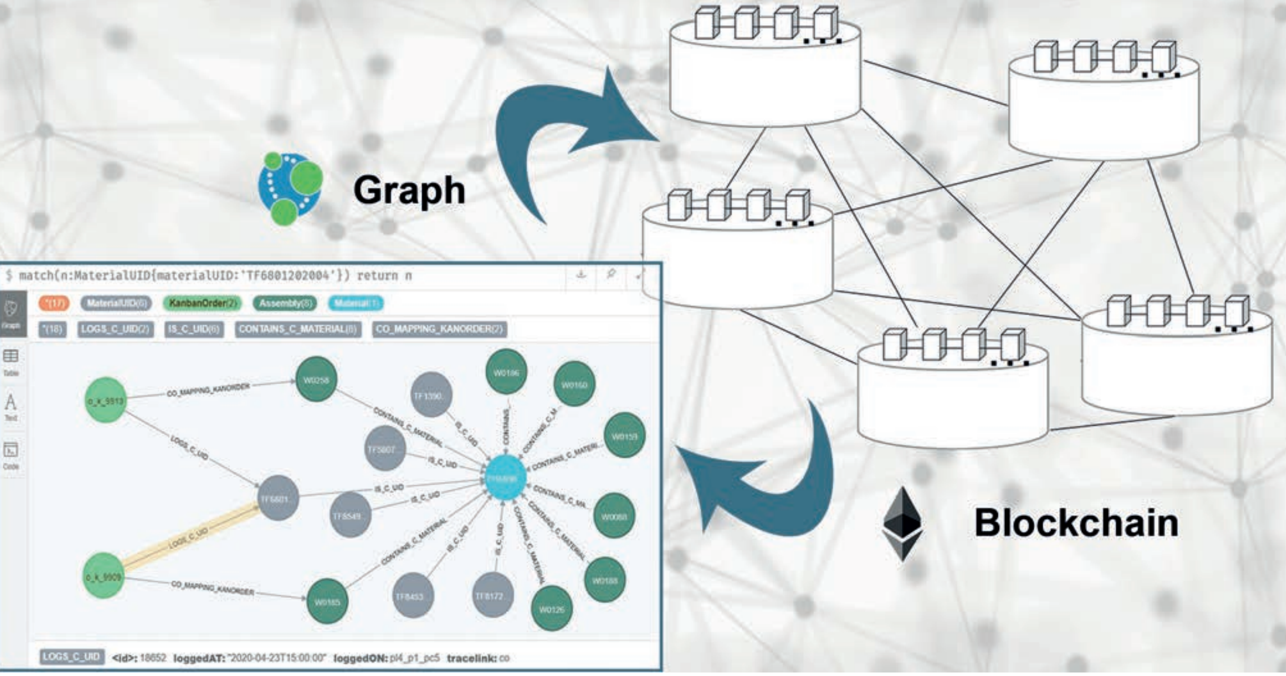


FAU Studien aus dem Maschinenbau 416

Marlene Kuhn

Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs

FAU University Press

Marlene Kuhn

Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs

FAU Studien aus dem Maschinenbau

Band 416

Marlene Kuhn

Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs

Dissertation aus dem Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS)
Prof. Dr.-Ing. Jörg Franke

Erlangen
FAU University Press
2022

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bitte zitieren als
Kuhn, Marlene. 2022. *Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs.* FAU Studien aus dem Maschinenbau Band 416. Erlangen: FAU University Press.
DOI: 10.25593/978-3-96147-616-9.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS-Server der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:
https://opus4.kobv.de/opus4-fau/home

Verlag und Auslieferung:
FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-96147-615-2 (Druckausgabe)
eISBN: 978-3-96147-616-9 (Online-Ausgabe)
ISSN: 2625-9974
DOI: 10.25593/978-3-96147-616-9

Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs

Modellbasierte Entwicklung eines Rückverfolgbarkeitssystems für die komplexe Fertigung mittels Blockchain und Graphen

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur

Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Marlene Kuhn, M.Sc.

aus Lengerich

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	20.09.2022

Gutachter:	Prof. Dr.-Ing. Jörg Franke Prof. Dr. Robert Schmitt, RWTH Aachen

Preface

This dissertation was written during my work as a research associate at the Institute for Factory Automation and Production Systems (FAPS) at the Friedrich–Alexander University Erlangen–Nuernberg (FAU). The work as a research associate and accompanying doctorial candidate was an exciting and educating experience, for which I am very grateful and which left me with many positive memories.

First, I would like to express my special thanks to my doctoral supervisor, Prof. Dr.-Ing. Jörg Franke, for the scientific and professional support during my dissertation. Your endless source of inspiration, innovation, and ideas has allowed me to pursue this goal and overcome the hurdles along the way. I would also like to thank Prof. Dr. Robert Schmitt, Head of Chair of Production Metrology and Quality Management at RWTH Aachen University, for contributing his assessment and acting as second reviewer for this dissertation.

Additionally, I would like to thank my colleagues at the institute FAPS and especially the wire harness research group for the great work atmosphere, the constructive discussions and feedback as well as the friendships that have developed over the years. I would especially like to mention my direct working partners Huong Nguyen, Florian Hefner, Felix Funk and Moritz Meiners. You have turned an otherwise challenging time into a time of exchange, laughter, and development.

I would also express my highest gratitude for my family and friends, who have empowered me to pursue this dream. In this context, I would especially thank my parents Ilona and Karlheinz, my sister Luisa, as well as my best friends Huong, Linda, Nils, Bernd, and Alex for their positive impact on my life and their guidance during this time. Finally I would like to thank my partner Tobias. The late nights of working together, your scientific feedback, and your personal commitment have made this possible.

Nuernberg, October 2021 Marlene Kuhn

Table of Contents

List of symbols and abbreviations

Abbreviation	Short for
AD	Autonomous driving
ANSI	American National Standard Institute
ATM	Assembly token manager
$attr_o$	Trace object attribute
$attr_l$	Trace links attribute
av_o	Trace object attribute-value pair
av_l	Trace link attribute-value pair
B	Batch
BOM	Bill of material
BOP	Bill of process
c	Configuration trace object or link type
CAD	Computer-aided design
CSC	Cut-strip-crimp
DApp	Decentralized web application
DLT	Distributed ledger technology
DPoS	Delegated proof of stake
E	Edge
$\exists!$	Uniqueness quantification
E/E	Electrical and electronic
ECU	Electronic control unit
e.g.	Exempli gratia
EPC	Electronic product code
EPCIS	Electronic product code information service
ER	Entity-relationship
ERC	Ethereum request for comment
ERP	Enterprise resource and planning system
EVM	Ethereum virtual machine
FIFO	First in first out

Abbreviation	Short for
FM	Functional module
G	Graph
Geth	Go Ethereum
G_D	Dynamic graph
G_S	Static graph
GPS	Global positioning system
HV	High-voltage
IATF	International automotive task force
ID	Identifier
IEC	International electrotechnical commission
IIoT	Industrial internet of things
IP	Intellectual property
ISA	International society of automation
ISO	International organization for standardization
JiS	Just in sequence
KBL	Kabelbaumliste
KSK	Kundenspezifischer Kabelbaum
LHD	Left-hand drive
l_{tr}	Trace link
Mat.	Maturity
MES	Manufacturing execution system
min	Minute
ms	Millisecond
NFC	Near field communication
o	Order trace object or link type
OEM	Original equipment manufacturer
O_{tr}	Trace object
O_{tr_SC}	Trace object within the supply chain model
p	Process trace object or link type
P1	Cutting/wire processing

Abbreviation	Short for
P2	Pre-fabrication of modules/special processes
P3	Assembly and testing
P4	Outbound logistics
PBFT	Practical byzantine fault tolerance
PDF	Portable document format
PLM	Product lifecycle management
PM	Production module
PoA	Proof of authority
PoS	Proof of stake
PoW	Proof of work
PPR	Product-process-resource
p_{seg}	Process segment
q	Query speed
r	Resource trace object or link type
RAMI 4.0	Reference architecture model industry 4.0
RDBMS	Relational database management system
RFID	Radio-frequency identification
RHD	Right-hand drive
R_{UID}	Trace reference
SBOM	Structured bill of material
SCM	Supply chain management
s	Second
s_i	Object status
s_n	Most recent object status
s_0	Initial object status
SQL	Structured query language
TA	Trace actor
T_{ltr}	Tracing function
TM	Traceability model
TM_{MFG}	Manufacturing traceability model

Abbreviation	**Short for**
TM_{SC}	Supply chain traceability model
TR	Technical requirement
TRM	Traceability reference model
TRM_{MFG}	Manufacturing traceability reference model
TRM_{SC}	Supply chain traceability reference model
T_S	Tracking function
ty_o	Trace object type
$ty_{o_{set}}$	Set or tuple of eligible trace object types
ty_{li}	Trace link type
$ty_{li_{set}}$	Set or tuple of eligible trace link types
U	Unique referenced c-object
UID	Unique product identifier
UML	Unified modeling language
URI	Uniform resource identifier
U.S.	United States
V	Vertice
VEC	Vehicle electric container
$vinID$	Vehicle identification number
WIP	Work in progress
wh	Wire harness
ws	Workstation
WSN	Wireless sensor network
XML	Extensible markup language

List of figures

List of tables

1 Introduction

Supply chains are becoming increasingly complex and decentralized, encompassing heterogeneous segments, organizational units, and countries of production. In global and distributed manufacturing, several companies contribute to the overall quality of the final product, which requires organizations to collaborate by monitoring their manufacturing flows and exchanging production-critical data [1]. In addition to the organizational challenges, the technological paradigm shifts of smart manufacturing and Industry 4.0 have transformed production systems towards higher exploitation of their data as well as an enhanced perception of the value of data [2]. This value does not solely hinge on the sheer volume of data but is rather based on the depths of insights that can be retrieved from it [2]. The main applications of those insights lie in the quality management of products and processes, such as recording product and process status [3] or analyzing product quality, scrap, or failure rates [2]. Production thus becomes more dependent on extracting quality-driven knowledge from data [4], leading to high efforts to develop effective recording, aggregation, and storage systems. A traceability system, which interconnects and maintains data across domains and manufacturing units, is seen as the main enabler to build a holistic database and derive these quality-driven applications [5]. In complex and distributed manufacturing, traceability systems thus need to encompass detailed manufacturing data as well as a holistic data view over the entire supply chain contributing to the final product.

However, providing those manufacturing networks with appropriate traceability solutions remains a great challenge. This is especially true for customized production systems such as complex mechatronic assemblies of the automotive or avionic industry. In customized manufacturing for a mass market, large and distinctive data histories are generated for each product, placing high demands on the underlying data model and traceability storage system. For those types of industries, existing traceability solutions are highly inefficient or even completely lacking [1]. Nevertheless, with the growing transparency requirements and the introduction of novel data technologies, the demand for suitable traceability solutions for complex manufacturing is now particularly high. In this context, the increasing maturity level of novel storage technologies, such as graph databases and distributed ledger technologies (DLT), also referred to as blockchain, induce new opportunities for traceability solutions in manufacturing.

1.1 Changes in the automotive industry

One key industry, which produces complex mechatronic products in decentralized manufacturing networks, is the automotive industry. The automotive industry has experienced a regulatory push towards higher traceability due to its skyrocketing recall rates over the last few years. In Germany alone, the number of recalls has increased fivefold since 2010 [6]. One reason behind this is the recent increase of technologically advanced vehicles that face the trade-off between rapid market introduction and system reliability [7], which leads to rising failure rates in the field. In this context, large mechatronic systems that perform electrified functions and assistance functions are most affected by quality problems and associated recalls [8]. Consequently, the recall risk and costs rise exponentially for products contributing to autonomous driving (AD) functions and advanced assistance systems [7]. One of these products is the vehicle's electrical and electronic system (E/E system), also called wire harness, which overtakes the critical role of providing the energy and data flow for all the electric components, sensors, actors, and electronic control units (ECUs) in the vehicle. The importance of the wire system industry in the overall automotive supply chain has continuously grown over the last decades. The global market size of automotive wire harnesses is estimated at US$ 81.2 billion for 2018 and is expected to reach US$ 135.2 billion by 2027, whereby the key growth drivers are the general increase in vehicle sales combined with a rise in vehicle function electrification [9]. While wire harnesses used to be a commodity product with failures causing inconveniences, such as a disconnected parking sensor, the increase of autonomous and electrified functions has changed the role of the wire harness, which now determines safety-critical functions like steering or breaking [P1]. The increase of failure severity paired with the large size and complexity of the wire harness has made it one of the heaviest, most quality-critical, and most expensive components in the vehicle [10, P2].

With the shared liability and responsibility for the final product, the vehicle manufacturers thus increasingly demand measures to ensure the safety and quality of all critical components by implementing a transparent traceability system [P1, P2]. In addition to the vehicle manufacturer's pull for traceability, a technological push for traceability can further be observed in the industry. Induced by increasing process digitization and the proliferation of data-driven applications, novel traceability technologies have emerged, which promise to facilitate managing not only internal production data but also data at the interfaces with external supply chain partners. The technological push thus encompasses technologies that ensure the availability of data, such as smart tags, sensors, and industrial internet of things (IIoT) devices, advancements in data storage technologies, such as NoSQL storages, graphs, or distributed ledger technologies (DLTs) and blockchain, as well as the ideation of new processing frameworks, protocols, and data analytics methods.

Consequently, the automotive industry now faces increasing traceability requirements with the opportunity to overcome the current lack of appropriate solutions through novel technologies. Whereby a few automotive sub-industries, such as the production of electronic components, already display a high traceability maturity in manufacturing, the wire harness industry has a very low traceability maturity and lacks adequate solutions. The difficulty in developing suitable solutions not only lies in the technical development but also in the general lack of systematic methodologies and data models for the development of manufacturing traceability. Within the course of this thesis, it will become evident that the wire harness industry is an ideal research object for the development of new traceability methods and solutions for the following reasons:

- **Need for action**: The arising traceability requirements for the wire harness industry are not only representative of the increasing requirements in the automotive sector, but also the changing customer behavior, which requires all kinds of industries to provide transparency and traceability of their operations, impact, and practices throughout their manufacturing and supply chain.
- **Concept:** The lack of existing traceability solutions allows the development of a future-oriented traceability system based on novel data technologies without adhering to legacy systems or existing infrastructures.
- **Transferability**: The proposed systematic methodology for traceability development is industry-independent. Moreover, the characteristics of the wire harness industry are abstracted throughout this thesis so that technological development can be adapted to industries with similar features.

1.2 Research questions and objectives

This thesis focuses on traceability system development for complex manufacturing processes and presents a detailed traceability solution for the representative use case of automotive wire harness manufacturing. To develop the traceability system, the further course of this thesis will demonstrate that three shortcomings of the state of the art need to be overcome, which are a lack of traceability modeling methods in the manufacturing domain (1), deficiencies in available traceability data models (2), and the limitations of conventionally used traceability technologies (3). In alignment with the methodological and technological shortcomings in current traceability research, which will be derived in detail in the theory section, this thesis addresses the following research questions:

RQ1	Are modeling methodologies from other research fields transferrable to systematically develop traceability systems in manufacturing?
RQ2	How does a data model need to be designed to enable tracking and tracing in complex manufacturing?
RQ3	Can graph databases and distributed ledger technologies effectively realize the data model and provide complete traceability in complex manufacturing?

This research provides the research contribution of (1) developing a traceability methodology adapted from the research field of software engineering, which allows to formally model and define traceability-relevant objects, linkages, and functions for manufacturing processes. Moreover, this thesis aims at overcoming the shortcomings of conventional traceability solutions by developing a holistic data model, which encompasses the data domains *product*, *resource*, *process*, and *order* (2). Based on the theory development, the data model for complex manufacturing is specified and the traceability system is implemented through an integrated solution consisting of a graph database and blockchain technology (3). The graph database enables a detailed traceability view within one manufacturing unit, while the blockchain solution delivers a macro perspective on traceability data across manufacturing units.

Throughout this thesis, the wire harness use case serves to verify the proposed method and concept and provides evidence for assessing the research questions. From an industrial and practical perspective, this thesis provides practical contributions by developing a holistic traceability solution for the wire harness industry. Additionally, wire harness producers and their supply chain partners can use this work to adapt their processes towards higher traceability. Moreover, the proposed technologies can be transferred to similar industries.

1.3 Structure of this work

This thesis consists of eight superordinate chapters as visualized in Figure 1. After the introduction (1), the theory section (2) gives an overview concerning the state of the art in wire harness manufacturing, traceability systems, and data-driven technologies. Based on the derived use case requirements and shortcomings of related works, the research framework of this thesis is presented (3).

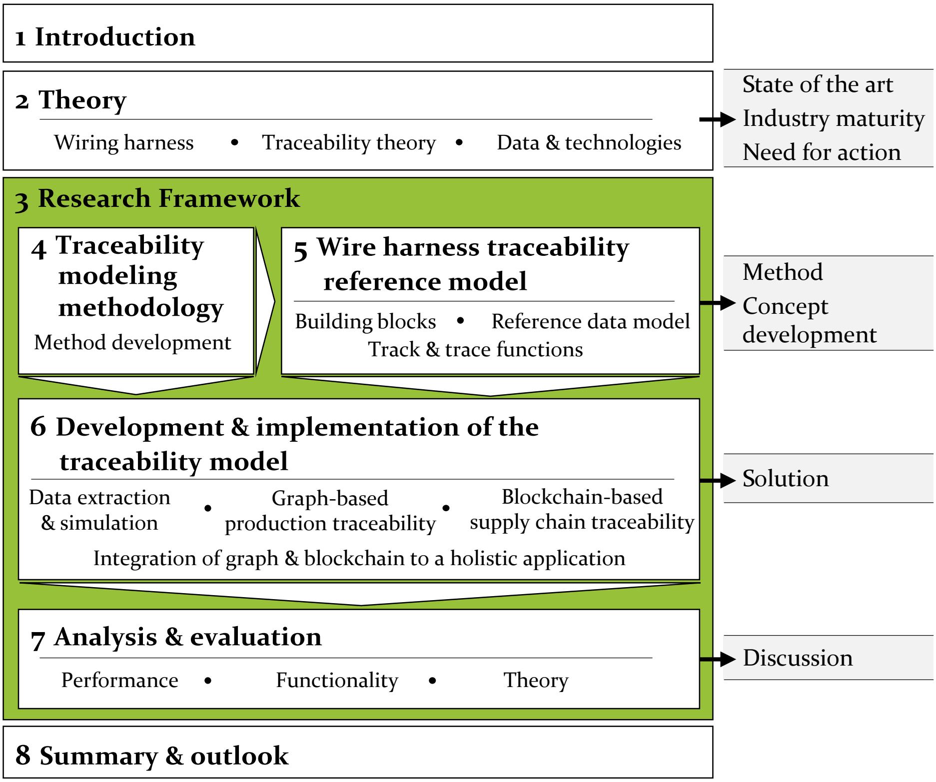


Figure 1: Structure of this thesis

In the following sections, each element of the research framework is developed. Section (4) thus presents the traceability methodology that encompasses the steps to systematically model and develop manufacturing traceability systems. In section (5), the traceability concept for complex manufacturing flows is developed. This chapter is followed by the main development and implementation part of this thesis.

In section (6), the data model for the wire harness manufacturing flow is specified, the tracking and tracing functions are developed, and the system is implemented through a storage structure consisting of a graph database for granular and internal manufacturing data and a blockchain for the macro supply chain traceability data. In the evaluation and assessment section (7), the research is discussed from a performance, functionality, and theory contribution perspective. The thesis is concluded with a summary and outlook (8) into future work.

2 State of the art and related work

This chapter discusses the state of the art for wire harness production and traceability research. Providing traceability for complex manufacturing flows is a great challenge, as they imply unique data histories that require aligned virtual representations and suitable data storage technologies [11]. Complex manufacturing processes can be classified using four complexity dimensions; heterogeneous structures (C1), connectedness and dependencies (C2), dynamism (C3), and uncertainty (C4) [P3, 12, 13]:

- Structural complexity (C1): A large number and variety of objects need to be virtually represented. It is induced through product structural complexity (amount and variety of parts, assemblies, batches) and production process structural complexity (heterogeneity of process logic and technologies).
- Connectedness (C2): Interdependencies between multi-hierarchical product structures and other traceability-relevant data objects need to be maintained and interlinked.
- Dynamism (C3): Flexible manufacturing strategies and customization lead to dynamically arising object relations and frequently changing object states, which need to be efficiently represented.
- Uncertainty (C4): Products do not follow a pre-defined flow but are spontaneously linked to processes and resources, while frequent changes complicate the predictability of the data structure.

The complexity characteristics thus combine product complexity and production complexity features and are not independent of each other but built on another. Exemplarily, the amount and heterogeneity of parts determine the number of production steps and linkages to be maintained, whereby a high degree of interdependency has shown to lead to dynamic and unpredictable behavior from a data perspective [P3]. The complexity characteristics serve as a guideline throughout this thesis by illustrating why the wire harness industry composes a unique traceability challenge and why available solutions have significant shortcomings when applied to complex manufacturing processes. This thesis focuses on these main four characteristics, however, further factors may contribute to the complexity. Examples in the literature include the change management procedures over a product's lifecycle [14] or the supply chain set-up regarding outsourcing, supplier integration, and depth of value-added [15].

2.1 Wire harness production

This chapter gives an overview of the wire harness product structure (2.1.1), its manufacturing processes (2.1.2), as well as supply chain set-up and resulting data management practices (2.1.3). A detailed process and software assessment is given in the last section based on empirically collected data (2.1.4). The description of the state of the art is based on wire harness literature and insights from the in-depth product and process analyses conducted during the empirical section (2.1.5).

2.1.1 Product structure of the wire harness

The wire harness is a mechatronic system responsible for transmitting the energy and signal flow in the vehicle by connecting the system's electric and electronic (E/E) components, electronic control units (ECUs), sensors, and actuators [16, P4]. It consists of wires, contact elements, and connectors as well as mechanic mounting material, bundle, and protection elements such as clips, grommets, tape, tubes, or foam [16, P5], as shown in Figure 2.

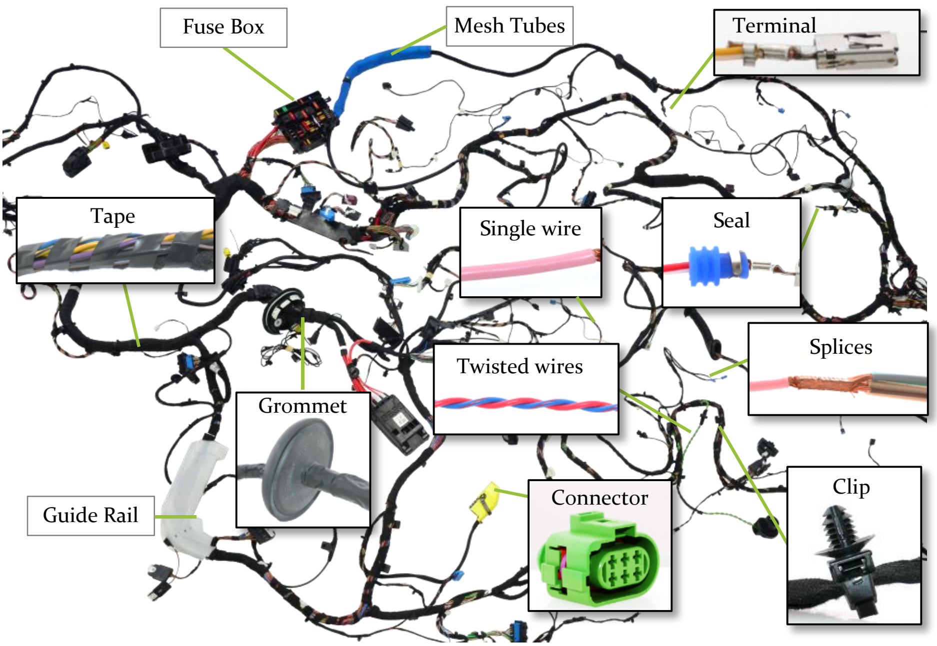


Figure 2: Overview of the typical components of a main car body wire harness

The variety of parts and the size of automotive wire harnesses have continuously grown over the past decades, whereby the shift from mechanical functions to electrified functions further emphasized this development [10]. In modern vehicles, critical and formerly mechanical functions are now enabled by the wire harness and its adjacent electrical and electronic components allowing steer-by-wire or brake-by-wire-systems to perform evasion, braking, or lane switching maneuvers. Although drive-by-wire technologies have not reached serial production, they are expected to be increasingly applied in future vehicle generations [17]. Depending on the vehicle size and configuration, the amount of wires ranges from a couple of hundreds to more than 3000 single wires per vehicle with a cumulated length of up to 4 km and a total weight of up 60 kg, effectively making the wire harness the single heaviest vehicle built-in component after the combustion engine for non-electric vehicles [10, 18]. A set of main product properties characterizes each component of the wire harness. In general, each component is defined by its *type*, *material*, *geometry*, and *assembly position*. To provide an overview of the system's configuration structure and features, these main components and their characteristics are discussed in the following.

The wires form the primary component group of the wire harness. A wire's geometry is defined through its cross-section and length. Typical cross-sections for automotive wires range between 0.135 and 10 mm^2 [19], whereby the diameter determines the maximum permissible current [20]. Single wires consist of copper strands, which are covered by an insulation sheath of different colors. In addition to regular single wires, wire types may be classified as twisted wires, coaxial or specially shielded wires, optical wires, and high-voltage (HV) wires. To generate a detachable connection of wires and electrical components, terminals are attached to the wire's ends and inserted into electrical connectors [21]. The main product properties of terminals are the contact type, geometry, material, and the terminal's pinning position in the wire-connector assembly structure. Each connector consists of a connector housing and several chambers in which the wire terminals are inserted. Thus, a resolvable linkage between the wire harness and the electric component is formed. For electrical connectors, a distinction is made between male and female connections. Housings with sealing requirements have either a single wire seal or a so-called blind plug, protecting the terminals or unoccupied connector chambers from environmental influences [10]. The main features of connectors are thus their type, material, geometry, color, as well as their wire-connector assembly structure (pinning).

A wire harness section is formed by joining parallel lines of wires with bundling material, such as adhesive tape, shrink or corrugated tubes, sleeves, or cable ties [20]. Therefore, the mechanical fixture elements take over a shaping function, which ensures that the flexible and non-rigid wires are transformed to the aspired form of the reserved installation space in the car. In addition to the form-giving function, mechanical mounting elements protect the wires from wear and abrasion, temperature, and liquid. As an alternative to mechanical protection elements, parts of the wire harness can also be foamed, which provides comparable formative and protective functions, especially in water impermeability applications. Water-sealed sections of the wire harness are separated from flooded parts using grommets. Grommets are sealed by injecting a sealing paste and are installed at wet-to-dry passages in the vehicle's installation space. Another class of mechanical mounting elements are fixture components, which position the wire harness in the vehicle body. Furthermore, fixture components reduce noise and clattering by restricting the movement of the wire harness in the installation space. Different categories of clips and fixing elements are most commonly used to securely connect the wire harness with the vehicle body. In this context, the mechanical protection elements, such as tape, foam, sleeves, and tubes also further contribute to the noise-canceling function in interaction with the fixture elements. For tape and tubes, correct assembly is defined by their position, while grommets and clips also need to be correctly oriented. As a final main wire harness element, fuse boxes protect the system from excessive currents. For the fuse box, which is attached to the main 12V-wire harness, the correct assembly position of the individual fuses is one of the main classifying product characteristics. Modern vehicles increasingly apply semiconductor technology to protect the connected E/E components [22], so that the protection function is outsourced from the wire harness to the component.

In the remainder of this thesis, the term *wire harness* refers to the main body wire harness, which connects all main E/E functions in the vehicle. In addition to the main body wire harness, some smaller wire harness modules are also integrated into the vehicle, such as air conditioning, doors, lighting, sunroof, or navigation system harness modules [23]. In terms of installation technology, the wire harness represents a structurally complex assembly of heterogeneous components created and integrated into higher product hierarchies through various joining processes [24]. Exemplarily, splices are applied to connect two separately produced modules.

Table 1 summarizes the main body wire harness's assembly components and their core *product features*. The data are based on analyzing a main body wire harness of a German middle-class vehicle with a basic configuration. The *amount* listed can thus be regarded as a lower limit for structural product complexity, as larger systems, like those with a maximum configuration for luxury vehicle car manufacturers, can double or triple in size. The column *types* displays the number of variants per product group. The analysis shows that all wires in the vehicle are unique, as the amount equals the number of types. For the other components, the same variants are used more than once within one product (e.g. 1200 terminals of 150 types).

Table 1: Product composition and features of an average main body wire harness

Component	**Amount**	**Types**	**Geometry**	**Material**	**Color**	**Assembly structure**
Wires	1500	1500	x	x	x	x
Special wires	100	100	x	x	x	x
Connectors	400	320	x	x	x	x
Terminals	1200	150	x	x		x
Seals	600	50		x		x
Splices	70	3	x	x		x
Grommets	16	11	x	x		x
Tape	no data*	180	x	x		x
Tubes	150	70		x		x
Clips and straps	200	30	x	x		x
Sum	4236	2414				

* The amount of tape measured in meter consumption varies depending on the operator and vehicle project. There is no publicly accessible data available.

Table 1 shows that the product's complexity lies in the system's configuration combinatorics of high amounts and heterogeneous type variants, while the wire harness's product features are comparatively simplistic. The product analysis further indicates that the data management challenge is determined by the correct association and administration of the variant solution space as well as the correct assembly of the individual components.

2.1.2 Wire harness manufacturing flow

The wire harness manufacturing flow encompasses a wide range of heterogeneous production steps. It is often divided into pre-fabrication, which contains tasks of component provision and preparation, and assembly, in which parts are joined into modules and the final system [24]. In this thesis, the process flow is described using four distinctive manufacturing categories, which are (P1) highly-automated cutting and wire processing, (P2) pre-fabrication of modules through assembly and special processes, (P3) wire harness assembly and testing, and (P4) outbound logistic including sequencing, kitting and packing [P5], as schematically shown in Figure 3. The production areas P1 and P2 can be assigned to pre-fabrication, while P3 and P4 belong to the assembly as defined in [24]. The customized assembly of wire harnesses is depicted through the different colors in the P3 area that represent unique variants.

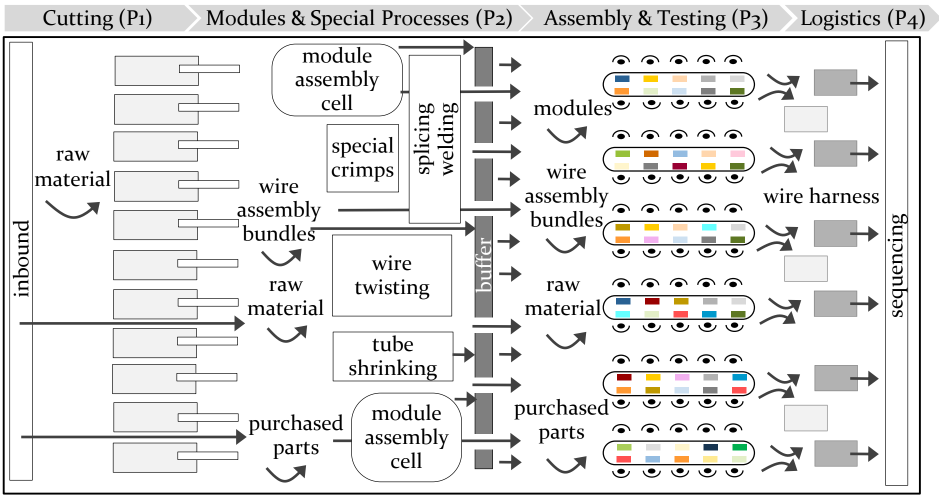


Figure 3: Wire harness manufacturing flow

The manufacturing process starts with the material delivery, which is then registered in the inbound logistics area. In the processing area P1, wires are cut to the defined length, the wires' insulation is stripped from the wires' ends, and terminal contacts are crimped onto the wires [19, P5]. Optionally, seals can be attached between terminal and wire to generate sealed contact points. In general, this process phase is wire-focused, meaning that the wire, which is coiled as a continuous row material in a cask, is transformed using the processes cutting, stripping, seal attachment, and terminal crimping. These process phases can be varied and combined as required so that any combination of sole wire cutting and single or double-sided sealed or non-sealed crimping contacts can be achieved.

Accordingly, in this first phase of manufacturing, the wires are joined with other batch materials resulting in the first discrete product, which can also be classified as an assembly. Assemblies are defined as geometric and topological product liaisons, characterized by their part-to-part links and sequence constraints [25]. Assemblies have hierarchy levels and can consist of sub-assemblies, parts, and sometimes basic or raw materials [26, 27]. Wire assemblies integrate a defined wire type, terminal, and optionally seal type and are produced according to an internal production order. This order is split into producible batches of a defined volume. Each batch is attached with a batch identifier using barcodes or direct wire marking, albeit marking is less common in the automotive industry. Exemplarily, an order of 300 wires of *type A* may be split into six batches of 50 each with consecutive batch unique product identifiers (UIDs), such as *bA1*, *bA2*, etc. The cutting process is characterized by a set of process parameters, whereby the crimping force represents one of the main quality features, which is supervised in-line. Wire lengths, crimp height, and pull-off force are further quality-relevant process parameters that are measured for wire samples. As pull-off measurements are destructive tests, they are conducted for small samples and apply to the entire batch's quality.

In the second manufacturing phase P2, the wire batches are brought to specialized process stations, which perform small assembly or transformation processes in workshop-like production cells [16]. These processes are usually conducted using semi-automated machines and might include splicing and welding, the attachment of shrink tubes, or the assembly or pre-fabrication of wire harness modules. Additionally, wire twisting and special crimping (e.g. for large diameters or unusual terminals such as ring terminals) are conducted in these process cells. Moreover, wire processing similar to process phase P1 is executed in phase P2 if order sizes are very small or the process requirements are too complex for continuous volume production in the cutting area. Quality-relevant process parameters are very heterogeneous as the diversity of processes and production systematics applied in phase P2 are also very variable. Exemplarily, shrinking temperature, tube leaks, welding parameters as well as pitch and final lengths for twists are monitored. As a result of the second phase, small wire harnesses and special wire assemblies are brought into the buffer storage along with the processed wires from the P1 area [16]. A very small minority of wires and modules is directly sent to the assembly line for subsequent processing.

The third process phase P3 constitutes the wire harness assembly, in which each system is put together on a wire harness formboard. Formboards, on which a 150 % drawing of the wire is mounted, sequentially transfer through the working stations on a loop conveyor [P5]. A 150 % drawing is a design representation that includes all possible, even mutually exclusive, connections and variants [28]. The 150 % drawing thus allows all possible variants to be built on each formboard. During assembly, wires from process phase P1 are placed onto the formboard and routed according to the defined assembly sequence and routing paths. The preservation of the geometrical structure of the routed wires is ensured by installation aids such as forks and jigs [21, 29], while the routing procedure might be further supported through worker assistance tools, such as guidance-by-light tools or electrical checks of routing points. The wires are then inserted into connectors, which are put on the formboard in their respective holder [P4]. After inserting the final wire into the connector, connector housings are closed and locked. In addition to wire routing, modules from phase P2 are placed on the formboard and integrated into the overall wire harness structure through joining processes, such as inserting, welding, and splice crimping. In addition to the routing, insertion, and joining processes, mechanical mounting processes are conducted [16]. This includes the placement of tubes, sleeves, and grommets, as well as the attachment of clips and cable ties. Moreover, parts of the wires are taped using different taping techniques such as full tape, spiral tape, or spot tape. Alternatively, parts of the wire harness are placed into a foaming chamber, which presses foam through a valve into the mold cavity. The foam then hardens and encloses the wire harness in a waterproof encapsulation. In phase P3, relevant parameters are the routing geometry, taping shape, and length, as well as the locked insertion of contacts into the connector. Additionally, the correct assembly configuration and the position of clips and grommets are quality-critical. After the assembly process, a quality test is conducted for each wire system in the P4 area. The connector housings of the wire harness are plugged into counter housings which are connected to a testing table. The electrical continuity and the electrical resistance are measured in a software-supported test cycle [20, 21], which holds a unique testing program for each wire harness variant or configuration. In addition to electrical testing, clip tests assure that each clip is mounted in the correct position and orientation.

Following the testing, each wire harness is packed into a radio-frequency identification (RFID)- or label-equipped bag, which is usually part of the closed-loop logistic packaging cycle with the original equipment manufacturer (OEM). The wire harness bags are then kitted into a transportation package allowing subsequently installed wire harnesses to be packaged and supplied together as one unit. Next, the wire harness kits are transported to the sequencing area, where the correct Just in Sequence (JiS) order of wire kits is generated and loaded into a truck. The wires are then transported to the OEM's inbound logistic area (P4).

In the following, quantitative and qualitative characteristics of the manufacturing process are presented and discussed with regard to process complexity characteristics. In alignment with the classification scheme for process and discrete manufacturing systems by Abdulmalek et al. [30], it can be observed that two different process systems are applied in the wire harness manufacturing flow. In the production area P1 and analogous work cells in area P2, highly-automated machines transform yard goods (wires) and bulk goods (terminals and seals) into discrete assemblies (crimped wires), which are managed as batches. It is not uncommon for a process manufacturing system to generate a discrete product result (here crimped wire) and to still classify as process-driven based on its overall characteristics [30]. The production logic is highly linear, variant-driven, and therefore comparable with systems in processing industries, such as textile, bulk, or fluid processing industries. Batches are produced in alignment with forecasts or using a consumption-oriented production strategy based on Kanban buffers. The work in progress (WIP) lies in the range of a couple of days, as demonstrated by [19], who measured 1.5-3.5 days WIP between P1 and P2. The linear and consumption-based production system generates static and predictable process planning and execution cycles, which are typical characteristics for process industries [30]. In the cutting area, processes are primarily optimized concerning machine set-up time, as set-up times constitute a great share of the total operating time [19, 21]. The processes do not operate in cycles or changing phases of production, set-up, maintenance, and down-time. For the execution of wire processing, a variety of mature automation solutions exist, allowing wires to be processed double-sided with a speed below one second per wire. It can be observed that wire processing occupies less than 10 % of the total value-added time while binding 80 % of the overall equipment investment costs.

Contrarily, the assembly process is labor-driven [16, 29]. The main reasons for the low level of automation are considered to be technological difficulties of handling non-rigid products [29], the high number of variants through configuration-driven order management [10], as well as a lacking consideration of manufacturing automation during the early phases of the wire harness product design [P2]. A typical wire harness production contains several assembly loop conveyors of 20-40 workstations, each operated by one to two assembly operators. The vast majority of process steps in phase P3 are entirely manual. This includes the positioning of parts, wire routing, insertion, the majority of the taping, and the mounting of mechanic protection material. Few exceptions are foaming, screwing, and ultrasonic welding, which are supported by semi-automated or hand-held machines. In the process phase P3, the production system can be characterized as a discrete assembly following the classification of Abdulmalek et al. [30]. Discrete bought-in parts, pre-assembled wires, and modules are joined to systems of higher product hierarchies. While the production phase P1 operates with process cycles, in which production phases of less than one second are repeated for hundreds or thousands of wires within one order, the process phases in P3 run with a takt time ranging between 2-5 min. Accordingly, each wire system is assembled by transferring the system through the assembly line and adding a set of defined product contents within each takt cycle. Depending on the size of the vehicle and the ordered system configuration, cumulated assembly times range within 400-1300 min, e.g. the assembly time for a Hyundai wire harness is at approximately 1125 min [19]. The product configuration and resulting processing time are thus rather dynamic from a data perspective. The evolvement from a process-driven flow to a discrete flow can be classified as a hybrid manufacturing system [30]. For the course of this thesis, the wire harness production system will thus be referred to as a *hybrid production system*:

Process-driven steps transform raw and bulk materials based on a production recipe into products with large batch sizes using defined cycle times. The discrete and product-driven system integrates distinctive sub-modules and parts to customized products within a variable sum of takt times.

2.1.3 Wire harness supply chain

When products are produced in cross-country supply chains, the inter-organizational information flow becomes crucial for managing recalls and maintaining liability documentation. According to [31], a *supply chain* encompasses:

> “all activities associated with the flow and transformation of goods from raw materials […] to the end-user, as well as the associated information flows.”

Within the automotive supply chain, harness manufacturers occupy the position of the first-tier supplier. In this work, the research focus lies on supply chains for central European car manufacturers, which apply the German order model of customer-specific car configurations. For European vehicles, the wire harness is the second most expensive cohesive component with a cost impact comparable to the car body [10, 23].

The high costs of modern wire systems are justified in the overall value of the entire wire harness in terms of material costs, production costs, development costs, and know-how [10]. However, the high total cost impact creates high pressure on the network concerning cost-optimal production locations. Contrary to the overall picture in the automotive industry, the wire harness industry is highly labor-intensive. The value network is thus determined by the availability of inexpensive workforces in proximity to OEM production sides. The labor-intensive production system strategy has led to a continuous search for cost-effective production locations [29]. To understand wire harness supply chains, including delivery paths and participating companies, the central European production network is analyzed using publicly available data sources[1] on manufacturing locations and employment numbers. Figure 4 shows the result of the analysis for major central European vehicle manufacturers. For the analysis, large OEM producing in central Europe are considered, such as Audi, BMW, Daimler, FCA, Ford, Opel, Porsche, PSA, Renault, and VW. As first tiers, production plants with more than 400 employees are integrated. The resulting factories belong to Aptiv, Dräxlmaier, Kromberg & Schubert, Lear, Leoni, and Yazaki.

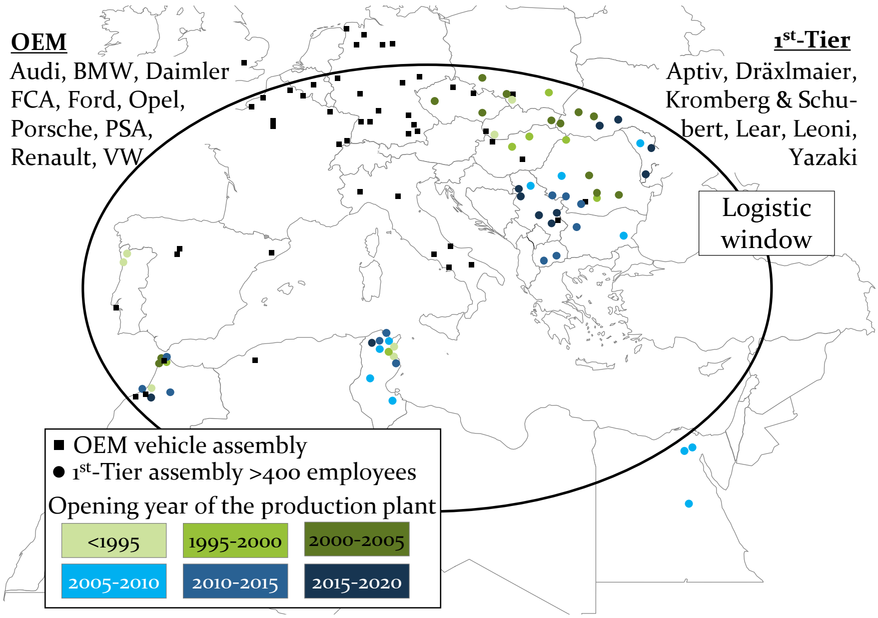


Figure 4: Wire harness supply chain network and locations at wage-distance trade-off

[1] annual reports of the listed companies, homepage information, and public trade databases

The analysis shows that wire harnesses are predominately produced in East Europe and North Africa The choice of production location is dominated by a two-factor trade-off between production costs and logistic costs. Simplified, the trade-off's equilibrium leads to an approximated logistic window, in which the costs for covering the distance to the OEM's production sites including transport costs, time costs, and risk costs balances the labor and manufacturing costs at the wire harness production site. In addition to other site-determining factors, like infrastructure, know-how, or taxes, the supply chain is predominately configured using those two variables. Additionally, the logistic window sets the maximum planning horizon for the OEM, which has lead to more production sites being built at the maximum distance for a profitable JiS and customized wire harness business model with an ordering horizon of a couple of days. In addition to the general overview of supply chain structure and the dominant site selection factors, the analysis suggests relocation movements and shifting of manufacturing volume from Eastern Europe to South-Eastern Europe and North Africa, as shown by the opening years of production sites. The analysis shows that wire harnesses production supply chains are decentralized and span across several factories and countries. The distributed structure with long distances leads to a network that is highly intransparent and difficult to monitor [P1].

As mentioned in the previous chapter, product variability is comparatively high in wire harness manufacturing. For the central European market, wire harnesses are configured based on an end-customer order and produced in lot-size one [P5]. Customers can configure their vehicles according to their wishes by choosing their preferences from a set of options. The majority of those choices directly affect the wire harness configuration. Even options concerning mechanical features often influence wire routing and thus wire length, routing geometry, and connection points. Therefore, the high product variability originates in the central European OEM's order strategy and the increasing importance of the vehicle's electric system, which is influenced by the majority of vehicle configuration choices.

Each orderable configuration module is represented by a so-called functional module (FM) in the wire harness system design. Thus, each FM directly translates to an order decision, such as seat-heating, a navigation system, or a hill hold control [10]. As every vehicle has some basic non-configurable parts, each main body wire system has a basic module, which contains the minimum required wire content, such as general power and data distribution, braking, steering, and battery systems. In addition to optional modules, FM are defined for left-hand or right-hand drive vehicles

(LHD and RHD). Accordingly, each wire system can be described as a set of mandatory and optional functional modules, which represent the configuration logic of the respective OEM as shown in Figure 5.

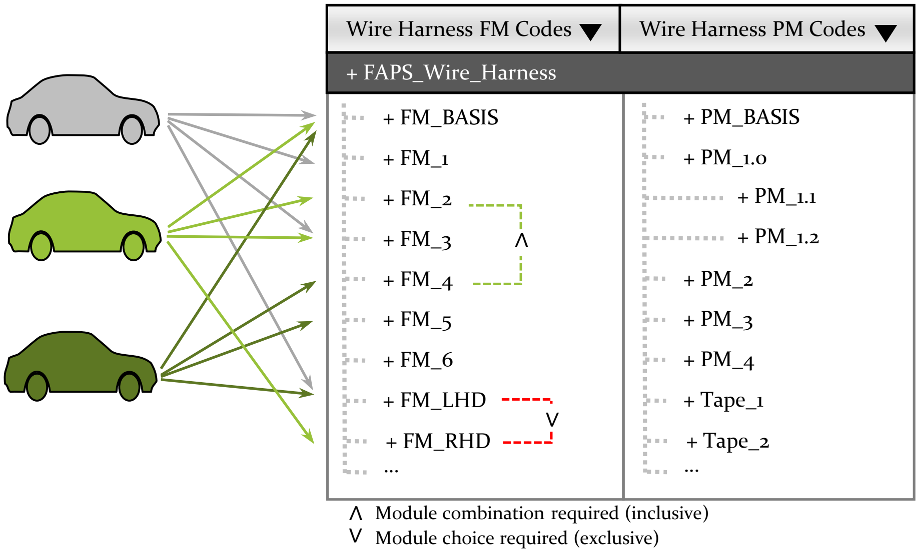


Figure 5: Wire harness modularization management induced by OEM's order model

This also means that the majority of FM come with certain restrictions and associated conditions, such that some modules must be ordered together (e.g. *FM_2* and *FM_4*) or are mutually exclusive. For *FM_LHD* and *FM_RHD*, one of the modules must be selected, however, selecting both is impossible. As FM represent the vehicle configuration logic, they are not very effective in managing the wire harness content from a production perspective. Accordingly, each FM is translated into a set of production modules (PM), which modularize the FM's content into producible units. Those units are often further divided into sub-PM or variants (e.g. *PM_1.1*) to mirror certain variant configuration restrictions of the order model or specialized modules (e.g. *Tape_1*) to group manufacturing steps across modules.

The rationale is to consolidate the aspired product content by ordering catalog features and then translating those features into functional wire harness modules. The configuration restrictions are stored in a file called complexity list. The functional features are then further modularized into producible units. Since the number of options is so high, each wire harness effectively composes a unique product configuration, which is only compatible with one specific vehicle. This model of producing unique wire harness systems in lot-size one based on a customer order is called KSK-model (German: Kundenspezifischer Kabelbaum or English: customer-specific wire harness). As described in the previous section, the wire system consists of several thousand components and pre-assemblies, which need to be ordered from second-tier suppliers and pre-manufactured before the final system is assembled. Since the OEM allows the customer to make short-term changes to the configuration, the finalized configuration is usually ordered by the OEM a few days (24-120 hours) before it needs to be available in the OEM's inbound logistics. Accordingly, order cycles are comparatively short-termed and their content is difficult to predict. The OEM pass their function prediction to the harness manufacturer as the function modules' take rate (e.g. *take rate (FM_BASIS) = 100 %).* However, volatile markets lead to volume and content fluctuations, directly impacting the manufacturing time. Figure 6 shows an evaluation in which the assembly time per wire harness order is plotted for an average two-shift production.

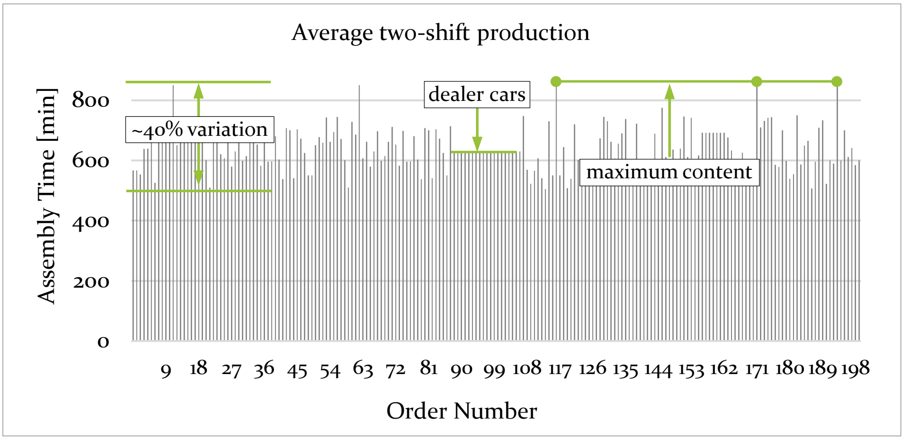


Figure 6: Assembly time fluctuation based on content variation for a sample production flow

For the analyzed wire system, an approximate 40 % content and thus assembly time-variation could be observed, whereby the depicted orders are presented in their ordered and not in their manufactured sequence. The analysis illustrates that both the production and the supply chain require a high degree of flexibility in their underlying data system to represent KSK logic efficiently. Together with the frequent and often short-term changes in product design as well as the general volume fluctuations, the KSK model leads to a highly dynamic manufacturing process with unique data histories. The needed flexibility and fast reaction times are further emphasized when considering the difficult and long delivery routes from the North African continent to central Europe. This analysis has shown that the wire harness final assembly can not only be classified as *discrete* but also as *order-based*. Order-based manufacturing systems need to respond quickly to dynamic changes [32], while the non-stability results in constant planning and manufacturing monitoring issues [33]. Success factors in dealing with those types of manufacturing systems have been identified to be appropriate modeling of manufacturing data and the usage of capable software infrastructure [32].

2.1.4 Wire harness software infrastructure

In the following, a brief overview of the wire harness manufacturing software infrastructure is given to introduce the main data flow as well as the involved administrative units, as shown in Figure 7. The engineering aspects of the data flow are based on [10] and own process analyses conducted for this thesis. The manufacturing aspects of the data flow are entirely based on case study results (2.1.5).

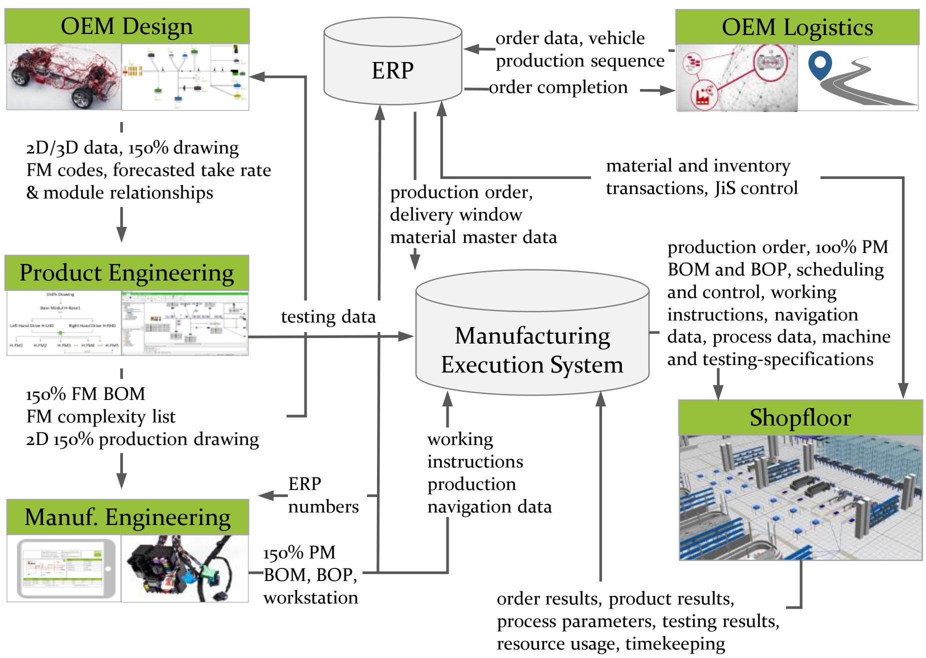


Figure 7: Data flow for wire harness engineering and manufacturing

The OEM designs the automotive E/E system from a functional perspective including the connection of all E/E components and the reservation of the installation space that limits the geometry and routing of the physical wire. Based on this, the OEM derives a 150 % 2D-drawing containing all configuration options and their geometry, such as length, diameter, or radius, from the perspective of vehicle installation. Additionally, all FM codes, their complexity list, and forecasted take rates are passed to the wire harness engineering unit. Product engineers then develop the physical wire harness by defining the material types, geometry, and physical properties of the product. As a result, the functional bill of material (FM BOM) with the modules' dependencies and exclusion criteria, as well as an enriched 150 % 2D-production-drawing with production-revised lengths, bending radius, and positional value calculations, are generated. Based on the FM BOM, a 150 % testing program for electrical pin-to-pin testing is created and sent to the manufacturing software system. In manufacturing engineering, FMs are modularized into PM and a 150 % PM BOM and its corresponding bill of process (BOP) are developed. Based on the PM BOM, the Enterprise Resource and Planning System (ERP) allocates an ERP number to each part. Additionally, each PM is assigned to a workstation and the line is balanced with regard to the expected throughput and defined takt time. The working instructions and production navigation data are created and sent to the MES. All data exchanges at unit interfaces (visualized by grey arrows) occur document-based.

Until this step, all data objects are planning and engineering objects referring to a 150 % virtual wire harness project. To initiate the physical flow and therefore, instantiated data objects, the OEM generates a wire harness order list according to its production schedule and sequence of vehicle assembly. Each order data object contains the configured FM codes and the sequence of JiS-calls. The ERP then issues the production order including a list of PM BOMs generated from the ordered FM as well as the aspired delivery window. Moreover, the ERP manages the inventory and material transactions of physical logistic flows at the shop floor. The MES then analyzes the ordered PM for the day and generates a 100 % PM BOM for each ordered system. It further controls the manufacturing flow by requesting certain process and product parameters as well as quality tests to be passed to the system. Moreover, it creates quality records for liability cases. However, the actual data generated in the shop floor is very limited, which will be analyzed in more detail in the empirical section (2.1.5).

2.1.5 Empirical assessment with the case study

To evaluate the maturity state of the current wire harness manufacturing process and value network and to derive future requirements, a case study was conducted. The systematic research methodology explores phenomena and variables within industry-driven topics in early stages of research and development [34]. It was chosen to overcome literature gaps in this research field and to provide the empirical data needed to develop future traceability solutions. This thesis follows the case study methodology as proposed by Yin [34] and Eisenhardt [35] for expert selection, data acquisition, analysis, and assessment. The case study was designed as a multiple (several cases), embedded (with several analysis units) study. In total, 54 experts from 26 companies and seven countries participated. The companies ranged from German OEM to first- and second-tier suppliers, as well as university experts and software companies. The interviews were transcribed, anonymized, and coded using coded clusters. Whereby this thesis focuses on the results concerning traceability, more detailed results were analyzed in [S1–S3] and published in [P1, P2]. In addition to the interviews, two cases were investigated in more depth for three years. The two cases included focused on *wire harness manufacturing* and *engineering software*, as shown in Figure 8.

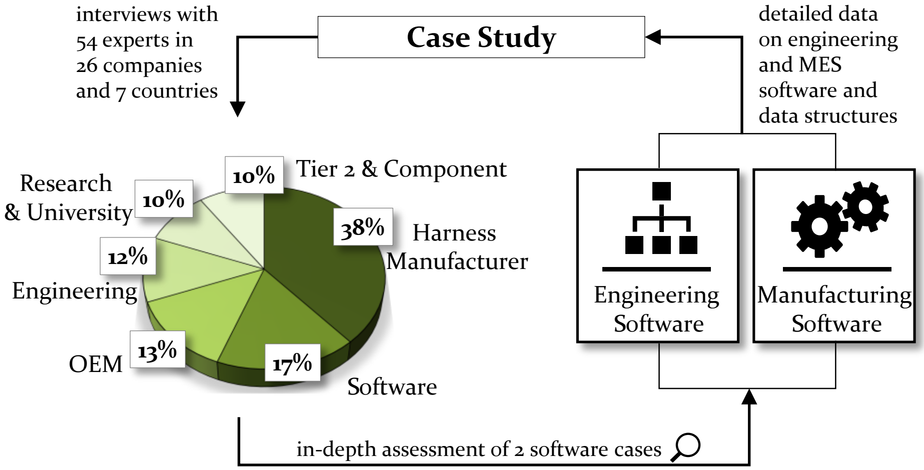


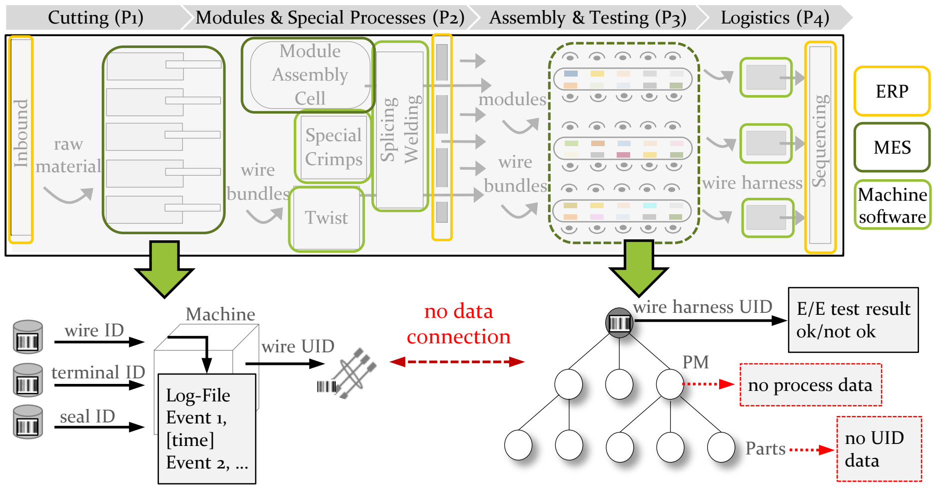
Figure 8: Case study design for the wire harness industry adapted from [P2]

Part of the previously outlined production and information system analysis (sections 2.1.1-2.1.4) also built on the analyses conducted for those two cases, however, they were described in the theory section as they represent common knowledge of industry-familiar experts, whereby this section focuses on novel insights regarding traceability maturity and requirements. The case study yielded that the overall main weakness in the wire harness industry is the discontinuity in processes and data. Specifically, lacking traceability and data continuity in operations were seen as even more severe obstacles than for example lacking automation or a high product complexity, which are the main challenges according to the literature [16, 29]. Data at interfaces are exchanged over documents or files instead of relying on a shared data model, leading to a low overall digitalization and low data modeling maturity. [P2] The in-depth analysis of the two software cases further supports the general interview findings that data management is one of the greatest weaknesses in the wire harness industry. The assessment of the software cases has shown that in a typical production line a wide range of monolithic and silo-structured software solutions and data storage systems are applied. One of the main challenges is that the systems in wire processing and assembly operate with very different and non-compatible logic, which is one of the biggest barriers to data consistency.

In wire processing, the manufacturing software focuses on scheduling orders, optimizing machine efficiency, and displaying work instructions, however, the software is not configured to provide traceability. As shown in Figure 9, production data are logged from an event-driven machine perspective to a TEXT-file. This document is stored within the machine log and is not directly readable by other systems due to its non-machine-readable format and semantics. For wires, the MES creates product UIDs which provides the baseline to create a traceability record. However, the case study showed that as data are only logged from a machine perspective, there is no structured data stored with a reference to that UID. Another weakness is that the software focuses on verifying serial numbers or IDs of input materials to ensure that the correct type of product is produced without enforcing batch management (logging which exact batch or item is produced). The software solutions are thus applied to monitor process stability, however, none of the collected data are transferred to an overall database. The data are thus only available during occurrence or within internal logs.

In the assembly area, only a few companies use software support. If available, a specialized assembly-MES provides the product structure per order *as planned*, showing which parts need to be integrated to align with a KSK order. However, the as-planned data are generated by the engineering software in a document-based and non-semantic format. The assembly-MES can thus provide dynamic product structures using those documents and

ERP orders, however, it does not interpret what data objects are displayed and is thus unable to link actual data to those objects.



Partial traceability	
Available data	**Data problems**
Serial ID are scanned and checked for correctness.	UIDs for incoming parts are not stored → no batch management.
Data streams are created per machine and stored in a log-file.	Data only available from an event-driven machine perspective
Wire UIDs are assigned to produced batches.	No data are stored with the wire harness UID.
Data relations can partly be associated via timestamp.	No database with linkable product, resource or process data is created.

No traceability	
Available data	**Data problems**
Serial ID are partly scanned and checked for correctness.	UID for used parts are not stored and associated with the final wire harness UID.
Process planning sequence is sent to each work station (PM BOM).	Process execution data and quality feedback are usually not stored.
Wire harness UID are assigned to the final product.	No manufacturing data are stored for the wire harness UID.
E/E test results can be linked to wire harness UID.	No database with actual product, resource or process data is created.

Figure 9: Data maturity in wire harness manufacturing based on the case study

The software thus ensures that the correct harness is built by displaying appropriate instructions, however, minimal data feedback is generated and built-in parts and sub-modules from previous steps are not scanned with their batch UID. Accordingly, all parts are assembled anonymously. While some parts are checked for correctness using their serial ID, the batch UIDs or part UIDs are not stored so that the batch UIDs created in wire processing are effectively lost from a data perspective. As UIDs from wire processing are not stored in the assembly-MES, there is no connection between the data within the wire processing area and data of the assembly line.

Moreover, bought-in parts such as connectors are only verified for conformance by checking serial identifiers (ID), but batch data is not logged. IDs can thus only verify which type of part was used but do not refer to an exact part or batch, which can only be accomplished through UIDs. All software products thus work completely independently with ERP functioning as a common inventory understanding. Whereby in the processing area at least a partial traceability is created through machine logs, the assembly area can be seen as a data black box with its only output being the final wire harness UID and the result of the electrical test. Software in assembly thus only operates with planning data and creates no actual traceability records. Accordingly, no overall traceability database is maintained, so that even basic traceability questions like *what batches were used for this product* or *on which resource was this product built* can not be derived. The case study has shown that the dyadic manufacturing software structure paired with the document-based data exchange between units and systems are major obstacles for ensuring data continuity and traceability in wire harness manufacturing.

In addition to the software structure, the most common failures in wire harness manufacturing were analyzed through the case study and literature to give a guideline for traceability system development. Due to the high amount of different process steps and technologies, as well as the handling difficulties of the wire system, the quality management of the wire harness production is comparatively challenging. In automotive wire harness manufacturing, the failure rate lies in the single-digit range, whereby the most common failures are caused by manual assembly [20], which alone induces failure rates in between 200 and 1500 parts per million [36]. Contacts are usually of good quality from a manufacturing standpoint, however, frequent failures occur by mounting contacts into the wrong pole chamber [21]. The most common routing errors occur because wires are not laid along with the specified routing geometry, which may lead to significant problems in subsequent process steps, such as the installment of the wire harness in the vehicle. In addition to assembly errors, failures in earlier process phases increase the failure risk in subsequent phases [21], as problems with the insulation or crimping contact might lead to quality issues during insertion into the connector housing. Technical errors in process phase P1, such as splicing effects on stripped wires, missing insulation, or faulty crimp termination, are currently effectively detected through in-line sensors of cut-strip-crimp (CSC) machines or testing procedures on product samples. However, the anonymous assembly and the usage of as-planned data through all process steps hampers the verification of a correct product configuration and structure. In this context, one case study participant stated that he or she "would expect 10-20 % of all vehicles have a faulty assembly configuration. This will become very critical for autonomous vehicles when it cannot be ensured whether the product was built as defined".

In summary, while some data are collected in wire harness manufacturing, the overall maturity is comparatively low. Some major issues are the inaccessibility of data logs, the unsecured linkage of data with super-ordinate objects, as well as an overemphasis on a document-based exchange, and a complete absence of semantic data. The difficulty in data management is partly inherent to the complexity characteristics, which aggravate the implementation of traceability solutions. The large number of variants paired with the many levels of assembly hierarchies result in a generally high structural complexity (C1) and interdependency complexity (C2). Moreover, the KSK logic leads to unique configurations and non-deterministic data records per product (C2, C3, and C4). The production volume and content fluctuations emphasize a high data dynamism and uncertainty in the production flow (C3 and C4). From a holistic data perspective, an additional challenge seems to be rooted in the *hybrid production system*, which presumably causes a set of continuity issues by exacerbating holistic process control and traceability across all process steps (C1 and C2). This hybrid logic with uniquely configured software solutions is atypical, as most manufacturing systems in literature display a more homogenous structure. This thesis will further show that the aspect of holistically connecting the data of two opposing manufacturing systems has also not been taken into account in the traceability literature yet (chapters 2.2 and 2.3).

In addition to the maturity assessment, the case study interviews focused on future requirements. The results, which have partially been published in [P1, P2] can be summarized as follows:

- The production strategy shifts from a purely cost-driven strategy to a quality- and safety-driven strategy, since larger parts of the product, are becoming safety-critical.
- Traceability was seen as the major lever to prepare the industry for the future of autonomous vehicles. It was mentioned by more than 90 % of experts, while automation, for example, had a consent of a bit above 50 % (open questions were asked)[2].

[2] Open question: Will autonomous driving impact the harness development and production process? If yes, how? What changes are necessary?

- Failures need to be efficiently tracked to the root. Detailed manufacturing documentation will become mandatory and legally binding by OEM with a focus that the product was built correctly.
- Wire harness manufacturers will face a shared responsibility and liability in the case of a failure or recall. A common traceability database should be consistent, trusted, and binding for all parties contributing to the final product.

2.2 Traceability fundamentals

In this section, the main principles of traceability are presented and results from adjacent research fields are discussed. Interestingly, traceability is a research field that different and independently-working research areas have investigated. For this reason, there is no coherent definition of traceability but different conceptions and interpretations across its fields of study. In production-driven research fields, traceability refers to “the ability to retrace process steps and verify that certain events have taken place” [37]. Cheng and Simmons introduced one of the first manufacturing traceability concepts, in which traceability is defined as the ability to monitor the status, assess the performance, and evaluate goal achievement across strategic plans, design data, as well as manufacturing planning and execution data [37]. In general, production authors define traceability as a data collection and reporting process for parts characterized by physical lot integrity and marked with identification technologies [37–39].

One of the dominant traceability research fields is the supply chain management (SCM) of food networks, as legal and normative requirements aiming at risk mitigation caused early advancement in this field [40]. There, traceability is seen as the ability to identify current and previously recorded states of a product or object, whereby the state primarily includes location data, association with other objects, and object properties [41]. Food and SCM research highlight the importance of following products and recording data over the entire value chain through every step of production [42]. In this context, (forward) tracking refers to the ability to find the current localization of a product in the supply chain or to follow a part downstream, while (backward) tracing refers to retrospective identification of characteristics and origin of a product starting from a set of given criteria [43–45].

The largest traceability research field is software engineering with a strong focus on software system modeling and requirement engineering. The work in this field developed almost completely independent from value-chain-focused research fields. Nevertheless, tracing virtual items follows analogous logic and principles as tracing physical products. Based on the IEEE Standard Glossary of Software Engineering, in which traceability is defined as “the degree to which a relationship can be established between two or more products, [...] for example, the degree to which the requirement and design of a given software component match” [46], traceability conceptions with a focus on objects and relationships became more dominant. In this context, authors define traceability as any relationship that exists between artifacts and the resulting ability to associate and link dependent items in the software development life cycle, in both forward and backward direction [47–50]. In software engineering, traceability is further split into horizontal traceability, which refers to the ability to trace correspondent items across models, and vertical traceability, which describes the ability to trace dependent items within a model [47]. Although the focus of this thesis is on manufacturing traceability, a definition following the work in the field of software engineering [46–50] and previous works [P6] is proposed:

Traceability is the ability to bidirectionally associate and trace dependent objects within a model (vertical traceability) as well as correspondent objects to connected models (horizontal traceability) for the benefits of enhanced system understanding, monitoring, and analysis.

This general traceability conception will facilitate the development of a comprehensive traceability understanding, which does not exclusively focus on physical objects but allows linkages of physical and virtual objects within and across traceability models. Combining the definitions of software engineering and SCM, tracking and tracing can be defined:

Tracking refers to the process of following and monitoring an object or status while tracing refers to the process of retrospectively aggregating object and relation data from a given set of criteria.

The given set of criteria function as a reference point for querying and aggregation, limiting the tracing function to a certain point of time, object, or location. A suitable traceability system thus enables tracking and tracing:

A traceability system provides and maintains the data records, data associations, and functions to enable tracking and tracing throughout all defined operations.

2.2.1 Traceability principles and frameworks

The benefits of implementing traceability in manufacturing are manifold. One of the most common objectives is the reduction of recall risk and costs.

As shown by the work of Murphy et al. [7], product complexity, outsourcing strategies as well as increasing external requirements have intensified the recall risk in the automotive sector. As a recall induces high costs, such as direct recall and repair costs, legal costs, and equity costs (loss of sale, reputation, etc.), enhanced risk mitigation and management strategies are needed [7]. In addition to preventing and managing recalls, a traceability system provides a clear documentation and facilitates holistic system understanding. It builds the base for analyzing the impact of changes and modeling the system's consistency [47, 51, 52]. From an internal perspective, a traceability system allows fast detection of anomalies. By statistically analyzing traceability records, trends can be observed, weak elements of the production system can be identified, and continuous improvement measures can be initiated [53]. Externally, traceability permits companies to align to rising normative and legal requirements, like ISO 26262, IATF 16949, and liability shifts, as well as rising customer expectations concerning failure monitoring and reimbursement. Furthermore, traceability is seen as an effective measure to manage the rising complexity in operations through systematic information system modeling [54, 55]. In summary, a traceability system improves the product and process quality while simultaneously reducing direct costs as well as risk-induced costs [37, 39, 53]. Many authors stressed the importance of building traceability models and frameworks to create a common understanding and methodology in developing traceability solutions, shown by the works of [37, 43, 49, 54–58]. Accordingly, traceability frameworks and models can be defined as follows:

The framework sets the conceptual context of what data are to be traced and which granularity scope is to be implemented. The traceability model formally defines and classifies traceability-relevant objects, their links and associations, as well as functions to be achieved.

While there exists no common theoretical framework or model [43], traceability solutions in their respective field are often based on similar characteristics and assumptions. In SCM, traceability models focus on part identification (what), origin and location data (where), as well as controller and organizational information (who) [41]. In manufacturing research, traceability frameworks encompass planning and operational production information and focus on product and process data. Exemplarily, the aforementioned framework by Cheng and Simmons includes product development and production data, such as process planning, scheduling, material management, testing, and control data [37]. Analogously, Wank developed a product-based framework in which lot traceability is divided into testing, material, logistic, and process information [39]. While some manufacturing traceability frameworks contain both, product and process data, most focus on one of them. In process-driven frameworks, the products are seen as fixed units (lots or batches), which undergo recorded transformation processes of which all related data are collected [38, 59]. Component-based frameworks put their emphasis on material structures, which are modeled as a connected product tree by registering all relations between sub-ordinate and super-ordinate material lots [38, 44]. Regardless of focus, all state-of-the-art frameworks require the unique identification of products or traceable units [43]. To set the scope for this thesis, the current literature is aggregated into a holistic traceability framework depicted in Figure 10.

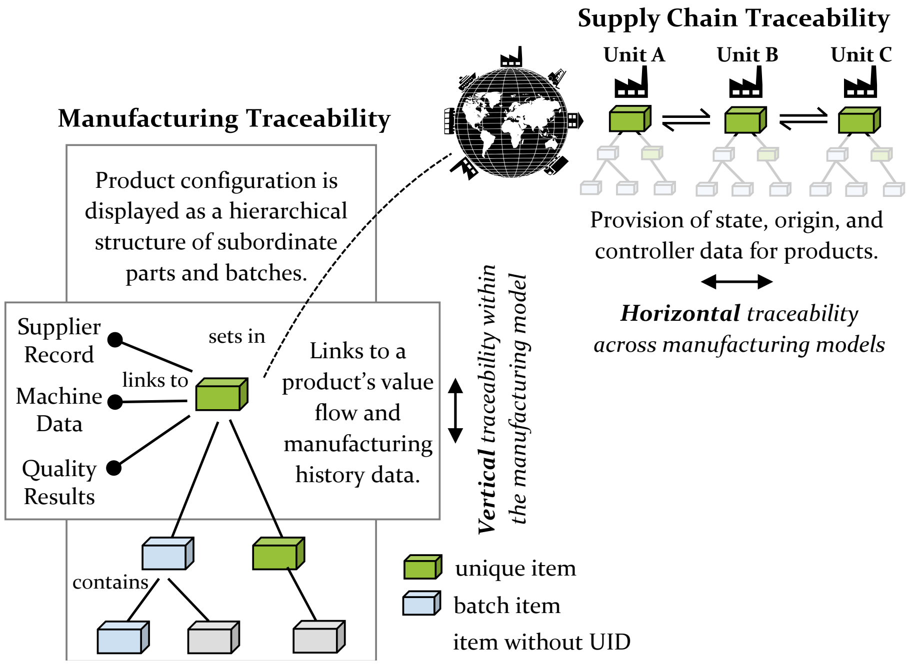


Figure 10: Traceability framework based on the preliminary works of [37, 38, 41, 43, 44, 59]

The traceability framework encompasses the product configuration, which contains the hierarchical structure of subordinate parts and batches. In alignment with the wire harness use case, the framework contains unique items, batch items, and items without identifiers. For each item, process information is stored throughout the manufacturing flow. As this thesis focuses on products that are produced in multi-factory manufacturing networks, different companies contribute the final product and thus traceability record. Accordingly, two main traceability models need to be developed and implemented; the manufacturing traceability model and the supply chain traceability model. While the manufacturing traceability model provides detailed configuration and process data, the supply chain traceability model enables a macro-view by monitoring the final product composition and status changes across organizational units. Data within the manufacturing model is classified as vertical traceability, while the association across organizational units in the supply chain provides horizontal traceability.

2.2.2 Traceability applications in different industries

As discussed in the previous sections, the main related research fields for this thesis are supply chain traceability, manufacturing traceability, and software traceability. To address the challenge of hybrid production systems, the traceability approaches of process and discrete manufacturing are analyzed separately as illustrated in Table 2, considering industries ranging from food or pharmaceuticals to electronics, avionic, and automotive.

Table 2: Evaluation of related traceability research fields

	Supply Chain Management	**Process-driven Manufacturing**	**Discrete Manufacturing**	**Software Engineering**
Research focus	proof of origin & location safety, quality & recall measures counterfeit/ contamination management	manufacturing monitoring & control architecture of information technology batch integrity	engineering & planning continuity manufacturing monitoring data modeling & continuity	traceability methodology, models, schemas & frameworks stakeholder perception
Technical solutions	RFID, EPC, EPCIS, XML, GPS, WSN standardization of identification, exchange formats & packaging	batch-driven data models/ software RFID, auto-ID WSN, IIoT & industry 4.0 frameworks	manufacturing ontologies PLM models (PPR, BOM-driven) CAD, ERP, MES	traceability meta-models, link schemas & model-building software traceability algorithms & functions

Application	Critical supply chains for any type of industry	batch-driven transformation: food, pharmaceuticals, consumer goods, etc.	complex products & assemblies: electronics, avionic, automotive, etc.	requirement engineering, model-driven software development
Mat.	high maturity	medium-high maturity	low-medium maturity	very high maturity
Drivers	safety laws & regulations consumer pressure	shopfloor monitoring & control digitization	complexity management failure/recall cost decrease	complexity/ consistency management quality & time
Authors	[42, 43, 58, 60–65]	[1, 37–39, 66–69]	[7, 26, 32, 59, 70–73]	[47, 49, 54–56, 74, 75]

Surprisingly, the fields have little overlap concerning their focus of study, methodologies applied, perceived drivers as well as solutions presented to address traceability challenges. In the field of SCM, traceability maturity (Mat.) is high and numerous publications exist, especially for food and pharmaceutical supply chains, in which regulative requirements are strict and counterfeits as well as contamination threat product safety. The research focus lies on identification and standardization technology frameworks like RFID and their relation to universal data standards such as electronic product codes (EPC), electronic product code information services (EPCIS), and location and transportation standards. Additionally, wireless sensor networks (WSN) and Extensible Markup Language (XML) data exchange procedures are discussed to provide interoperable data exchange across global value chains and organizations.

In process-driven manufacturing, traceability is established to achieve manufacturing monitoring and control through appropriate information system architectures. High emphasis is put on RFID solutions paired with WSN, auto-ID, and IIoT frameworks to ensure batch integrity and data continuity in batch-driven transformation processes. Solution maturity is comparatively high, as process traceability is obtained by continuously matching the main product's UID with data throughout the flow.

In discrete manufacturing, the majority of traceability solution addresses product modeling through dynamic and structured BOMs (SBOMs) or digital twins across digital toolchains. The solutions emphasize variant and complexity management and rarely integrate actual and instantiated production data, such as real product identification, process data, or resource parameters. They focus on design traceability using computer-aided design (CAD) or on product lifecycle management (PLM) tools and emphasize the modeling of manufacturing systems using ontologies and the often applied product-process-resource (PPR) schema. Interestingly, there is barely any research, which integrates instantiated data, similar to the shortcomings found in sections 2.1.4 and 2.1.5. Data are thus maintained as planned without storing actual results.

The most mature and oldest traceability research field is software engineering with traceability applications in requirement engineering and model-driven software development. In contrast to the other traceability research fields, the presented work does not focus on presenting tailored solutions for specific traceability problems but develops commonly accepted traceability methodologies, model schemas, and languages. As a result, software engineering is the only traceability research field with well-established methods, modeling frameworks, and traceability tools, which help practitioners to independently develop application-tailored traceability solutions. In software, traceability is used to increase product quality, decrease development speed, and propagate changes.

2.2.3 Related work in the automotive sector and discrete manufacturing industries

In the automotive industry, traceability has been a topic of increasing relevance, especially within the last ten years, in which constantly rising recall rates have brought the industry more and more into disrepute. In 2016, a record amount of 50 million vehicles were recalled in 339 unique recall campaigns in the United States (U.S.) with a great share caused by the Takata airbag recall, which has been blamed for twelve fatalities and more than 180 injuries in the U.S. alone [8]. In Germany, the number of recalls has been growing since 2010 from approximately 180 recall campaigns to over 600 campaigns in 2019 [6]. One of the main reasons behind the recalls is the rise in technologically advanced vehicles, which lead to more complex system configurations with unknown failure types in the field [8]. With increasing electrification and the growing product complexity of the E/E system, the likelihood of recalls has increased analogously for E/E components, so that the E/E system made up 12-15 % of all German automotive recalls in the last years [8]. Exemplarily, 1.1 million vehicles were recalled within one single campaign due to cables that caused vehicle doors to open while driving [8], while Daimler recalled more than one million cars due to defective steering column wires [76]. In addition to the wire, connectors, ECUs, and mounting parts of the harness were analogously affected, as exemplarily shown by the BMW transmission control connector recall in 2017 [77]. With the introduction of AD functions and components, the recall risk rises even further. Murphy et al. found that products with AD and driving assistance technology already make up a large share of recalls and display significantly higher recall growth rates compared to non-AD parts [7].

While previous research often assumed that failures leading to recalls have their origin in design issues, current analysis shows that only one-third of recalls can be traced back to the design, while the other two-thirds were manufacturing- or assembly-related [8]. Accordingly, in the last ten years, vehicle manufacturers increasingly focused on manufacturing monitoring and made public claims on the introduction of a traceability system. Exemplarily, BMW and Aston Martin introduced an RFID-based manufacturing traceability system to ensure the correct assembly of parts, while VW and Jaguar implemented a supply chain traceability system for their logistics flow using RFID [78, 79]. Analogously, Toyota and Chinese OEM implemented MES, which rely on RFID-based data continuity to track parts production [1]. In the context of this development, Michalos et al. found that RFID has been the dominant solution and one of the main enablers in the automotive industry to address product traceability issues [80]. Even in very recent publications, RFID and other identification technologies such as barcode, data matrix code, QR-code or near field communication (NFC), are used as means to provide data continuity across process domains aiming at improving process visibility, real-time traceability, and assembly control [1, 81, 82].

The analysis in this thesis clearly shows that the vast majority of automotive and discrete manufacturing traceability research focuses on information systems and software that relies on unique product identification to bridge data. Interestingly, there seems to be the underlying bias that non-tagged parts can not be traced, reinforced by the fact that the identification-dependent MES design requires product UID scans to associate further data with it. Effectively, data for non-tagged products can not be recorded by those systems. In the following course of this thesis, this philosophy or assumption will be referred to as the *tagging bias*. Based on the analysis of current traceability research and proposed identification-based traceability systems, the following shortcomings can be stated:

- The assumption applies that unmarked products cannot be traced. Products without identification tags have the reputation of causing insurmountable data gaps in traceability systems. Accordingly, complex product structures with non-marked intermediate assemblies can not be fully traced with available traceability solutions.
- In many works, data connectedness is thus not an inherent characteristic of the information system's data model but is generated through physical data decoupling and consolidation. Data gaps are bridged through UID scanning. However, in smart and data-driven manufacturing, multiple entities of the factory can be used to maintain and link data as demonstrated in the reference architecture model industry 4.0 (RAMI 4.0) [83].
- Manufacturing traceability either focuses on process-driven models, in which detailed manufacturing data are collected for a defined item or batch UID, or product-driven models, which focus on digitally mapping product structures, but lack means of integrating actual data feedback from manufacturing operations and resources. For the use case of wire harness manufacturing, both models need to be integrated due to the hybrid production systematic of process-driven wire processing and discrete final assembly.
- Systematic modeling approaches and languages from the field of software engineering have not been adapted for manufacturing purposes and are thus not applied. Manufacturing traceability systems are developed tailored to the use case.

2.3 Traceability data models and technologies

Implementing traceability in manufacturing requires several interacting technologies and information systems. By analyzing relevant traceability technologies and Industry 4.0 models presented in [1, 39, 67, 82–84], a technological structure in alignment with RAMI 4.0 can be proposed as shown in Figure 11. The first layers deal with physical assets and their integration. In many works, the lowest layer purely represents smart or identifiable products, however, in alignment with the RAMI 4.0 model, all types of traceability-relevant objects such as parts, resources, logistic units, or machines should be included in this layer. The communication layer describes the technological processing or middleware structure needed to collect, aggregate, and pre-process the data. Information system architectures at this layer include readers, sensors, and networks for data collection, data transmission protocols, as well as software and services to process the data (e.g. extraction, conversion, distribution, filtering, etc.). The information and functional layer hold the data model, storage architecture, and the resulting functionalities of the traceability system, while the business layer include the application front-end and data visualization.

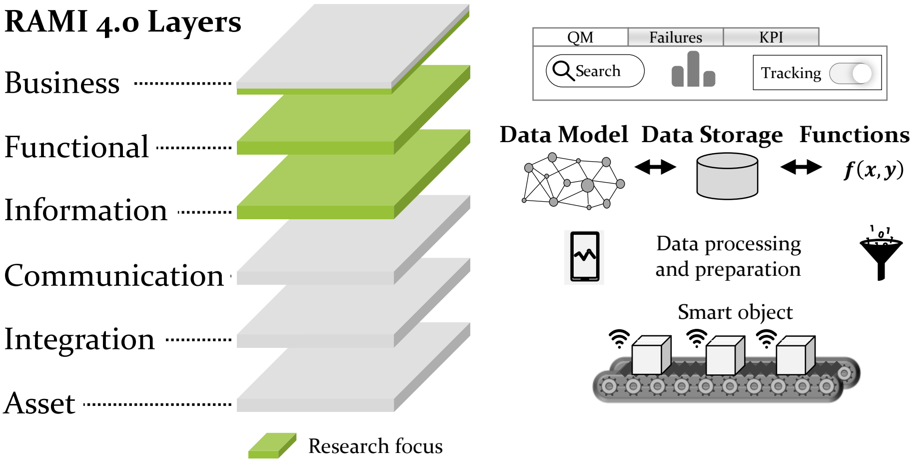


Figure 11: Traceability architecture layers based RAMI 4.0 [83]

While much research has focused on the asset and intregration or communication layer, the focus of this thesis is put on the information and functional layer in alignment with the traceability challenge in wire harness manufacturing and tagging bias discovered in state-of-the-art solutions. In the following sections, a detailed assessment is given by focusing on data models (2.3.1), functions (2.3.2), and data storages (2.3.3).

2.3.1 Data model

In the data model, objects are described using common terminology, like a standard semantic or ontology, and modeled in their relation to other objects. The data model builds the core structure or backbone of any traceability system as it determines the system's objects, scope, and data granularity. Moreover, the data model is the driving force for the subsequent selection and implementation of traceability databases and technologies. A model can be defined as an abstraction of a system achieved through projection, classification, and generalization [54, 85]. Models are required to formally describe and evaluate data systems [86] and are usually characterized by mapping features (a model is based on an original), reduction features (a model only includes a selection of properties), as well as pragmatic features (a model is built for a certain usage and audience) [85].

While different model abstractions exist, such as conceptual, logical, and physical models for relational database modeling, two model conceptions are differentiated for this thesis: the *type model* and the *instance model*. Type models provide a general representation of a system and include universal system aspects derived through object classification, whereby instance models capture a specific system configuration with concrete values and properties and are thus often referred to as snapshot models [85, 54]. The different model concepts are visualized for the example of a wire harness in Figure 12, which was adapted from the models in [85]. Each wire is linked to exactly two connectors, whereby each connector requires the insertion of one wire or more given by the multiplicity constraints (2 and 1..*).

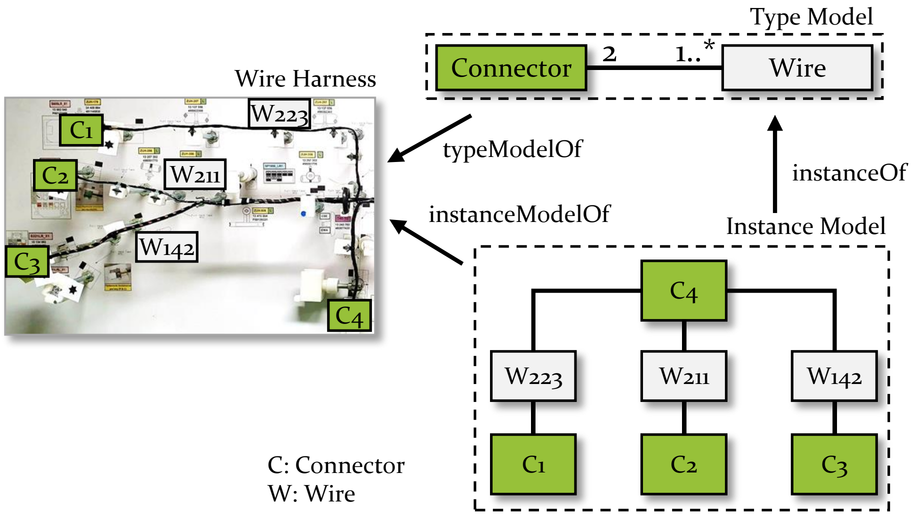


Figure 12: Type and instance data model for a wire harness example adapted from [85]

For the *type model*, classification is used to map instances to model types, which allows grouping instances into classes by abstracting from their different properties [54]. In the shown example, *Connector* is a classification of *C1*. Contrarily, *C1* is an instance or instantiation of *Connector*. A traceability system can thus be holistically conceptualized and formally described by developing appropriate *type* and *instance data models*.

When developing the information layer, it first needs to be decided which objects and adjacent assets need to be considered within the traceability system. The modeling can then be realized through different modeling methodologies and notations, whereby the choice of a modeling method often correlates with the chosen abstraction level and targeted physical implementation (2.3.3 Data storage). Exemplarily, entity-relationship (ER) modeling methods are very commonly applied as they allow a smooth translation into relational database management systems (RDBMS) (e.g. [38, 41, 69, 87]). Entities and relationships belong to sets with common properties. Exemplarily, entity *C1* belongs to the entity set *Connector*. It has attributes that are defined for all objects in the entity set. ER-models can be developed using Chen-notation [87] or unified modeling language (UML). Alternatively, data models for document-based systems are often developed using standard schemas, whereby graph databases apply UML and graph modeling techniques using object nodes and semantic relationships [49]. Independent of the modeling method, the resulting data model determines if and how different data objects are logically connected as well as how the data are translated to a physical structure.

The related work analysis has shown two opposing concepts for process and discrete industries and the analysis of their underlying data models reveals a similar picture. In alignment with the targeted production, process- and product-driven data models are commonly developed, which each put different emphasis on certain data objects and relations, as shown for the type data models in Figure 13.

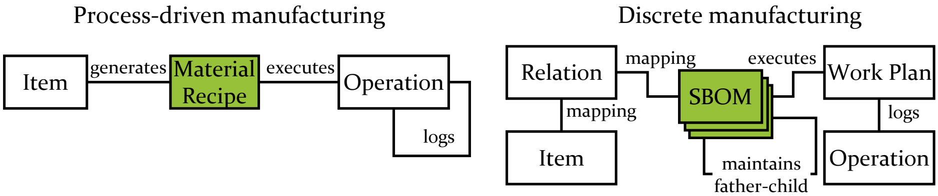


Figure 13: Type data models for process- and product-driven manufacturing

In process-driven data models, a material (item) triggers a machine to conduct a production recipe, which is a defined process formula consisting of sequenced process steps. The recipe then executes those steps, which can be conducted by one machine or over a fixed sequenced production line. Each operation logs relevant quality results, such as process parameters or batch UIDs, which are determined by the recipe's parameter specifications. Exemplary, Jansen-Vullers et al. [38] present a data model for the food industry, in which material lots or batches are linked to operation data based on a production recipe or order. Consumed batches are matched to the recipe's master item. Analogously, Khabbazi et al. [88] develop a very similar data model for lot-based traceability information systems. Product structures can also be stored in the operation log or in a separate item-lot-relation-object. Process-driven models are thus not configured to directly represent complex product structures with multiple input and multiple output hierarchies. For parameters that are recorded for consumed batches and parts, timestamp-based assignments can be applied to associate measurements with sub-parts. This is because these parts do not appear as actual objects in the model but are only as logged parameters within the operation. Process-driven data models have shown to optimize the volume of records in automated and recipe-based processes that generate many parameters [89].

In discrete and assembly-driven manufacturing, the main data entity is taken over by the SBOM. An SBOM-centered data model focuses on father-child relationships which are maintained through the consumption of parts to new hierarchies [27]. Example data models were developed by Zhou et al. [90] or Tang and Yun [91] whose data models focus on complex product hierarchies through a set of interlinked SBOM objects. The model centralizes product data and links different quality data instances to their adjacent product hierarchies [91]. The method is optimized to store father and child relationships, allowing to generate as-built history data quickly [89]. In product-driven data models, data bridging often relies on product UIDs throughout the product tree [57]. As product-driven data models are applied for multi-tier or multi-site production chains, information can be lost when quality data are not coherently linked to UIDs [40]. Product-driven data models have recently gained more interest to help build so-called digital twins and digital shadows for complex assembly structures by associating design data, process data, and shop-floor data to their corresponding SBOM elements throughout life cycle stages and status transformations [73, 92]. Exemplarily, Zhuang et al. apply product-centric and SBOM-driven data models to generate digital twins for airplane manufacturing [73]. In the avionic industry, parts are comparatively expensive and safety-critical, so that a linkage of data to parts with UIDs has shown to be feasible.

The chosen data model is hence influenced by the data scope, functional goal, and industry characteristics. The choice determines the ease with which the physical system can be virtually represented as well as the functional scope and data granularity, that can be realized. It is worth noting that complex and customized assembly industries tend to product-driven data models and industries with a sequenced production with a few variants per line tend to process-driven data models. Regardless of the model type, any chosen model needs to consider which data objects and linkages need to be realized for the traceability use case. A typical phenomenon is that a resource-supervised operation software logs all its parameters and measurements while an MES solution maintains the product configuration so that certain measurement results can not be related to a specific product.

Data relations can thus become difficult to reconstruct during manufacturing if not inherent to the data model.

In addition to defining objects and their associations, recent data models stress the importance of the semantics of object relationships, which can be directly used in the later software and database structure for querying and knowledge discovery [93]. Moreover, a higher focus on bridging data of different domains can be observed [83, 93, 94]. Xu et al. [59] and Zhuang et al. [73] stressed the importance of modeling the dynamics of different lifecycle phases by integrating static models like product structures and dynamic models like time-sensitive process states to a more holistic model picture. A well-established object schema for the information layer is the so-called *product-process-resource* or *PPR schema*, which classifies manufacturing data objects into three domains. The product domain should be a minimal representation of the product containing only its intrinsic elements [94]. Intrinsic elements are considered to be the component and assembly composition as well as object attributes such as material, color, or IDs [94]. The process concept within the PPR schema includes more dynamic data such as time-sensitive events. Resources within the PPR concept are a general description of means or assets that are needed to carry out certain processes or achieve defined product levels, such as humans, tools, machines, or manufacturing areas [94, 95].

To achieve a standardized meta description and definition of the PPR data, different data standards (e.g. eclass), manufacturing ontologies, and semantics were developed [32, 70, 72, 93]. However, interoperability based on the ANSI/ISA-95 (IEC 62264) standard framework [96] has become the most widely accepted approach for manufacturing data. The ANSI/ISA-95 standard describes the terminology, object models, and attributes for software, which operates in between ERP and control systems such as MES or traceability software. It can be assumed that an ISA-95 compliant model describes a representative data structure for the investigated traceability use case. To analyze the semantics and investigate the standard's applicability, Figure 14-Figure 16 show an example of an ISA-95-compliant type model for a wire harness production module using unified UML notation and ER modeling technique. To delineate the PPR elements, product entities are marked in blue, the process elements are visualized in yellow and the resource elements are framed in a gray. The logic of reading UML-based data association, inheritance, dependency, or aggregation, can be found in the key in Figure 15, part 2.

According to ISA-95, products can be modularized through *product segments*. The PM (module of the wire harness) is thus classified as a *product segment type* and links to the wire harness, which is represented by the *product definition* entity. The wire harness is linked to a general BOM as well as a 150 % manufacturing BOM and an assembly BOM. The PM is specified through twelve materials of different classes (*material specification*) like twisted wires, connectors, tubes, or grommets. The material is further attributed by some main intrinsic characteristics as visualized in part 2 and part 3. All product objects, material specifications, and BOM objects make up the product domain of the model (blue visualization). The *process* domain is included through process segments, visualized in yellow. The example PM is linked to eight process segments, which are conducted in a defined sequence, visualized in part 1. In the given model, a PM can thus be associated with a set of defined processes that describe the production logic as planned. The resource domain is included using several specification objects, such as *work cell*, *personnel,* and *physical assets* as depicted in part 3. They describe the resources needed to produce the PM.

The PM data model demonstrates that the ISA-95 standard provides the terminology to model different objects over different hierarchy levels for each PPR domain. *Resource* objects, exemplarily, can be distinguished by equipment classes of different hierarchy scope such as enterprise, site, area, process cell, production line, unit, or work cell. Analogous hierarchy concepts are used for *product* objects, which are described by their composite segments, BOMs of different perspectives, as well as assembly, material, and lot classes. *Process* objects are managed in the same way by distinguishing different hierarchies and business scopes, such as definition, schedule, and performance.

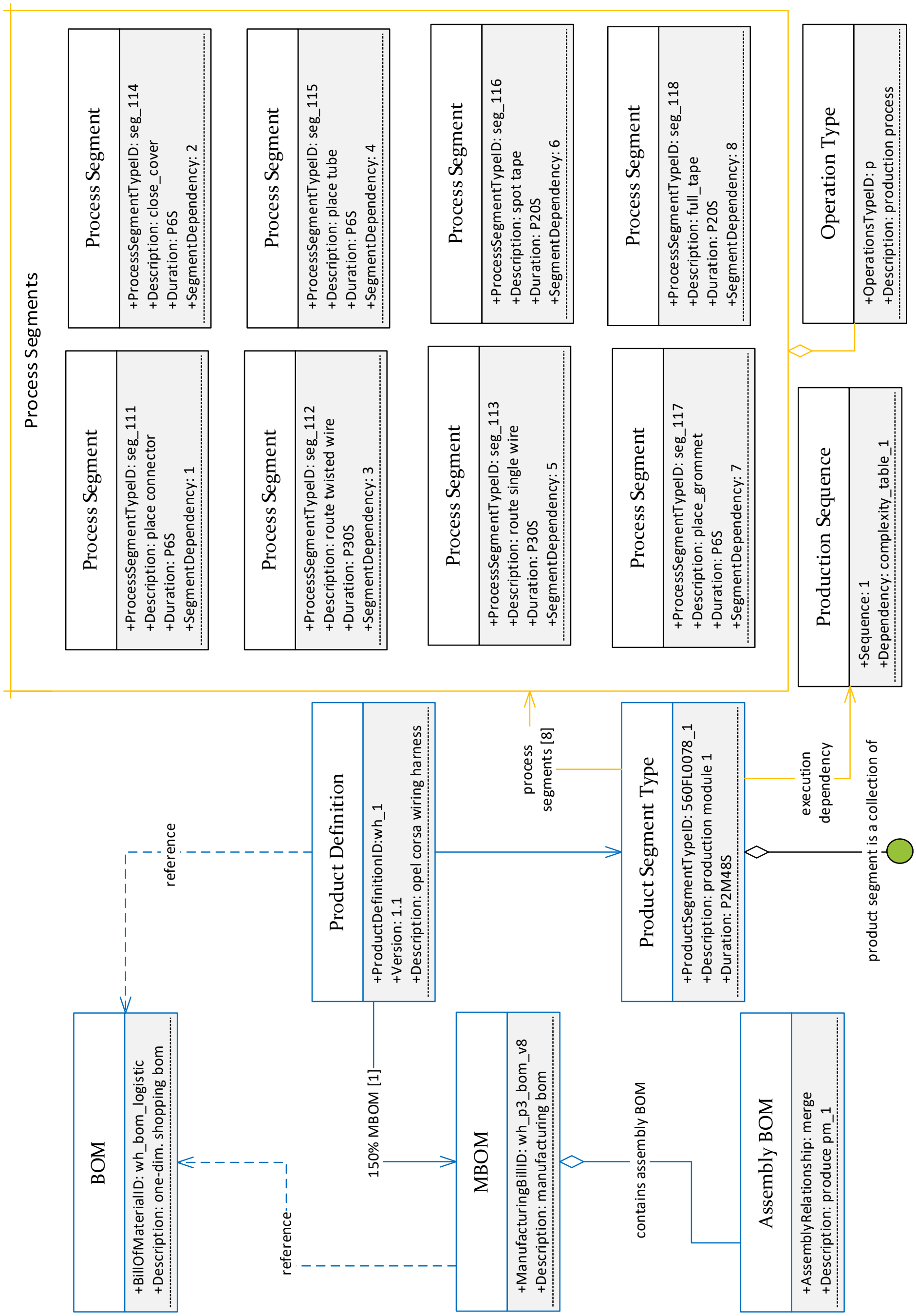


Figure 14: ISA-95 compliant UML-model for a wire harness module, part 1

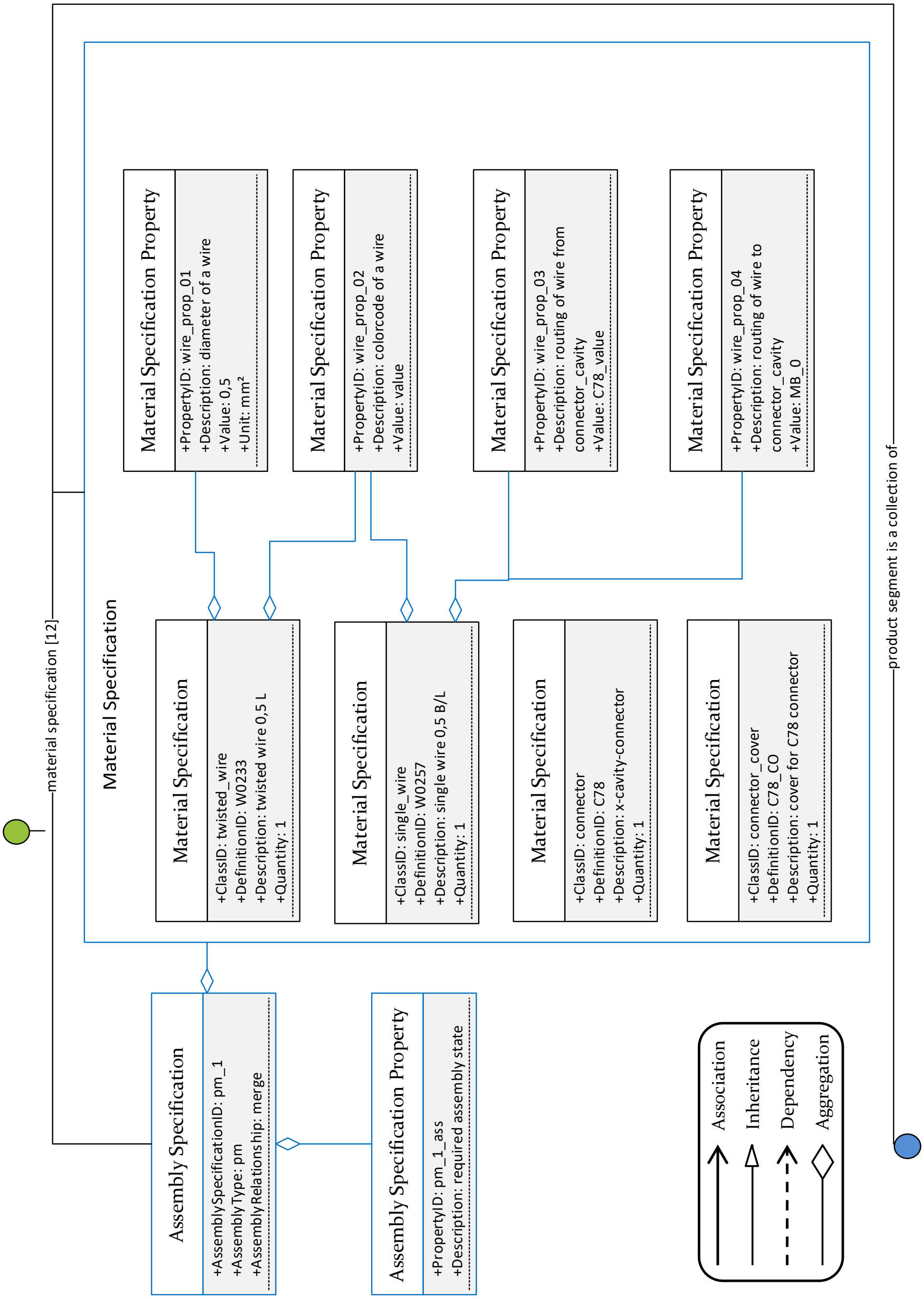


Figure 15: ISA-95 compliant UML-model for a wire harness module, part 2

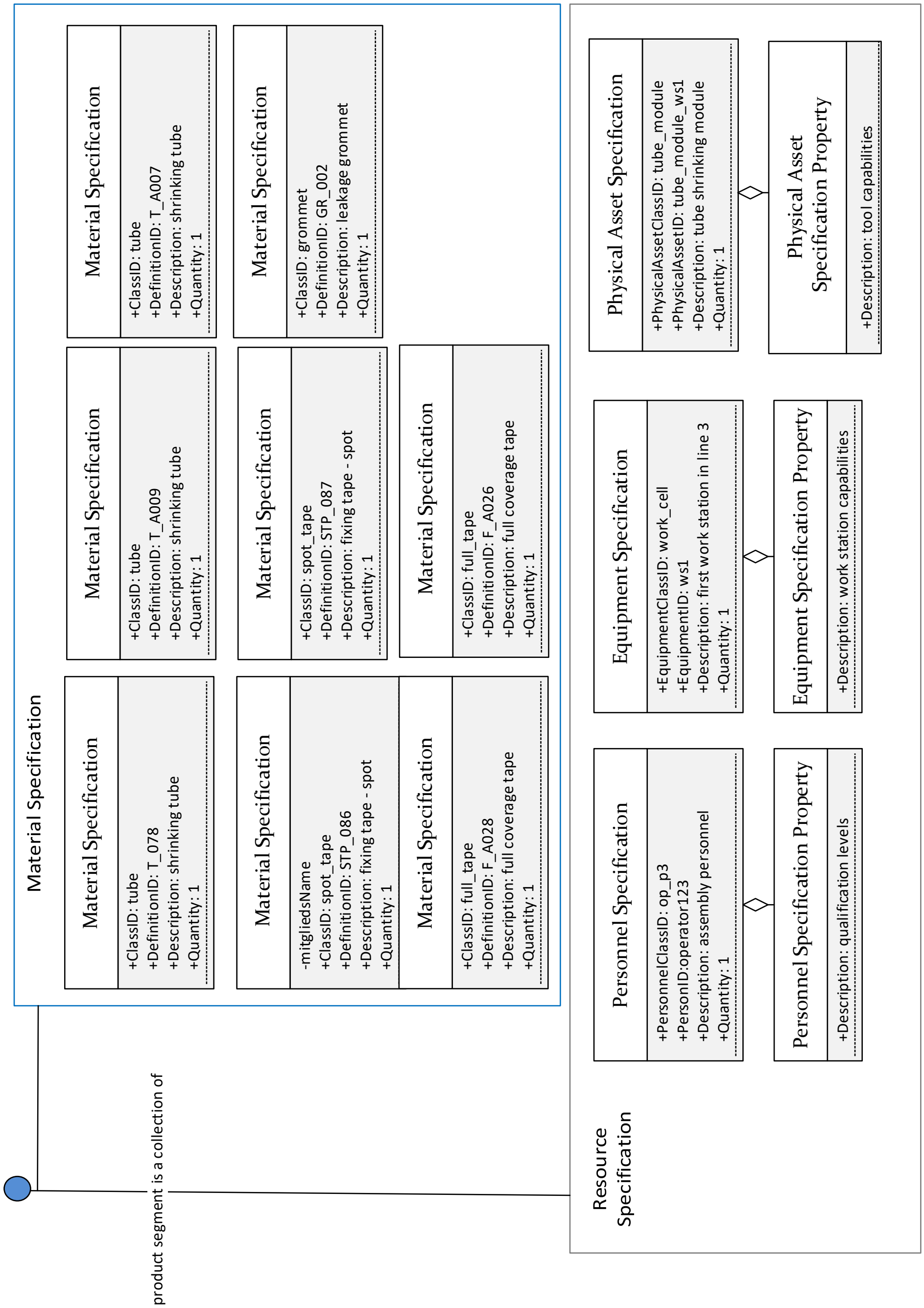


Figure 16: ISA-95 compliant UML-model for a wire harness module, part 3

By assessing the applicability of the standard and schema, the following aspects can be brought forth:

- The standard provides a general description of object types using simple terminology in alignment with the PPR schema.
- It has a broad industry application and therefore allows for seamless integration of MES with ERP and manufacturing control software.
- It incorporates different domain abstraction levels as well as business perspective levels, such as object definition, object execution, and object schedules.
- The open framework of objects and classes with general properties allows customized object modeling according to the use case.
- Production tracking and performance monitoring are promoted as enabled functions within the ISA-95 scope.

If it is assumed that a data model is built according to the ISA-95 definitions, traceability needs to be achieved using the model's objects and semantics. After defining the product, as it was demonstrated for the example PM above, traceability-relevant data need to be collected and linked to the definition model. Within this thesis, a set of modeling tests were conducted with the aim of creating a traceability record, for example, for the illustrated PM. In the test series, the *definition* objects are connected to *response* objects, which include time and state attributes, while *specification* objects are linked with *actual performance* objects, which refer to the actual properties recorded. Exemplarily, a specified wire diameter needs to be linked to a physically measured diameter. Alternatively, a traceability record can be achieved using the so-called *batch production records*, which is an unstructured collection of *definition*, *performance*, and *scheduling* objects as well as *workmasters*, *batch recipes*, *sample*, and *event* data aggregated for a specific batch. When applying the ISA-95 standard and the ER-modeling technique, some challenges could be observed:

- **Object definition focus:** The focus of the standard is primarily on the description and specification of object properties and terminologies. Performance or batch records, which could be used for the analysis of traceability data, are a simple collection of possible performance-relevant objects. Traceability records are aggregated for a certain operation, product(-segment), or batch identifier, which limits traceability queries to a few points of entry.
- **Ridgid structure:** The performance aggregation data necessitate significant data overhead. The object structure is very rigid, which

is further emphasized by the ER-modeling technique. Dynamic behavior of all objects within the PPR domains, such as time-sensitive transactions and events, is difficult to model with the proposed structure.

- **Lacking connections:** Data relationships have no semantics nor attributes and are modeled as simple aggregation, composition, or association links in class diagrams. Data connections are realized by a loose collection of (isolated) objects for a defined use case. The standard focuses on the entities and their properties. Modeling relationships as inherent semantic characteristics is not possible.

The application of the ISA-95 terminology facilitates integration with manufacturing and ERP software. However, although the framework claims to cover production tracking, the modeling tests suggest that a simple collection of data objects with fixed entry points does not provide the aspired level of flexibility and granularity. Moreover, dynamic behavior, patterns, or complex relationships are difficult to conceptualize with the proposed rigid structure. To enable traceability in modern and increasingly complex manufacturing systems, more advanced semantics and data models need to be developed on top of the standard terminology. In [93], [95], or [97], the shortcomings are addressed through semantic links that connect different data objects of PPR domains, for example through semantic relationships (e.g. process *is performed by* resource). However, as those models do not focus on the traceability use case, the number of maintained relationships across domains remains limited. Another reason for the low degree of connectedness could be the usage of ER-modeling techniques or their implementation in RDBMS and exchange formats like XML, which often lead to data models with low connectivity and entity-focused data structures. Accordingly, the commonly used RDBMS modeled with ER-technique might need to be reconsidered. In software traceability, some authors suggest developing connected data models through graph-based modeling [65, 66]. Graph data models can be type or instance models that are designed as directed, labeled, property graphs or generalizations of them [98]. They are applied when the information about data relations is as important as the information about the data entities and allow for a more natural modeling view on data [98]. Graph-based models have shown to be superior in representing complex and heterogenous object relationships using semantically rich structures [99]. As shown in Figure 17, the type model and instance model enable a close translation between concept and instantiation.

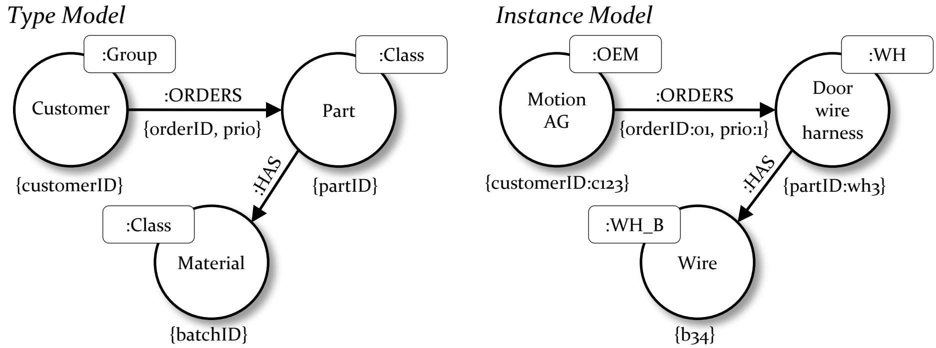


Figure 17: Graph-based type and instance model

Moreover, the connections between data object not only exist during conceptualization but are maintained for the implemented model as semantic relationships. Together with the advancements of graph-based storage systems for industrial applications (see section 2.3.3), graph modeling could thus be effectively used to address the current shortcomings in developing manufacturing traceability systems. Due to its promising features and the shortcomings of ER and RDBMS, the graph-based approach is pursued further in this thesis. The development and explanation of the modeling logic and notation chosen for this thesis will be provided in the methodology sections (chapter 4).

2.3.2 Functions

The previous chapter demonstrated the importance of creating appropriate data models, which support the aspired traceability use case. To enable the information layer's functions, the system's aspired information retrieval requests need to be developed. *Functions* are implemented as information retrieval procedures and algorithms and determine the rationale for data aggregation, association, and analysis [37, 38]. Chen differentiates between basic information retrieval concepts for ER-models, which are: (a) selection of a subset of values from a value set, (b) selection of a subset of entities from an entity set, (c) selection of a subset of relationships from a relationship set, and (d) selection of a subset of attributes from an attribute set [87]. Example information retrievals could be *the length of the wire with a wire ID of w223* (F1) or *the machine on which the wire w223 was produced* (F2). F1 would thus correspond to a selection of a length-value from a wire entity with an ID w223 which belongs to the concept (a). In F2, a machine-product relation is selected from a relationship set that can be assigned to concept (c). After developing all data retrieval functions through appropriate algorithms, they can be provided to the end-user over the functional layer of the framework in Figure 11. This allows users to trigger functions intuitively and have the results displayed as easy to interpret visualizations.

From a traceability perspective, data retrieval functions are often complex and can stretch beyond different data systems and models. Exemplarily, Agrawal et al. [41] define a set of traceability functions for an ER-model which encompass multiple relational databases. The functions include pedigree queries to retrieve supply chain data, location-retrieval queries, and BOM queries to derive father-child relationships for an object [41]. While traceability data models and storage systems are commonly discussed in the literature, functions are often not formally defined. Moreover, the presented functions focus on relational systems (e.g. [41, 87]). The recovery of data relationships in RDBMS requires several JOIN queries, which combine data of different domains to a new table. For highly connected and interrelated data models, JOIN queries have shown to cause performance and functionality shortcomings [100]. Accordingly, functions need to be systematically defined to derive which and how the traceability results need to be provided to the user. Moreover, functions for connected data models are needed to align to the addressed use case of complex manufacturing.

2.3.3 Data storage

In alignment with the model and functions to be achieved, the traceability data need to be implemented and maintained through appropriate data storage systems. There is not a single traceability storage technology, but a whole range of data management and storage systems exist. This section thus gives a short overview of the most common storage systems, which are relational storage and document-based storage. Moreover, upcoming traceability storage systems such as graph-based storage and blockchain-based storage are discussed.

For manufacturing traceability as defined in the framework in Figure 10, the dominant data storage systems are relational storages. RDBMS, like MySQL or Oracle databases, maintain data in a tabular structure, whereby the columns determine the entity's attributes and the rows represent the corresponding values or records [101]. Data objects are thus represented by a table record with the columns functioning as the object's attributes, while linkages to other objects are generated by database keys [101]. As shown in the example of Figure 18, the relational system stores an instance of a wire in a table called *part* with the part's identifier as the database key. Using that key and the two tables *order* and *work unit*, it can be derived which order belonged to the part and on which workstation it was produced. The vast majority of traceability publications implement their traceability data model in an RDBMS (e.g. [38, 39, 69, 71, 91]). While relational systems are very powerful in managing rigid objects and data records, they are ironically ineffective at handling data relationships [84]. Thus, the relationships are not an inherent characteristic of the data structure but exist only at modeling time [P6].

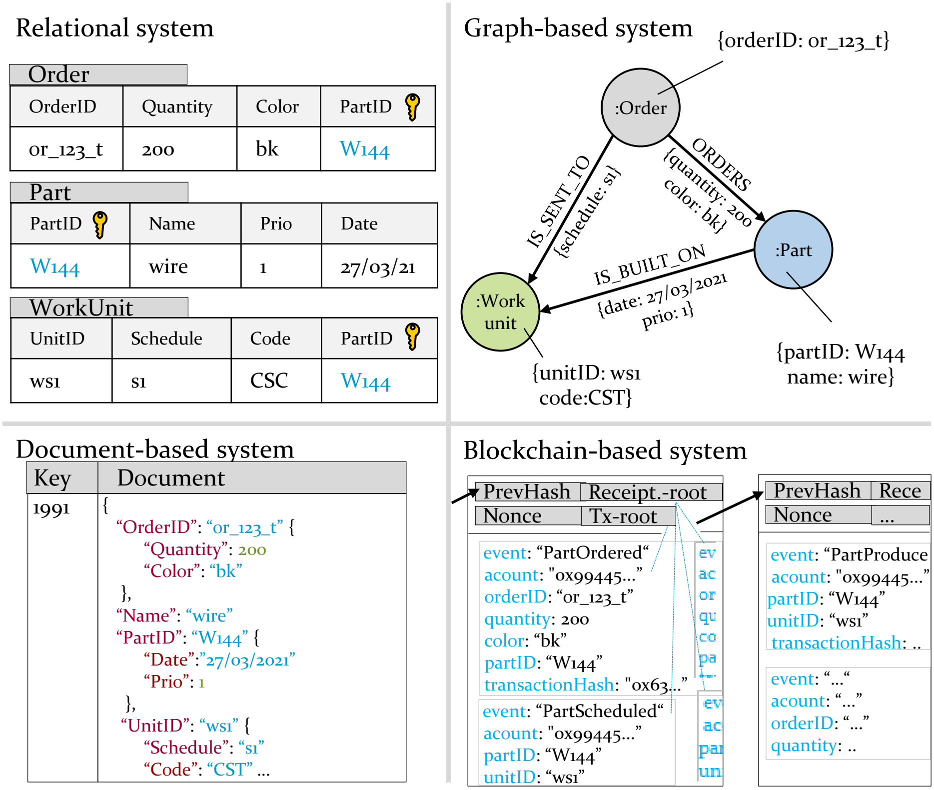


Figure 18: Schematic differences of relational, graph-, document- and blockchain-based storage systems

An alternative to RDBMS are document-based data storage systems which are used to optimize the flexibility and structure of conventional systems [102]. Document-based storages can store any kind of document, which can be referenced over a document key. While the document itself can encompass the same information as the relational system, the structure of how the data are maintained is different as shown in Figure 18. Document storage is not very efficient in managing complex or semantic data relationships and is thus not ideal for the manufacturing traceability use case. For supply chain traceability, however, document-based storage is more common and achieved through standardized exchange documents, such as TraceCore, an XML-based document for high-level traceability data [103].

Relational databases have been used since their introduction by IBM in the 1970s and document-based storage systems have gained increasing interest in the 1990s with the uprise of document formats like XML and suitable storage systems. With the introduction of the graph database Neo4j in 2010, graph-based storage has become an alternative for maintaining data in all kinds of business contexts [104]. In a graph database, data objects are stored by vertices (V) or nodes while data relationships are maintained as edges (E) [105]. The edges can be directed and provided with properties, making the data relationship inherent to the overall structure. Graphs are hence effective where the relationship between entities is a driving force in the design of a database and where information about data interconnectivity or topology is as important as the data itself [100]. The most common graph model is the labeled property graph [106]. Labeled means that objects are grouped into domains (:Order, :Part, :Work unit), and property means that attribute-value pairs are used to add qualities to nodes and relationships. To query the graph and to write commands to the database, the declarative and open-source graph query language *cypher* is most commonly used [107]. Cypher was inspired by expressive querying so that its constructs are human-readable based on English prose (e.g. ORDERS, IS_SENT_TO, etc.) [107]. In manufacturing, concepts based on graph databases arose with the increasing maturity and prominence of Neo4j. Exemplarliy, Huang et al., developed a semantic knowledge management system for PLM based on Neo4j [84]. Moreover, Neo4j has been used to develop data-driven maintenance services in manufacturing [108].

In the review of graph-based systems in manufacturing, Weise et al. [100] found that graph models are used to plan assembly structures or analyze geometric relations of assembly parts in the design phase. However, works that achieve an actual implementation of their data are still rare in the manufacturing domains. The authors further criticize the lacking account of product variability in the available models. They further discuss possible reasons for the found shortcomings, which might be the novelty of the technology or the high complexity of implementing graph-based systems based on the available data. [100]

As the fourth relevant storage technology, this thesis discusses blockchain technology. The following paragraph is a close textual excerpt from the author's publications in the *Journal of Manufacturing Systems* [P7] and *Computers & Industrial Engineering* [P8]: Blockchain has emerged as an agglomerated system of information technologies and protocols, which enable promising features for traceability applications, such as maintaining a trusted and immutable ledger of data records and transactions in a peer-to-peer environment. Based on the primary work of Satoshi Nakamoto (Bitcoin) [109] and Vitalik Buterin [110], a range of blockchain-based traceability systems have been proposed for different industries. Early concepts highlighted the technology's suitability to facilitate higher supply chain transparency and data integrity and drafted first models for digital agent management and blockchain-based data entry across supply chain units [111, 112]. Most of the previously proposed blockchain-based traceability solutions integrate object or asset data as primary data, events, or even off-chain data, however, physical assets could also be represented using so-called blockchain tokens [110], which enable a direct virtualization of physical assets to the blockchain domain. Depending on the blockchain design and framework, such as Bitcoin, Hyperledger Fabric, or Ethereum, data can hence be stored and maintained differently. As shown by the example in Figure 18, an Ethereum-based storage encompasses logs of recorded transactions and emitted events. Using asymmetric cryptography and hash algorithms, events and transactions are verified and packed into blocks through consensus protocol in a process called mining. These events essentially build a database in the blockchain, which can be queried for traceability relevant information. As each block contains the hash of the previous block, a chronological chain of data is formed, providing a clear and traceable history of transactions. The hash values not only function as a linkage between blocks but the integrity of the related data can be proven, as it will remain its value as long as the original data remain unchanged. Due to the blockchain characteristics, the storage technology is especially suitable for decentralized applications, such as supply chains. [P7, P8]

Looking at the traceability storage systems, it becomes evident that different technologies can be applied to achieve traceability in manufacturing. Each of those storages comes with its advantages and disadvantages as well as characteristics of how data and data relationships can be represented, which further determines which traceability functions can be built for the system. The choice of traceability storage thus needs to support the defined data model and functions, while simultaneously adhering to industry specifications and requirements. For this reason, a combination of multiple technologies may be needed to cover the entirety of the data model. An additional aspect that needs to be considered is the degree of centralization or decentralization required for the chosen use case. As derived by the complexity characteristics and the assessment of the wire harness production flow, complex manufacturing systems require a production network structure consisting of several participating companies or factories. It can thus be assumed that the final product is produced by several production sites, which requires a data storage that provides transparency and data integrity in a decentralized environment. Simultaneously, each's company detailed manufacturing data encompass sensitive intellectual property (IP) about products, resources, and operations, which companies do not want to disclose with the entire manufacturing network. The chosen storage technologies thus need to enable transparency for the shared aspects of the data model across all participating companies, and protection of the detailed data of each's company production. A mixture of centralized storages for manufacturing-internal data and decentralized storages for network overarching data is thus reasonable.

2.3.4 Shortcomings of the state of the art

The previous sections demonstrate the related works in traceability development and the existing data models and technologies, on which traceability systems are built. This chapter summarizes the shortcomings of the analyzed state of the art, which serves as a reference point for the development of the traceability methodology and concept. The shortcomings are discussed with regard to the application (a), methodology (b), data model (c), and data storage (d).

From an application perspective (a), it can be derived that there are no traceability models or concepts, which address the here targeted wire harness industry. Moreover, there is a general lack of solutions for industries with analogous characteristics, as complex manufacturing flows with a multi-level product structure produced in a non-linear and mass-customized production flow are not sufficiently addressed from a traceability perspective in current solutions [93]. Traceability systems designed for linear and deterministic production flows have shown to be not transferable to complex manufacturing environments [113], so that novel solutions are required. This thesis thus aims at developing a data model for complex manufacturing flows (section 5) and implement it through suitable traceability storage technologies for the wire harness use case (section 6).

From a methodological point of view (b), the analysis of the state of the art has revealed that there exists no systematic methodology for the development of traceability systems in manufacturing. While the research field of software traceability purely focuses on the development of modeling methods, terminology, and semantics, manufacturing traceability usually presents tailored data models and storage solutions for a specific industry application. Each application thus focuses on a defined domain or traceability goal and provides a customized technological system [32]. This not only complicates traceability system development but also prevents effective transfers of systems between applications as well as system adaptions in the light of new requirements or changes. The systematic modeling approach in software traceability, which relies on a uniform description of traceability terminology and the use of systematic abstractions mechanism to model different levels of granularity, has boosted the research in this field [49, 54]. In this context, traceability advocates in software engineering have constantly demanded a model-driven development approach with a focus on granularity hierarchies and the inclusion of data semantics across domains [49]. Manufacturing traceability thus could benefit from an analogous modeling methodology, that systematically determines which objects need to be traced, which links are to be achieved, and which functions need to be realized. This gap is to be addressed through the development of a system modeling methodology in section 4.

As a third shortcoming, current research lacks suitable data models (c) for the defined use case. This shortcoming was discussed earlier and has its roots in MES development, which either focuses on a process- or product-driven data collection. MES can thus fall short on collecting all traceability data for a hybrid production system consisting of process-driven and product-based steps. Moreover, the over-emphasis on physical product marking exacerbates holistic traceability for product structures, in which not all subassemblies are physically itemized. Models designed based on unique identifiers thus contribute to the traceability bias that anything that is not marked could not be traced. The data model challenge thus has an overlap with the application challenge (a). This thesis aims to demonstrate that a systematically derived data model can enable traceability for both, product- and process-driven manufacturing flows without the need to fully rely on identification tags. Another shortcoming is that current models mostly focus on static planning data, whereby dynamic transformations are not within the scope of most model structures. The data model developed in chapter 5 hence incorporates static and dynamic elements to provide a holistic traceability picture.

From a data storage perspective (d), current applications put a high emphasis on relational storage technologies in manufacturing and exchange formats across organizational borders. The relational model promotes a high degree of data independence [87], which is not the goal to be achieved in traceability applications. Data relations are thus not realized within the model but in the business or functional layer so that the data remain rather disconnected [104]. Moreover, the modeling of relational systems requires a translation between type and instance models (or logical and physical models) so that relations that exist in the concept phase are not inherent to the physical model. While manufacturing traceability focuses on centralized RDBMS, data sharing across manufacturing units is accomplished with document-based exchange procedures or cloud-based relational storage systems. The storage system thus not only raises the challenge of data representation but also of data centralization, sharing, and exchange. Tracing products over multiple tier manufacturing networks and supply chains thus relies on shared data that need to be mutually trusted and tamper-proof [93], which becomes especially relevant in the context of a recall or liability claim. Through the proposed traceability methodology, this thesis derives suitable traceability technologies to provide traceability data within manufacturing and across manufacturing units (chapter 6). Based on the technological implementation, this thesis further assesses the proposed storage technologies with regard to performance and functionality (chapter 7).

3 Research framework

The research framework outlines the steps that need to be taken to conduct a research project and is usually presented as a visual schematic [114]. Figure 19 shows the research framework of this thesis which consists of four major steps: the development of the modeling methodology, the development of the traceability reference model, the implementation of the traceability model, and the evaluation of the traceability model. Through these four aspects, the posed research questions are to be answered and solutions to the shortcomings of current works are to be found.

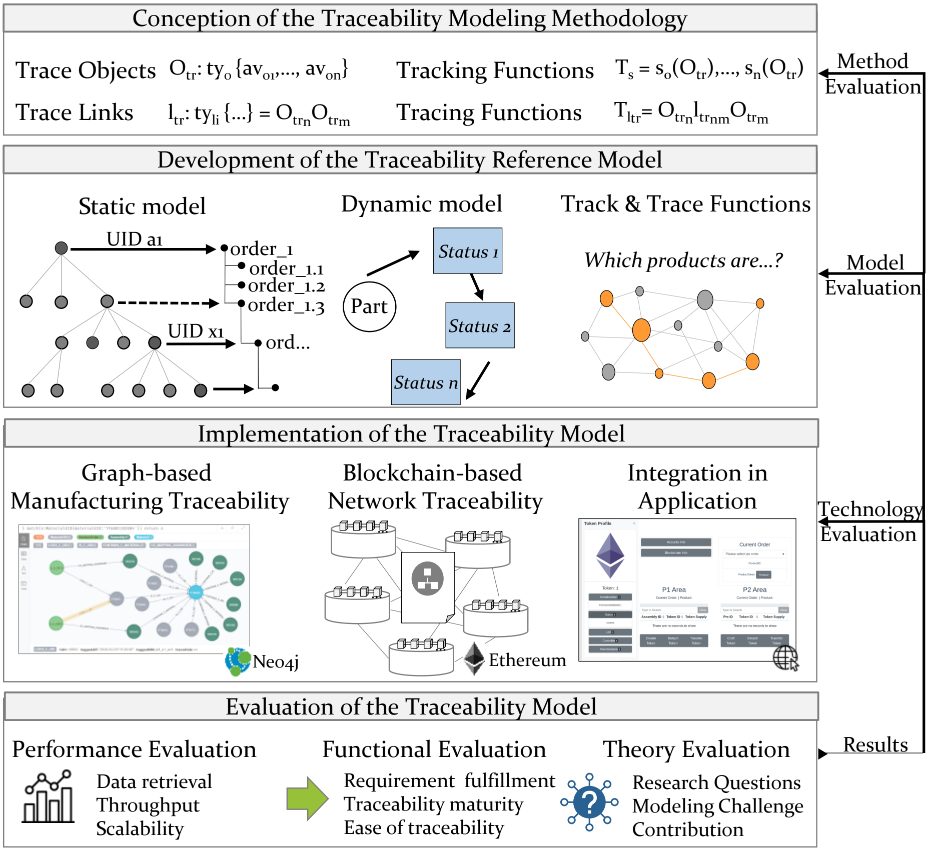


Figure 19: Research framework of this thesis

The development of the traceability modeling methodology is conducted in chapter 4. It defines the relevant traceability building blocks such as trace objects, trace links, as well as tracking and tracing functions and enables a more systematic approach towards traceability development in alignment with the mature research field of software engineering.

The traceability reference model, which is described in chapter 5, is developed on top of the traceability building blocks. The model defines the data objects and relationships for complex manufacturing systems from a static and dynamic perspective. The reference model conceptualizes how the derived shortcomings of the tagging bias and hybrid production logic can be effectively overcome. As it focuses on the general traceable objects, their interactions, and aspired functions, it serves as a blueprint for the concrete model implementation. Moreover, as the model is developed using a graph-based modeling technique, it allows defining semantic relationships as an inherent feature of the traceability system.

In chapter 6, the traceability model is implemented for a concrete wire harness traceability data set. For the implementation, a graph storage based on Neo4j is chosen to represent the detailed manufacturing data across all object and link types. To virtually represent the product configuration and product states across manufacturing units, a blockchain-based supply chain traceability system is developed. For the blockchain system, Ethereum is used as the development framework, and products are represented through a token-based approach. Both systems include static and dynamic aspects. They are integrated over a web application, from which the user can interact with the systems.

The fourth aspect of the research framework addresses the evaluation of the methodology, reference data model, and technological implementation. The evaluation, which is conducted in chapter 7, includes a performance, a functional, and a theory section and derives the overall contribution of this work.

4 Traceability modeling methodology

The assessment of the state of the art has shown that developing traceability systems for a specific use case often leads to narrowly defined traceability solutions with limited transferability and scalability. Learning from the success of the research field of software traceability, a traceability solution should instead be conceived by inferring individual qualities from general schemas and models. Accordingly, this thesis proposes a systematic traceability modeling methodology that allows defining traceability systems on an abstract level before implementing the technology-specific solutions. The proposed method is based on the most acknowledged methodological work in traceability research, which has been conducted by Ramesh and Jarke [49], which was then further adapted [48, 54, 55, 75, 115]. The methodology of this thesis transfers the approaches of [48, 49, 54, 55, 75, 115] to the manufacturing domain and has been partly published in the *International Journal of Computer Integrated Manufacturing* [P6]. As shown in Figure 20, it starts by defining the *traceability building blocks* (section 4.1) from which the *traceability reference model* is then built (section 4.2).

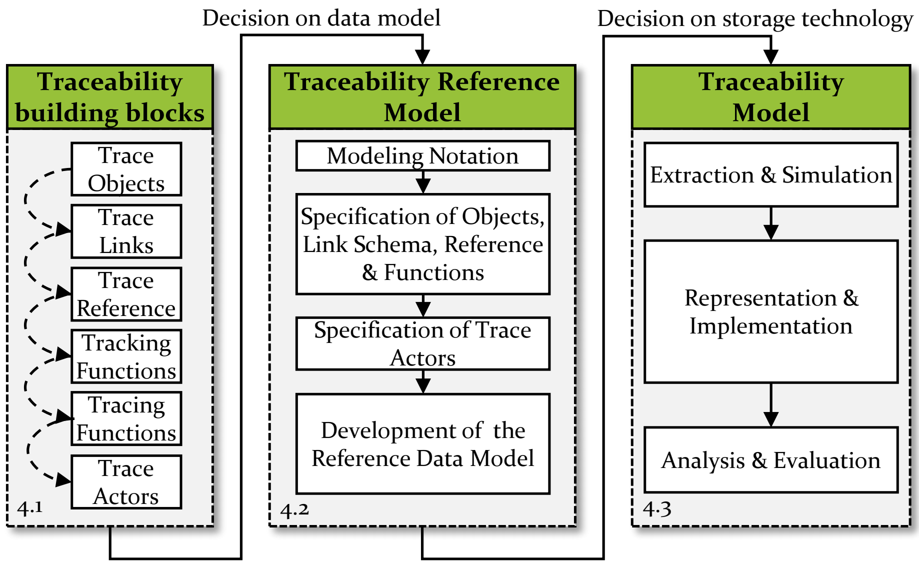


Figure 20: Traceability modeling methodology

Based on the reference model, the application-specific traceability model can be developed and implemented in a chosen storage technology (section 4.3). Technology implementation thus occurs after the formal definition of all aspects of the traceability system, which implies that the reference model can usually be implemented by more than one storage technology.

4.1 The building blocks of traceability

The *building blocks* of a traceability model are characterized as the basic ingredients that make up any traceability system. Those building blocks, which encompass *trace objects*, *trace links*, *tracking* and *tracing functions*, as well as *trace references* and *trace actors*, are the core elements to be defined during traceability modeling. First conceptions of trace objects, trace links, and tracking and tracing functions were published in [P6] and [P5].

Trace objects

A traceability model consists of *trace objects*, which represent the physical or virtual artifacts that are to be tracked and traced. Each trace object (O_{tr}) belongs to a specific domain or type (ty_o) and can be specified through attribute-value pairs (av_o), as defined in (1) [P6].

$$\begin{aligned} &O_{tr}: ty_o\ \{av_{o1}, \dots, av_{on}\};\ av_o = (attr_o\text{: } value_o) \\ &e.g. \rightarrow W223: product\ \{color: green, cs: 0.5mm^2\} \end{aligned} \tag{1}$$

Exemplarily, wire *W223* belongs to the object type *product* and is attributed by its green color and its cross-section (cs) of 0.5 mm². For each traceability system, the trace object types need to be specified (2), building a tuple or set of eligible object types ($ty_{o_{set}}$).

$$\begin{aligned} &ty_{o_{set}} = \{ty_{o1}, \dots, ty_{on}\} \\ &e.g. \rightarrow\ ty_{o_{set}} = \{p, r\} \end{aligned} \tag{2}$$

In contrast to software engineering, manufacturing trace object types can refer to physical objects such as products (p) and resources (r), as shown in the example in (2). Moreover, virtual objects, such as requirements, goals, measurement parameters, or design specifications could be defined.

Trace links

The relationship between two independent trace objects ($O_{tr_n} O_{tr_m}$) will be referred to as the *trace link* (l_{tr}). Trace links describe *explicit* and *semantic* associations between trace objects. These associations can be further specified with attribute-value pairs (av_l), as stated in (3) [P6]. Exemplarily, a trace link associates the resource trace object *M1* and the product trace object *W223* as follows: M1 produces W233 with an operation time (t) of 5 s.

$$l_{tr}: ty_{li}\ \{av_{l1}, \dots, av_{ln}\} = O_{tr_n} O_{tr_m};\ \ av_l = (attr_l: value_l)$$
$$e.g. \rightarrow PRODUCES: pr\ \{t: 5\ s\} = M1\ W223 \tag{3}$$

During trace link definition, suitable link types (ty_{li}) are identified for the traceability model, leading to a set or tuple of eligible link types ($ty_{li_{set}}$), as stated in (4). Links of a certain link type have common properties, such as semantics and common attribute sets, and should correspond to a particular data domain. In the example in (3), the link type is defined as a product and resource association *pr* and attributed by the operation time.

$$ty_{li_{set}} = \{ty_{li1}, \dots, ty_{lin}\} \tag{4}$$

Tracing function

The *tracing* or *trace function* describes the method of observing and retrieving a set of trace objects and links for a certain traceability purpose. A tracing function ($T_{l_{tr}}$), which is defined for all object and link types ($\forall\ O_{tr}, ty_{li}$) starts at a trace object of interest (O_{tr_1}) and then traverses relevant objects and links until all target objects and links have been reached (5) [P6]. Using the illustrations from above, the tracing function could for example derive all associated production resources for a given product by traversing the "PRODUCES" trace links.

$$\forall\ O_{tr}, ty_{li}\quad T_{l_{tr}} = O_{tr_1}\ l_{tr_{12}} O_{tr_2}, \dots, O_{tr_n}\ l_{tr_{nm}} O_{tr_m}$$
$$e.g. \rightarrow\ T_{l_{tr}} = M1\ PRODUCES\ W223, \dots \tag{5}$$

Trace reference

A *trace reference* (R_{UID}) describes how an instantiated trace object can be uniquely identified within the model, stated in (6). Trace objects with trace references thus represent identifiable physical or virtual objects in the manufacturing flow, classified as *instances* in RAMI 4.0 [83]. For product trace objects, batch and item UID are commonly used as trace references, for example, the batch UID *b34* as an instance of the trace object *W223*. A trace reference can thus refer to a unique item or a group of instantiated trace objects (batch). If abbreviating data paths are created during the production flow, e.g. by splitting or merging batches, a new R_{UID} needs to be created to maintain clear associations.

$$R_{UID}(O_{tr})$$
$$e.g. \rightarrow\ R_{UID}(W223) = b34 \tag{6}$$

For each model, the possibility of uniquely identifying trace objects needs to be carefully considered to evaluate the model's capability to associate data unambiguously. Trace references thus aim at making trace objects identifiable, which can be achieved in many ways, such as physical identifiers, timestamps, geo-stamps, hash-values of measurements, or impedance curves. To clarify, a product represented by a serial number only describes a type of product that can have a multitude of trace references representing specific physical trace object instantiations. Process identifiers, which only describe a type of process, such as *process 1.11 = routing W223 from C1 to C4*, could get trace references by combining timestamps, location identifiers, and process identifiers. However, it is not necessarily economic and efficient to require trace references for all trace objects. During modeling, it needs to be considered which trace objects are crucial for assigning data, thus requiring trace references. The usage of trace references aims at tackling data black boxes while providing the modeling framework to design traceability solutions independent of the product tagging bias.

Tracking function

A *tracking function* (7) describes the method of following and monitoring the status (s_i) of a trace object. Tracking functions are only eligible for trace objects with a trace reference. Accordingly, a tracking function (T_s) can be thought of as a sequenced walk from the initial status (s_0) of an object to the final and most recent status (s_n) [P6], for example, batch *b34* of *W223* can be tracked from its first status *in production* to its final status *final test*.

$$\forall\, O_{tr} \text{ where exists } R_{UID}(O_{tr})$$

$$\begin{aligned} &T_s = s_0(O_{tr}), \dots, s_n(O_{tr}) \\ &e.g. \rightarrow\ T_s = s_0(b34(W223)), \dots s_n(b34(W223)); \\ &\qquad\quad s_0 = in\_production, s_n = final\_test \end{aligned} \tag{7}$$

The status changes thus describe the inherent transformation of an object throughout its lifecycle, which can lead to updates of object properties.

Trace actor

A *trace actor* (TA) defines the operating person, machine, or system that has control or sovereignty over a certain trace object. They can be used to manage and monitor different granularity levels of control over trace objects, such as products, resources, or logistic units.

4.2 From the building blocks to the traceability reference model

Based on the building blocks, the traceability reference model (TRM) can be derived. According to [49] and [75], TRMs represent generally applicable type models for defining traceability relationships and objects. One key challenge in the development of a TRM is to ensure that it effectively represents heterogeneous physical trace objects like products or resources and virtual trace objects like orders or requirements. The reference model thus corresponds to a *physical/virtual* to *virtual* mapping. Software traceability models are always a pure *virtual to virtual* mapping, which facilitates data referencing since virtual objects can be more easily provided with trace references, for example through indexes or identifiers, without the need to consider physical object requirements.

Traceability reference model

The traceability reference model is a type model, which integrates and specifies the traceability building blocks for a defined traceability domain or industry. The reference model functions as an application-independent reference schema, whose development allows to significantly shorten the modeling time of tailored solutions, which has shown to be up to 80 % in software traceability [49, 116]. The efficiency and effectiveness of a traceability reference model depend on the chosen trace object types and link types. Their selection and definition are considered to be the greatest challenge in model building [49, 54, 55]. In manufacturing, the chosen *trace references* further determine, whether data can be undoubtedly assigned to certain objects. The modeling of trace references is hence crucially important to solve the *physical/virtual to virtual* modeling challenge. Due to the superior ability of graphs to represent connected data and traceability relations [75, 106], this thesis proposes to develop traceability models as graphs (G) and formalize them using graph-based modeling notation. A graph-based TRM allows the direct representation of trace objects as nodes or vertices (V) while trace links can be modeled as semantic property-edges (E), as stated in (8).

$$TRM = V, E \ \ where\ V = \{v_1, v_2, \dots\}\ and\ E = \{e_1, e_2, \dots\} \qquad (8)$$

Table 3 specifies the syntax, semantics, and visualization for the TRM. The syntax refers to the valid elements of the model, while semantics describe the rules or meaning of how the elements are represented. The notation defines how the elements are to be visualized. For graph-based notation, several approaches exist [98]. For this thesis, a notation in alignment with [104] and [117] was chosen.

Table 3: Graph-based modeling syntax, semantics, and notation for the TRM

TRM Syntax	**Term**	**Graph**	**TRM Semantics**	**Semantic Example**	**TRM notation (Visualization)**
Trace object	O_{tr}	Nodes/ Vertices	First letter upper case	Wire123	Circle
Trace object type	ty_o	Node label	":" + first letter upper case	:Product	Circle color, text on circle
Trace object attributes	av_o	Node properties	{lower cases with ":" as separator}	length:5	Text on circle
Trace link	l_{tr}	Edges	→/–/←	Wire123 → Crimp5	Directed or non-directed arrow
Trace link type	ty_{li}	Edge type	":" + first letter upper case	:HAS_ CRIMP	Text on arrow
Trace link attributes	av_l	Edge properties	{lowercases with ":" as separator}	{crimped: time-stamp}	Text on arrow
Trace reference	R_{UID}	Node label + property	as defined above	WireUID {uid: b_34}	Circle color, text on circle
Trace object state	s_i	Node label + property	as defined above	State {s1: blocked}	Circle color, text on circle

In the graph-based TRM, each trace object is represented by its own node, while trace links are represented as edges between those nodes. Trace object types are represented as node labels since labels can be used to group nodes into domains or sets. Trace link types are modeled as edge types, which represent the trace link sets semantically. The attribute-value pairs of objects and links are modeled as node properties and edge properties, whereby trace references and states are also modeled as unique nodes with properties specifying their reference or current state. Based on Table 3, Figure 21 shows an example of how the visualization notation specified is implemented for trace objects and trace links. In graph-based visualizations or notations, nodes are represented as circles, whereby labels can be written onto the node and differentiated using different color schemes.

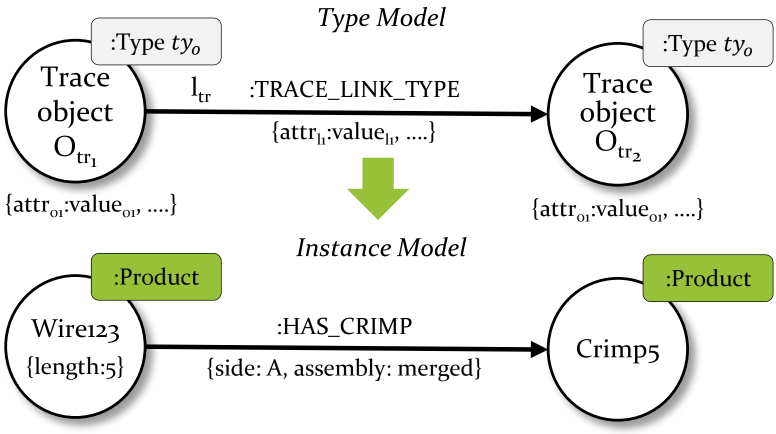


Figure 21: Graph-based modeling notation example for type and instance models

Edges are visualized as arrows with the link types written in human-readable semantics above the arrow in capital letters. Instantiated, the model links the *Wire 123* node to the *Crimp5* node using a "HAS_CRIMP" edge. If the model gets implemented as a graph database, graphs can thus be applied throughout modeling and implementation to achieve granular and connected traceability models [75, P6]. Moreover, the information on trace links can become inherent elements of the data model. This method of consistent modeling and implementation based on graphs constitutes a significant contribution of this work as it eliminates cumbersome translations and allows the integration of complex data relations.

Manufacturing and supply chain TRM aspects

As shown in the traceability framework section (2.2.1), different model abstraction levels can be considered. In alignment with the addressed industry, this thesis distinguishes between the manufacturing traceability reference model (TRM_{MFG}) and the supply chain traceability reference model (TRM_{SC}), which are sub-models of the TRM (9). The TRM_{MFG} provides a detailed view of production data, whereby the TRM_{SC} combines the production view of each participating manufacturing unit to a shared network perspective on the final product.

$$TRM_{MFG} \subseteq TRM, \; TRM_{SC} \subseteq TRM \tag{9}$$

All aspects of the TRMs will be consistently modeled as graphs independent of their technological implementation.

Static and dynamic TRM aspects

As shown in the previous section, the inherent characteristics of a trace object can be modeled using object attributes (e.g. color, length, type). However, trace objects further undergo dynamic status variations and updates throughout their lifecycle. As these status changes are circumstantial attributes and are only valid for a certain period (e.g. product *is blocked* or resource is *in maintenance*), it is not advisable to model them as inherent trace object attributes. This thesis thus distinguishes different dynamics of traceability data. The consistent part of the graph model will be referred to as the static graph (G_s) [P6]. The variable and dynamic parts of the graph model will be referred to as the dynamic model component (G_D) [P6]. The dynamic model is therefore characterized by a significantly higher amount of trace object instantiations, which undergo location and time-sensitive state changes. In the TRM, the corresponding trace objects and links can be identified using *_s* for static and *_d* for dynamic elements. Both, the static and dynamic models thus form sub-sets of the entire model. Since these models are connected at various points (e.g. a certain product is connected to static and dynamic sub-graphs), the static and dynamic sub-models also have a common intersection or model overlap, as shown in (10)-(11).

$$\begin{aligned} G_s &= O_{tr\_s}, l_{tr\_s} = V', E' \quad V' \subseteq V \ \land \ E' \subseteq E \ \land \ G_s \subseteq G \\ G_D &= O_{tr\_d}, l_{tr\_d} = V'', E'' \quad V'' \subseteq V \ \land \ E'' \subseteq E \land G_D \subseteq G \end{aligned} \tag{10}$$

$$G_s \cap G_D \tag{11}$$

As records need to be considered from a manufacturing and manufacturing network and thus supply chain perspective, the TRM_{MFG} and TRM_{SC} both contain dynamic and static model elements.

4.3 From the traceability reference model to the traceability model

Based on the TRM, the actual traceability model for the chosen use case or application needs to be derived. The traceability model (TM) specifies the defined trace objects, links, references, status, and functions for a concrete traceability system while maintaining the overall structure of the TRM. To build the TM, the process steps of *extraction*, *representation*, and *analysis* need to be considered [54] in alignment with the proposed modeling methodology. The first phase of *extraction* refers to the collection and enrichment of traceability information from available sources. In manufacturing, it can be assumed that the envisioned trace objects and links are not available for extraction at the point of building the TM. This may be based on the model being built before line start-up, data not yet being collected because it has not been specified, or previous systems not being designed for traceability purposes. Accordingly, the model should be built through extraction, where possible, and simulation for lacking data to model the data without any gaps. The simulation of synthetic data can usually be achieved with domain experts by generating small programs that create data with analogous properties, volume, and behavior. The benefits of achieving more holistic modeling and verification based on the synthetic data thus usually outweigh the simulation effort and possible errors.

In the second phase *representation*, the data inputs are transformed to the envisioned format, semantics, and connections using an implementation technology. The main technology holding the traceability model will be the storage technology as it maintains the data model and enables the traceability functions. A TRM is therefore open to several technologies and allows different technological implementation. The traceability model representation then corresponds to a concrete technological implementation of the reference model, which can be evaluated from a modeling perspective (How well does this solution support the TRM?) and technological perspective (How well does this solution operate from a functional and performance perspective?). While in software traceability, a high congruence between TRM and TM is usually achieved as both refer to software code objects and associations. The physical implementation in a manufacturing context can lead to significant deviations between the reference model and its implementation. The storage technology often has to fulfill further requirements, such as security or speed, which have to be weighed against the translation congruence to the reference model. Moreover, the different aspects of the TRM (manufacturing vs. supply chain or dynamic vs. static) may result in different technological choices. Exemplarily, the TRM_{MFG} could be implemented using a relational database paired with a streaming log file for its dynamic aspects, while the TRM_{SC} gets implemented in a document-based file system including both static and dynamic aspects.

The *evaluation* takes place in the last step of the modeling phase. During assessment and analysis, the tracking and tracing functions are developed and implemented on top of the data representation. By translating the functions into appropriate querying algorithms, the traceability model's objects and links can be traversed, aggregated, and the query results can be generated. During this last modeling phase, the traceability model can be holistically analyzed, and output visualizations can be implemented.

5 Wire harness traceability reference model

In this chapter, the traceability reference model is developed for the wire harness industry. Based on the assessment of the wire harness and traceability state of the art, this thesis defines four trace object types; *product-configuration trace objects* ($ty_{o1} = c$), *process trace objects* ($ty_{o2} = p$), *resource trace objects* ($ty_{o3} = r$), and *order trace objects* ($ty_{o4} = o$) [P5]. The choice of tracing configuration (product), process, and resource data are in alignment with the PPR model. By adding order trace objects to the traceability reference model, this thesis aims at facilitating the virtual representation of the customized and order-driven manufacturing logic. The wire harness TRM thus holds four eligible trace object types, defined in (12) [P6]:

$$ty_{o_{set}} = \{c, p, r, o\} \tag{12}$$

For *c-objects*, which are trace objects of object type configuration, a high number and diversity of products need to be digitally represented in multi-hierarchical assembly structures while relationships with the corresponding resource, process history, and order objects need to be maintained. Analogously, the model should seamlessly allow identifying direct linkages to the four object types, such as which resources and processes were used for an order or which process results were generated at a certain resource. Accordingly, trace links within and across all defined trace objects are eligible for the wire harness TRM leading to ten trace link types (13) [P6]:

$$ty_{li_{set}} = \{cc, co, cp, cr, oo, op, or, rr, rp, pp\} \tag{13}$$

A link between two *c-objects*, such as the association of super-ordinate and sub-ordinate part, will be referred to as a *cc* link, while links between a resource and process object correspond to an *rp* link, which could represent the association of a process result with a certain tool or machine.

5.1 Wire harness trace object and reference schema

In this section, all object types are defined, and their attributes are specified. For all elements within the TRM, ISA-95 terminology was applied for naming the attributes if the norm contained a corresponding element. The most challenging objects for the wire harness industry are of the object type *product configuration* ($ty_{o1} = c$). For the wire harness industry, *c-objects* can be differentiated as *materials* (incoming goods and discrete parts), *as-*

semblies (parts that consume other c-objects in the P1 or P2 area), *production modules (*large assembly modules that are built in the P3 area*), function modules* (product modularization from an ordering perspective), and the *final product* (the wire harness), as shown in Figure 22.

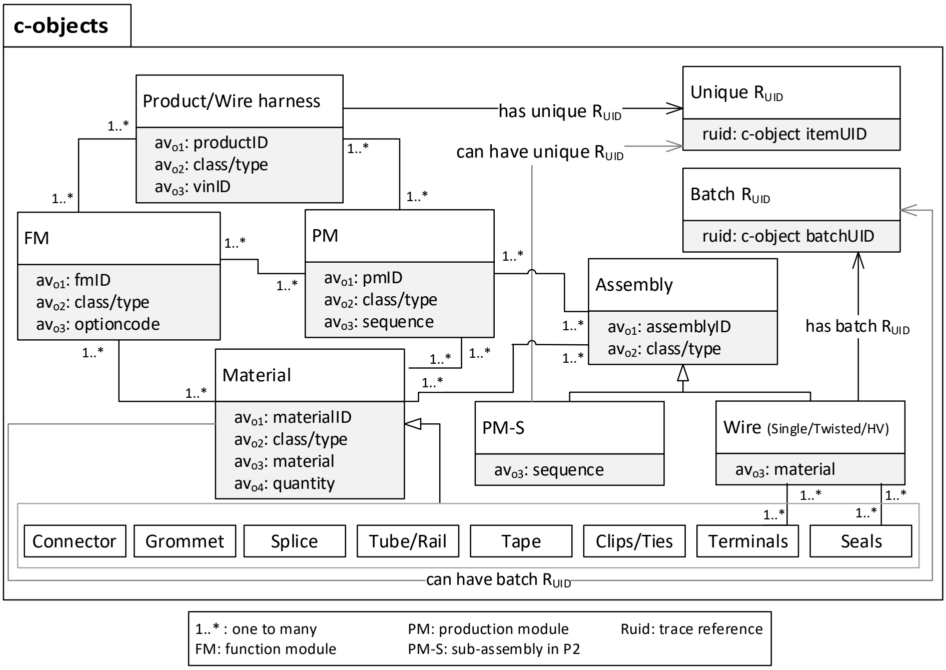


Figure 22: Trace objects schema for configuration objects

The product analysis in Table 1 can be consulted to find the basic *materials* of a wire harness product configuration, such as seals, grommets, or tape. As c-objects can be consumed by several other c-objects, they are specified with one to many relationships (1..*). In contrast to existing models, this thesis defines a crimped wire as an A*ssembly c-object*, as terminals and seals are consumed by the super-ordinate wire object. This change of specification facilitates tracing the wire's consumed materials. Moreover, sub-assemblies from the P2 area inherit from the assembly c-object type. As specified in (1), trace object attributes (av_o) need to be defined. As attributes should only contain the trace object's inherent characteristics, the required attribute set is kept to a minimum. For manufacturing use cases, it makes sense to set the first attribute (av_{01}) to the object's serial identifier. The serial ID describes an explicit c-object from an engineering perspective and does not refer to a specific object in the physical production flow. Accordingly, it can not function as the object's R_{UID}, stated in (14).

$$\forall\, O_{tr}\!: c, \qquad av_{o1} = serial\_ID(O_{tr}) \neq R_{UID} \tag{14}$$

Furthermore, the object's class or type are provided as the second attribute (av_{o2}). For the wire harness, the vehicle identification number ($av_{o3}(FM) = vinID$) is included, which unambiguously matches a wire harness with its designated vehicle. Moreover, function modules contain their option codes ($av_{o3}(FM) = optioncode$) to create a connection to the order or ERP system. Production modules are defined with a sequence constraint, which allows producing modules in the correct sequence for unpredictable KSK configurations ($av_{o3}(PM) = sequence$).

In the final product, some objects may be managed as unique parts with unique data sets ($R_{UID} = itemUID$ if lot size = 1), while others are managed as batches sharing the same production properties and characteristics ($R_{UID} = batchUID$ if lot size > 1). If batches are split, a new trace reference should be created and linked to ensure consistent data associations. Within the configuration process, specified amounts of different batches, such as batch B_i consisting of identical elements *i* and batch B_j consisting of identical elements *j*, are joined with other unique parts, such as U_k containing k, to form uniquely referenced c-objects U (15-16). Thus, subsets of both batches then compose U, while remaining elements of those batches may be built into other c-objects [P7]:

$$B_i = \{i, i, \dots, i\},\; B_i^U \subseteq B_i;\; B_j = \{j, j, \dots, j\},\; B_j^U \subseteq B_j;\; U_k = \{k\} \tag{15}$$

$$U = B_i^U \cup B_j^U \cup U_k \tag{16}$$

This is an important issue from a modeling perspective, as it means that higher-level c-objects link to the same lower-level c-object trace references. When later traversing the graph from a lower-level batch reference, different graph continuations and integration paths will be observable. For c-objects, it is reasonable to maintain batch-driven trace references for incoming material and wire assembly c-objects. For PM-S, unique references can optionally be included, while the final wire harness always has a unique trace reference. For regular PM, the case study has shown that a provision of trace references is unreasonable since they only exist as a separate module for a very limited amount of time during production. The TRM will assume no trace references for them.

As a second trace object type, *p-objects* ($ty_{o2} = p$) are specified. In alignment with the hybrid manufacturing systems in the wire harness industry, the TRM specifies *assembly processes* and *recipes*, which are associated with *process segments*. *Process segments* (p_{seg}) thus make up the main group of *p-objects* as they allow processes to be monitored operation by operation. For process segments, the first attribute is specified as the process segment identifier (av_{o1}), which can also not be used as the R_{UID} (17).

$$\forall\, O_{tr}: p, \qquad p \in p_{seg}, \qquad av_{o1} = segmentID(O_{tr}) \neq R_{UID} \tag{17}$$

Further attributes for process segments are defined as the material specification (av_{o2}), process description (av_{o3}), process duration (av_{o4}), and parameter specification (av_{o5}), as shown in Figure 23. Assembly processes encompass all process segments in P2 and P3 areas to form subassemblies and production modules, such as placing parts on the formboard, routing, taping, or welding. In assembly, each segment stands isolated, as shown by the 1-1-association. After product configuration, sequencing, and line balancing, the assembly segments form a sequence of required production tasks.

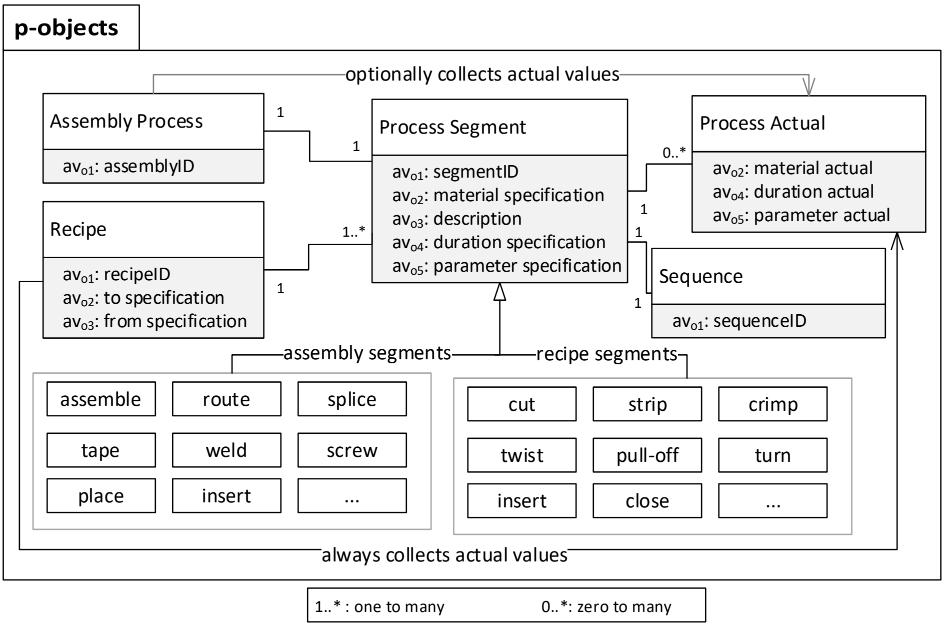


Figure 23: Trace objects schema for process objects

In wire processing, recipes group several process segments to pre-defined wire production formulas (1-1..*-association). Recipe p-objects are identified over a *recipeID* and *to* as well as *from* specifications, which specify operations and materials for both wire ends (to- and from-end). For recipes, each process segment can later be linked to actual process results, such as actual measurements or process parameters and the process sequence.

For the majority of manual assembly processes, no actual parameters are logged and verified, however, they can optionally be included analogous to recipe segments as visualized by the grey association in Figure 23. The provision of trace references poses a great challenge for p-objects. Exemplarily, a process segment p_{seg1} which describes the task of placing a connector on a formboard, is performed hundreds of times per hour at one workstation alone, while in a larger supply chain, workstations in multiple factories conduct processes with the same identifier. Trace references could be generated by, for example, combining a process segment identifier and an order identifier, which points to a process used for a specific order. As in wire processing, machines perform the same cut operation several thousand times per hour, the trace reference would need to be further enhanced with a timestamp. Combining segment identifiers, order identifiers, and timestamps, process segments could theoretically be referenced. However, an unavoidable inaccuracy remains, especially for fast process segments that require synchronized timestamps with a tolerance of milliseconds. In the TRM's data model, this thesis thus avoids process trace references to collect data and matches process results to other objects' trace references.

The third trace object type encompasses all *resource* trace objects (*r-objects,* $ty_{o3} = r$). As shown by the examples in Figure 24, r-objects are structured similarly and inherit the same attributes from the *Resource* object. The first attribute is defined as the r-objects equipment identifier. The equipment identifiers usually refer to a specific piece of equipment and can thus function as the object's trace reference, as stated in (18).

$$\forall\, O_{tr}\!: r,\quad av_{o1} = equipmentID(O_{tr}) = R_{UID} \tag{18}$$

However, the R_{UID} may also be stored separately, if the equipmentID does not refer to a specific resource but a resource type, which is often the case for workstations. Moreover, resource classes or types (av_{o2}), descriptions (av_{o3}), and their equipment element levels should be defined (av_{o4}). Equipment element levels allow classifying resource hierarchies, such as area, line, or unit, in alignment with the ISA-95 terminology. The resource objects displayed in Figure 24 contain the most common resources in wire harness production, however, further objects can be added using the modeled logic and hierarchy. In the shown schema, tools (av_{o4} = asset) are associated with the plant directly, as only very few tools are monitored in wire harness manufacturing. With increasing automation, tools, as equipped per resource, could also be integrated using the logic displayed in Figure 24 by linking them to their parent resource directly.

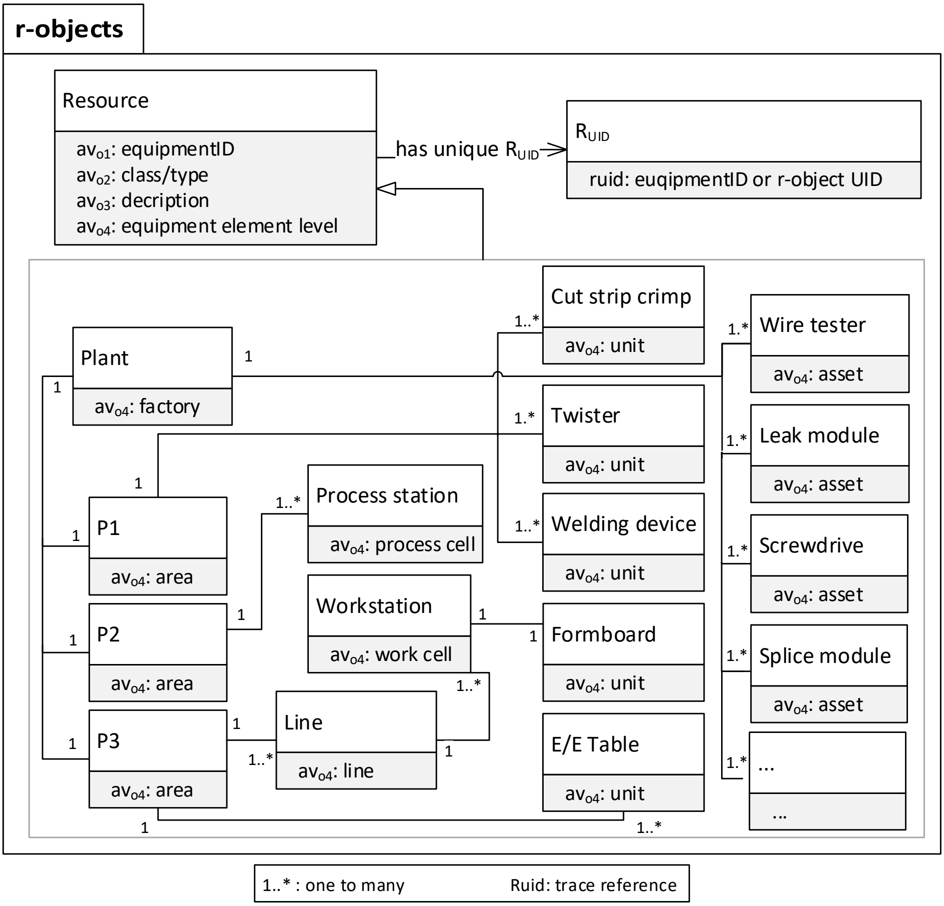


Figure 24: Trace object schema for resource objects

The fourth trace types are *order* trace objects (o-*objects*, $ty_{o4} = o$). Orders are defined using wire harness *orders*, *sub-orders* for prefabricated modules, and consumption-oriented *kanban orders* for wires. Analogous to r-objects, o-objects are identified over the order's ID (av_{o1}) which can simultaneously function as the objects' trace reference (R_{UID}), as stated in (19).

$$\forall\, O_{tr}: o, \quad av_{o1} = (sub/kanban-)orderID(O_{tr}) = R_{UID} \tag{19}$$

Further attributes are needed for the schedule (av_{o2}), and the sequence or prioritization (av_{o3}). As kanban orders encompass more than one ordered part, the amount (av_{o4}) also needs to be included, as shown in Figure 25.

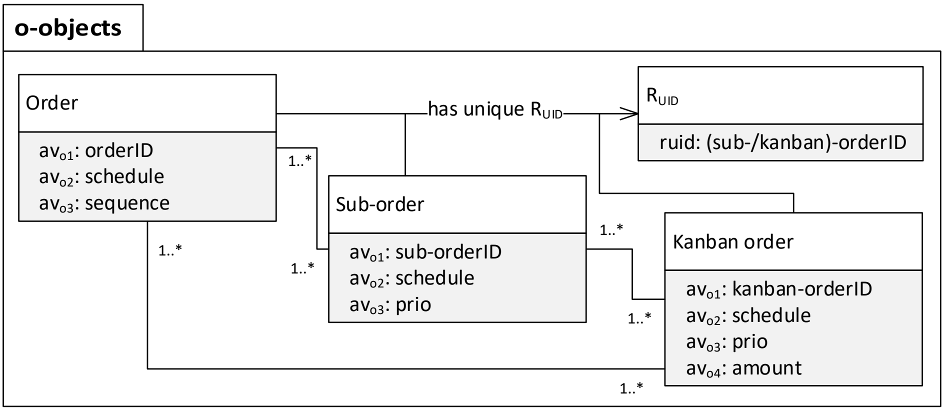


Figure 25: Trace object schema for order objects

The trace object schema thus holds the main object types, their required attributes to keep the model consistent, and their trace references to later match data records to specific objects. Any trace object instantiation can be enhanced with more attribute-value pairs if needed, which will be demonstrated during implementation.

5.2 Wire harness trace link schema

The trace link schema specifies the semantic connections between trace objects and determines how data can be associated within and across object types. Table 4 displays the trace link schema for the wire harness industry in alignment with the generally eligible trace link types. The link semantic follows the defined graph notation. The link schema follows the two general concepts of applying the semantic ":CONTAINS_*" for trace objects within a domain ($t_{yo\_x} = t_{yo\_y}$) and ":$t_{yo_x}t_{yo_y}$_MAPPING" to connect trace objects across domains.

To better model the different hierarchy levels of inter-domain object links, trace links within the same object type (cc-, rr, pp-, or oo-link) are further specified through semantic adaptions. Exemplarily, a PM containing a wire assembly would be modeled as "*PM1* -:CONTAINS_C_ASSEMBLY → *w255*". This adaption can be used to filter on specific trace links, however, it can easily be omitted using the "*" in the notation. Additionally, some general trace links are specified for objects of the same type (xx) or different types (xy). The trace links ":IS_*_PARAMETER" and ":HAS_*_PARAMETER" are defined to link further parameter specifications to a certain trace object and to match the actual parameter results to the specification.

Table 4: Trace link schema for the wire harness traceability reference model

Link	Link Semantic	Semantic adaptions	Trace link functionality
cc	:CONTAINS_C_*O_{tr}:*c*	*FM/*PM/*ASSEMBLY/ *MATERIAL	links c-objects of different hierarchy levels
co	:CO_MAPPING_*O_{tr}:*o*	*ORDER/*SUBORDER/ *KANBANORDER	links a c-object to an order trace reference
cr	:CR_MAPPING	-	links a c-object to its matching resources
cp	:CP_MAPPING	-	links c-objects to its process segments
rr	:CONTAINS_R_*O_{tr}:*r*	*LINE/*WORKSTATION/*EQUIPMENT	links r-objects of different hierarchy levels
pr	:PR_MAPPING	-	links a p-object to its matching resources
pp	:CONTAINS_P_*O_{tr}:*p*	*PROCESSSEGMENT	groups process segments
oo	:CONTAINS_O_*O_{tr}:o	*ORDER/*SUBORDER/ *KANBANORDER	links o-objects of different hierarchy levels
or	:OR_MAPPING	-	links an o-object to its matching resources
op	:OP_MAPPING	-	links an o-object to its matching processes
xx	:HAS_*_PARAMETER	-	links a trace object to non-inherent parameter specifications
xx	:IS_*_PARAMETER	-	links actual parameters to their specification
xx	:IS_*_UID	*C/*R	links a trace object with its trace reference
xy	:LOGS_*	*R_UID/*C_UID/ *P_PARAMETER…	logs trace references or parameters to an object's trace reference
xx	:HAS_STATUS	-	logs the current status to an object's trace reference

Furthermore, some trace objects are defined over a serial identifier, which is not necessarily the trace reference of the object. The connection between an object and its trace reference is achieved using the semantic ":IS_t_{yo}_UID" (e.g. "*b55* – :IS_C_UID → *w255*"). Over the trace objects' lifecycle, it will thus be connected to a multitude of trace references. To be able to bridge the difficulties of missing trace references for some mid-level trace objects, the trace link ":LOGS_*" is included. This link later allows collecting data of objects with non-existent or non-consistent trace references to another object's trace reference, which carries the data to the next hierarchy level. Lastly, the ":HAS_STATUS" link allows connecting time- and location-sensitive status data to a referenced trace object. This link serves as part of the dynamic model to monitor status changes of objects. Similar to the trace object schema, the trace links can be extended with more links when designing the data model for the chosen use case.

5.3 Wire harness tracking and tracing functions

Tracing functions are information retrieval procedures that traverse relevant trace objects and links to answer traceability-relevant questions in retrospect using the traceability data storage. Table 5 specifies the tracing functions for every trace link semantic. Moreover, it displays the tracking function for the ":HAS_STATUS" trace link, which allows observing time- and location-sensitive status updates for an object with a trace reference.

Table 5: Tracking and tracing functions with trace objects and links adapted from [P5, P6]

Link	Link Semantic	Tracing functions ($O_{tr_n}\, l_{tr_{nm}} O_{tr_m}$)
cc	:CONTAINS_C_*O_{tr}:c	What is the product configuration (children or parent c-objects) for a certain c-object?
co	:CO_MAPPING_*O_{tr}:o	Which products belong to an o-object? Which orders were applied to manufacture a c-object?
cr	:CR_MAPPING	Which resources were used to manufacture a c-object? Which other c-objects were built with that resource?
cp	:CP_MAPPING	Which process segments and parameters can be associated with a c-object? Which other c-objects can be associated with the same process segments or parameters?
rr	:CONTAINS_R_*O_{tr}:r	Which super- or sub-ordinate resources can be associated with an r-object?
pr	:PR_MAPPING	Which process segments can be associated with an r-object? Which other r-objects conduct these processes?
pp	:CONTAINS_P_*O_{tr}:p	Which process segments can be associated with a p-object? Which other p-objects include these processes?
oo	:CONTAINS_O_*O_{tr}:o	Which sub-/kanban orders belong to an o-object? Which other orders consumed these orders?
or	:OR_MAPPING	Which r-objects can be associated with an o-object's trace reference? Which other orders used these objects?
op	:OP_MAPPING	Which p-objects/parameters can be associated with an o-object? Which o-objects had these objects/parameters?
xy	:LOGS_*	Which trace references, parameters, or measurements can be associated with a trace object's trace reference?
xx	:HAS_*_PARAMETER	What are the parameter specifications of a trace object?
xx	:IS_*_PARAMETER	What are the actual parameters results for a trace object?
xx	:IS_*_UID	What is the trace reference of a trace object?

Link	Link Semantic	Tracking function $(s_0(O_{tr}), \ldots, s_n(O_{tr}))$
xx	:HAS_STATUS	What status (path) can be associated with a trace object's trace reference?

The displayed functions define the most common and most important functions for wire harness traceability. Analogous to the object and link schema, more functions can be added if needed for the aspired use case. This can be achieved by defining further trace objects and links or by combining functions to more complex traceability queries (e.g. querying from a c-object's trace reference over connected resources to process segments).

5.4 Control and sovereignty of the wire harness traceability data through trace actors

Last, the control degree over trace objects needs to be specified. The control degree, which is managed through trace actors, allows assigning, transferring, and renouncing the rights to create virtual identities of objects, to transfer objects, and to update object attributes, parameters, or status information. As visualized in Figure 26, control can be transferred from one actor (TA_m) to another (TA_n). This allows representing shifting degrees of control through physical changes, for example when an o-object leaves the P1 area and is within the control of a system of the P2 area.

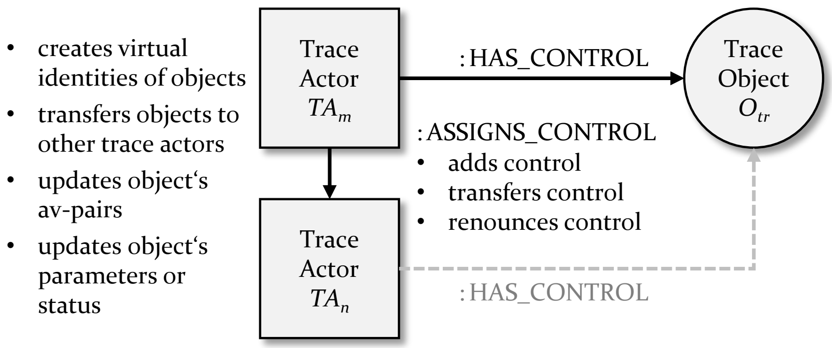


Figure 26: Managing trace object control through trace actors

Moreover, it enables distributed and granular monitoring, such as managing different control levels in one factory to enable machines or logistic systems to act autonomously. In alignment with the wire harness case study requirements for a traceability system of shared data sovereignty in a distributed production network (section 2.1.5), the first instance of trace actors is set to the company level of all actors contributing to the final product.

Each company then can transfer control to the second level; its inner traceability actors. An MES TA could for example transfer control to a machine TA. At the interfaces to other companies, object control can be signed over to a company-external TA. Exemplarily, a wire harness leaves company A and is sent to company B. As a result, control rights are transferred from a TA of company A to a TA of company B.

5.5 Wire harness reference data model

The wire harness reference data model builds on the defined objects and links and needs to enable the specified tracking and tracing functions. It is developed in alignment with the defined graph-based modeling notation and can be classified as a *type model*. Figure 27 shows the static wire harness manufacturing reference model (TRM_{MFG}). A predecessor was published in the *International Journal of Computer Integrated Manufacturing* [P6].

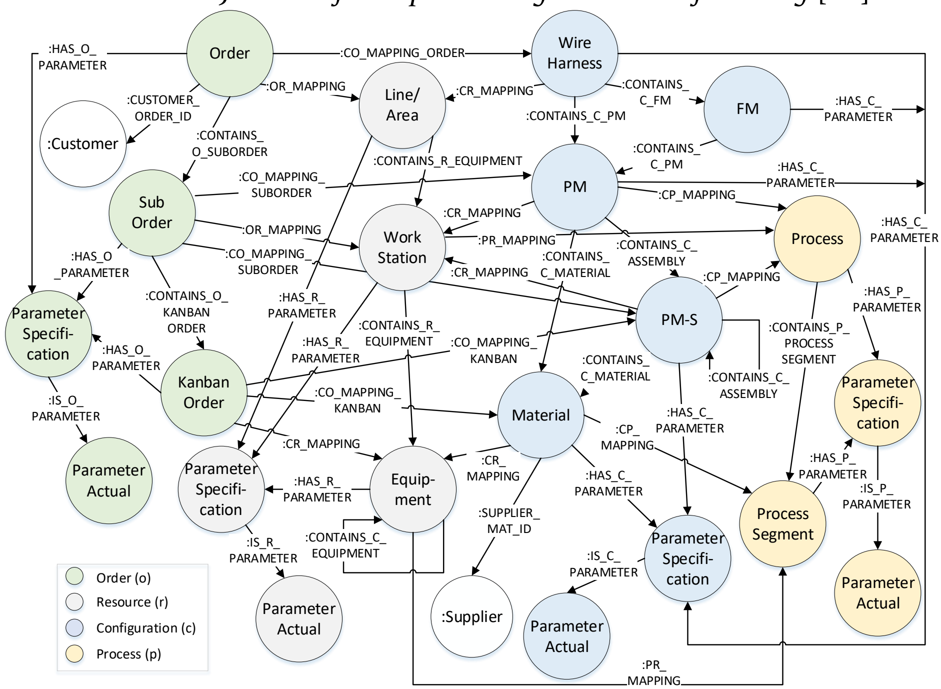


Figure 27: Static traceability reference model for wire harness manufacturing adapted from [P6]

In the traceability graph, trace objects are represented as nodes, while trace links are visualized as node edges. The modeling convention of capitalizing the first letter of nodes and using colons and all capitals for edge semantics is followed. Moreover, trace object types (o, r, c, and p) are included as node labels, visualized through the different color schemes. A graph model might appear unstructured at first glance due to the high amounts of direct relationships, especially when comparing the model to the visualization of relational models. While relational models aim for data independence, the defined trace links for the graph model aim for the contrary; the provision of granular and direct relationships within and across trace object types. The graph model is thus a more realistic reflection of the real-world relationships between the objects in a production environment. Each trace object is modeled with its main hierarchy groups, which is necessary to create sufficient traceability granularity. The data model contains the following elements:

- Product trace objects encompassing the wire harness, function modules, production modules, assemblies, and materials
- Resource trace objects including the production areas, lines, work stations, and equipment (machines, formboards, or tools)
- Process trace objects with process segments that encompass recipes and assembly processes.
- Order model containing orders, sub-orders, and kanban orders.

Each of the trace objects in Figure 27 is specified by the attribute-value pairs defined in the trace object schemas (Figure 22 - Figure 25) and controlled by a trace actor. Attributes and trace actors are not visualized in the data model for readability reasons.

In contrast to the models analyzed in the state of the art, in which only a few connections across object types exist (e.g. associating the final product with a line), the proposed model maintains the connection across all levels of the sub-graphs. The proposed static TRM_{MFG} thus provides the structure to derive the product composition for each wire harness by traversing the trace links of the blue sub-graph (":CONTAINS_C_*") starting from the functional view (FM) over the production modularization (PM) to the assemblies and materials. Each c-object further maintains direct links to processes, resources, and orders. Exemplarily, a wire assembly is built using a wire-specific recipe which is linked by a ":CP_MAPPING". The recipe contains several process segments (":CONTAINS_P_PROCESSSEGMENT") such as sample production, cutting, or crimping, each characterized by parameter specifications and actual parameters measured per segment. Moreover, the trace link ":CR_MAPPING" allows associating on which process station the wire was produced. For the resource, the used equipment, parameter specifications, and measurements can be derived using ":CONTAINS_R_EQUIPMENT", ":HAS_R_PARAMETER", and ":IS_R_PARAMETER"). While process parameters are used to describe manufacturing characteristics like cutting length or crimp height, resource parameters contain resource-specific qualities, such as stroke rate or temperature. When the TRM_{MFG} is later instantiated, one parameter specification node will be linked to thousands of actual parameter measurements including their specific results and timestamps. The traceability challenge is to match planning and specification data with instantiated objects and execution or status data. This requires a mapping between objects as planned, objects as built, and their parameters or status as measured. To derive the c-object composition as-built and to assign status updates to a specific object in the manufacturing flow, trace references are needed.

For the wire harness TRM_{MFG}, Figure 28 shows a more detailed representation of the c-object sub-graph consisting of trace objects with unique trace references, batch trace references, and without trace references. As defined in the trace link schema, trace references are associated with their corresponding c-object using ":IS_C_UID" links.

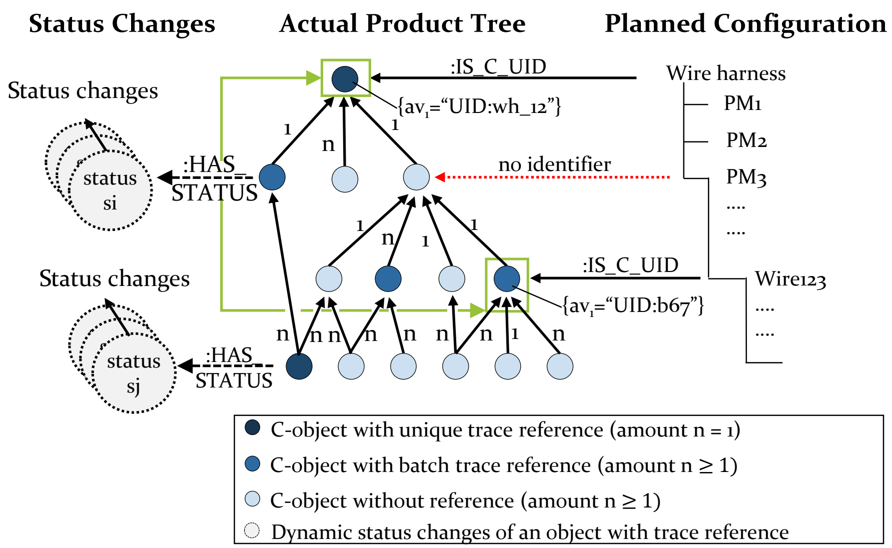


Figure 28: Challenge of matching referenced trace objects to a planned product configuration

Material c-objects are integrated to higher hierarchy levels, whereby batches can be split and shared by different sub-modules in the c-object sub-graph [P7]. When integrating a referenced c-object to another referenced c-object, the actual product tree can be derived as built. However, when referenced c-objects are integrated into non-referenced c-objects, data losses occur. In the given example, the planned configuration says that a wire harness contains a *PM3* which contains a W*ire123*. In the actual product configuration tree, it can be derived that *PM3* consumes the specific batch *b67*, however, when *b67* is integrated into the PM, a connection to a non-referenced c-object is created. When traversing the final wire harness trace reference *wh_12*, it can thus only be derived that it consisted of a PM of type *PM3* and an assembly of type *Wire123* but information on the exact assembly can not be retrieved. Accordingly, any of those sub-levels of the c-object sub-graph that are not marked with identification tags could lead to losses of data associated with referenced trace objects in lower levels of the configuration graph.

The goal of the section is to derive a data model that bridges the non-referenced c-objects as visualized by the green line. As a solution, this thesis proposes a new approach to traceability modeling by creating a connected support structure of o-objects. O-objects, which only take a minor role in process- and product-driven data models, and which are not considered at all in all PPR-driven models, provide the needed framework to overcome the described challenge. Due to the customized and thus order-driven wire harness production, the sub-graph of o-objects can maintain direct connections to its corresponding c-object, as shown in the static TRM overview (Figure 27). As o-objects are always referenced ($\forall\, O_{tr}: o,\quad av_{o1} = R_{UID}$), they allow bridging data black boxes of sub-levels caused by non-referenced c-objects using ":CO_MAPPING_*" links. The manufacturing software case (case study in section 2.1.5) showed that in wire harness manufacturing, order associations can be more easily maintained than product associations due to the customized manufacturing steering structure. As pre-produced kanban and sub-orders are linked to specific storage locations and first in first out (FIFO) material supply stations, the manufacturing software can identify the material's corresponding order without the material being physically marked. Moreover, as many materials are directly commissioned in the customized flow, the order identifiers can be directly linked by the MES. While the order structure allows more freedom within the data model and saves investments for physical tags, it places high demands on compliance with the logistical flow (e.g. correct quantity booking, compliance with FIFO, prevention of material mix-ups).

Figure 29 shows a more detailed representation of how the o-object-driven TRM overcomes the challenge outlined in Figure 28. Each wire harness has a unique trace reference (*wh_12*), which describes a specific product configuration as ordered (*order o-134*). As the TRM_{MFG} is built for a KSK production, the wire harness c-object is always referenced. In the already outlined example, *wh_12* contains *PM3*, which in turn contains *Wire123*.

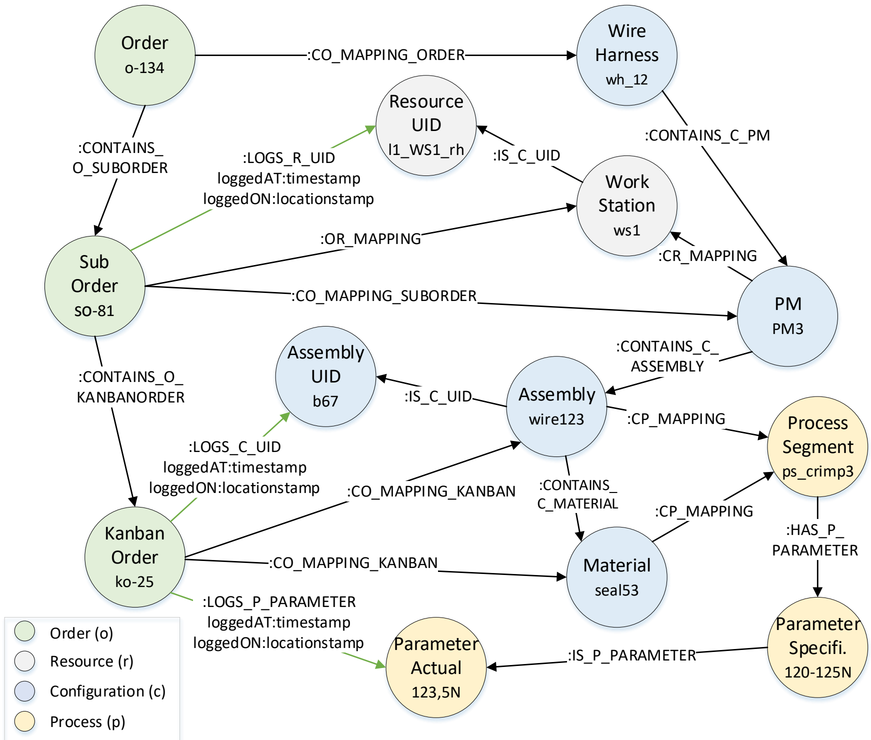


Figure 29: Overcoming data black boxes in the TRM_{MFG} through order trace objects

To log the actual batch *b67*, a connection to its kanban order *ko-25* is generated using the ":LOGS_C_UID" link (green edge). Using the order graph and its links ":CONTAINS_O_*", the trace reference *b67* can be associated with the final wire harness reference *wh_12*, effectively overcoming the continuity gap of *PM3*. Using the same logic, other actual data entries, such as resource trace references or parameters can be associated to the order graph. In the given example, *PM3* is produced on *ws1*, which characterizes any first workstation in any of the project's assembly lines. The connection ":CR_MAPPING" thus shows the general compatibility of the module and workstation and represents the connection as planned. However, the exact workstation *l1_ws1_rh*, which references a specific workstation of type *ws1*, needs to be logged in the actual production flow. As the workstation's trace reference cannot be associated with its non-referenced c-object, a time- and location-stamped trace link ":LOGS_R_UID" is created which logs the workstation's reference to the sub-order trace reference *so-81* (green edge). Analogously, the actual parameters measured in its production process (e.g. *123,5 N*) rely on ":LOGS_P_PARAMETER" trace links to correctly associate the measurements. All logs are attributed with time and location information to facilitate temporal and location-sensitive tracking and tracing functions. Since the o-object sub-graph consists of entirely referenced trace objects, it enables a barrier-free path to traverse to the actual parameters and R_{UID}.

A tracing function starting from the *wh_12* reference could thus link to its sub-ordinate o-objects, which then further open up the graph to actual or as-measured r-, c-, and p-objects. The o-based TRM_{MFG} is especially effective for the wire harness use case for two reasons: First, the o-based structure directly reflects the customized manufacturing system, which requires consistent linkages between all hierarchy levels in the c-object graph. These linkages can thus be further leveraged for traceability and control purposes. Second, the hybrid manufacturing logic is represented by kanban orders on the one hand and sub-orders or orders on the other hand. While kanban orders operate independent of the customized system, the TRM_{MFG} requires mapping kanban orders with their (sub-)orders using ":CONTAIN_O_KANBANORDER" links. This facilitates connecting the data collected per system and is independent of production software that is optimized for each area. Process-driven software of the P1 area and product-driven of the P3 area can thus contribute to a shared traceability storage. This means that the TRM_{MFG} does not require committing to one type of MES for the entire production process for a data-continuous model, but allows mapping both models through order-assigned trace links.

As described in the previous section, the trace object's status changes need to be monitored, as shown in Figure 30. A status can be described as the condition of a referenced trace object at a certain point in time. The graph that encompasses all status changes represents the dynamic TRM_{MFG}. Status changes can only be observed for explicitly referenced trace objects and will thus be modeled for r, c, and o objects for the wire harness TRM_{MFG}. The collection of actual parameters and references using ":LOGS_*" trace links is part of the dynamic and static TRM_{MFG}, as they are needed for tracing and tracking functions. The dynamic and static graph thus create an overlap ($G_S \cap G_D$) at the nodes referring to trace references. The status s_1 corresponds to an object's first status, which then changes to the next status s_2, and so on. Observing these status updates in-real time enables trace object tracking.

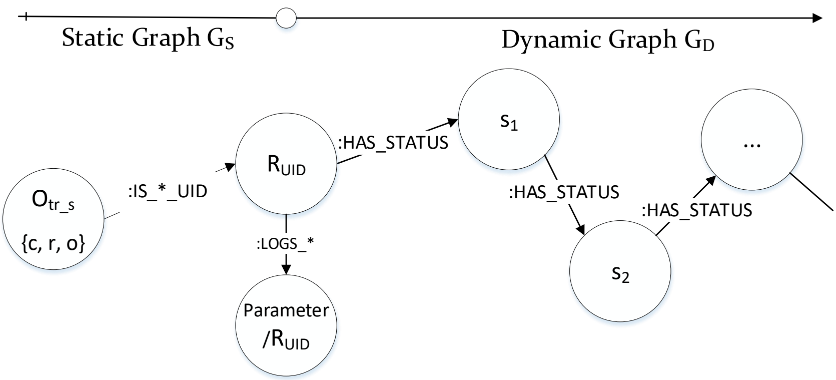


Figure 30: Dynamic state changes in the TRM_{MFG} adapted from [P6]

Finally, data sovereignty and control through trace actors (TA) needs to be defined within the TRM. Each company providing c-objects to the final wire harness structure manages a set of TA within their factories, such as machines, testing stations, production areas, or production lines. The factory level TA will be represented by a company's MES which then further assigns control to subjacent TA on the shop floor. Those TA then have the rights to create new c-objects, to merge c-objects to higher hierarchies, and to transfer c-objects to the next responsible TA, which can also be a TA external to the own company. Moreover, status updates need to be published by the responsible TA. Figure 31 shows the logic of how different TAs in a wire harness production network hold responsibility for the shared product. In the given example, three factories ($TA_{Factory\_A1}$, $TA_{Factory\_A2}$, $TA_{Factory\_B1}$) contribute to the final wire harness. Their internal manufacturing graphs (e.g. G_{A1}, G_{A2}, G_{B1}) correspond to their manufacturing TRM (TRM_{MFG}). They are instantiated per factory for all object and link types leading to highly granular and detailed production traceability models. Combining sub-graphs from each factory's TRM_{MFG}, a shared supply chain TRM can be derived (TRM_{SC}). Aligning with the state of the art on supply chain traceability, the TRM_{SC} only contains c-objects and their statuses. The TRM_{SC} thus combines all sub-graphs of c-objects ($O_{tr\_SC}: c$) including both, the static configuration structures and the dynamic status updates $s_i(O_{tr\_SC}: c)$, to create the shared product view. The TA per company holds data sovereignty and control for the c-objects and states as long as they are also within their factory sub-graph. Once the objects are transferred to the next internal or external TA, the control of the corresponding c-objects are also transferred. The resulting TRM_{SC} hence combines the c-objects of each TRM_{MFG} to an inter-company c-object view. The TA ensure that data sovereignty and control are assigned to the responsible actor. Each TRM_{MFG}

can thus provide vertical traceability (traceability within a model) and the TRM_{SC} ensures horizontal traceability (traceability across models) as per traceability definition.

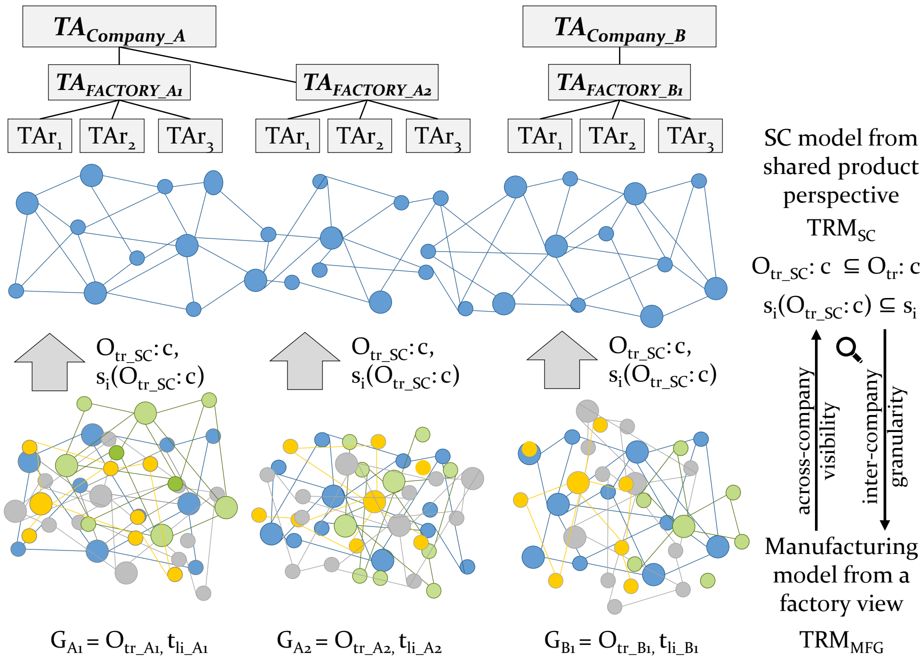


Figure 31: Extraction of sub-graphs to provide cross-factory traceability

If products were produced by one company, the TRM_{MFG} would suffice to model all traceability objects and links. As wire harnesses are manufactured in global supply chains, documentation requirements or recall campaigns are also managed by the entire manufacturing network. The supply chain model TRM_{SC} thus provides a macro view on traceability containing the final product structure $O_{tr\_SC}: c$ as static trace objects and the product's changing states $s_i(O_{tr\_SC}: c)$, as dynamic objects. Moreover, the actor associations allow resolving responsibility issues and limiting functions to the actors with product control rights. In case of a recall, the TRM_{SC} serves as an entry point to identify affected c-objects within the shared product model (blue graph visualization). The responsible companies subsequently analyze their more detailed TRM_{MFG} across all object and link types (G_{A1}, G_{A2}, G_{B1}) to identify root causes or other affected parts. This differentiation allows each company to hold its detailed data and thus intellectual property internally while still providing the data of interest to the rest of the supply chain.

6 Development and implementation of the traceability model

The TRM_{MFG} and TRM_{SC} serve as a framework to develop and implement suitable traceability models (TM_{MFG} and TM_{SC}). The traceability models thus classify as instance models that instantiate the TRMs for a specific wire harness use case. A TRM can be implemented using different database technologies, however, the data complexity and resulting choice of technology determine how well the TRM can be represented as well as the ease and performance of how tracking and tracing functions can be realized.

6.1 Complexity assessment and technical requirements

Learning from the shortcomings of the state of the art, this thesis aims at a seamless translation of type and instance models by applying database technologies that easily align to the TRM and its aspired functionalities. The use case´s data complexity determines the needed data volume, interconnectivity, storage duration, performance, and scalability, which need to be translated into suitable technical requirements. When instantiating the TM_{MFG}, a data technology that puts equal emphasis on trace objects and links is required to enable the defined degree of interconnectivity across all defined object and link types. For the TM_{SC}, on the other hand, technologies enabling seamless virtualization of c-objects and TA-allocations across manufacturing units are needed. While for the TM_{MFG} a high level of connectivity and granularity needs to be ensured, the TM_{SC} focuses on one trace object type and needs to be foremost suitable to operate in a decentralized environment. Both TM implementations need to enable dynamic features, such as status updates. Moreover, the chosen data technologies need to consider the industry-specific requirements derived in the case study section 2.1.5. The following technical requirements (TR) are stated:

TM_{MFG} technical requirements:

- TR1: Ability to integrate all elements of the TRM_{MFG} as inherent entities with consistent representation across all object and link types.
- TR2: Ability to provide short- to mid-term storage with long-term archive features.
- TR3: Ability to support a high amount, a high variety, and a high granularity of entry types and data formats.

- TR4: Ability to maintain the defined semantics for all elements of the TRM as inherent features of the database.
- TR5: Ability to implement all defined tracking and tracing functions as direct database queries.

TM_{SC} technical requirements:

- TR6: Ability to represent the c-object structures and c-object statuses across factory and company borders.
- TR7: Ability to assign and monitor shifting TAs and thus c-object control to ensure clear error allocations.
- TR8: Ability to support long-term to permanent storage for liability cases and recall claims (≥ 10 years).
- TR9: Ability to provide trust in a decentralized environment by ensuring cross-company transparency and data immutability.

General technical requirements:

- TR10: Ability to operate in near real-time. For storing the data, a latency below ten seconds is defined. Data retrieval from both TM should be near-instant to the user (≤ 1000 ms according to [118]).
- TR11: Ability to provide an unlimited amount of instantiation options to align to the high variant space (≥ 1000 c-objects per KSK). KSK wire harnesses are represented without data duplicates.
- TR12: Ability to provide links to data in other systems (e.g. design drawing, documents, etc.). All links connect to trace references.
- TR13: Ability to scale to higher data volumes, which can be queried efficiently and fast (volume of ≥ 10^8 c-objects per factory per year).

Based on the technical requirements, the defined TRM, and the shortcomings of conventional traceability storage systems, this thesis provides a solution that builds on g*raph databases* for the TM_{MFG} and *blockchain technology* for representing the TM_{SC}. To address the third research question, the following sections will demonstrate how the graph databases and blockchain effectively realize traceability for complex manufacturing.

6.1.1 Technical alignment of the graph database

When building the manufacturing traceability model, the data must be represented with high granularity and many direct connections to give the most accurate picture of the product and process history. Graph databases are the optimal fit for a highly connected TM_{MFG}, as they seamlessly translate from the TRM, integrating all defined objects, links, and functions

(TR1, TR3, TR5). Within the graph database, all trace objects can be directly implemented as graph nodes with trace links functioning as semantic edges between the nodes. The graph database's strong capability to maintain data semantics (TR4) and relations (TR1) thus makes it superior to RDBMS or document-based storage systems for the wire harness TM. Moreover, trace object types can be implemented using labels within the graph, which facilitates dealing with a high data variety and heterogeneity. The possibility to flexibly integrate schemas within the graph allows a direct translation of the defined traceability object and link schema (5.1 and 5.2) to the traceability database. Tracing functions (TR5) can be implemented as graph algorithms, which traverse relevant nodes and links and which can be limited with suitable constraints (e.g. by limiting the query to nodes with a specific label or property). In this context, the graph database is optimized for retrieving complex and heterogenous data connections and thus provides more capable algorithms for retrieving relations [99, 104]. Table 6 summarizes how the characteristics of graph databases match the requirements.

Table 6: Alignment of the technical requirements and graph-inherent features

Traceability requirements	**TR 1**	**TR 2**	**TR 3**	**TR 4**	**TR 5**
Graph database's inherent features	Inherent objects and links	Short/ mid-term storage	Data complexity	Semantic capability	Tracking/ tracing functions
Maintain objects as property nodes	x	x			
Maintain relations as property edges	x	x		x	
Data classification through labels	x		x		x
Flexible schema integrations			x	x	
Inclusion of semantics				x	x
Inherent model interconnectivity	x		x		x
Integration of different data formats and types			x		
High performance for complex transactions			x		x
High performance in retrieving data relations					x
Scalable storage with archiving functions		x			

6.1.2 Technical alignment of the blockchain

As shown in the previous sections, the TM_{SC} must offer a macro-level of transparency and documentation while providing safe transactions and immutable storage in decentralized environments. DLTs are among the most promising and future-oriented storage technologies for decentralized traceability applications [111, 119–121, P8]. Compared to conventional storage technologies, blockchain evokes a wide range of advantages for the TM_{SC}, which are realized by the interaction of blockchain features, such as digital identities, consensus algorithms, transaction verification, or shared storage. In previous publications, a detailed assessment of how those features contribute to traceability was given [P1, P7–P9]. First, blockchain provides a profound database for long-term storage while maintaining trust and immutability through consensus mechanisms and data hashing (TR8, TR9). The ability to include c-objects and their states directly in the system (TR6) can be achieved using blockchain tokens that represent each c-object as a digital entity. These digital entities can be transferred and traded along the value chain by blockchain-embodied trace actors. Specifically, trace actor association and control can be implemented through blockchain accounts (TR7). As the products move along the supply chain, control can be transferred to the next blockchain account in the value flow, which provides the aspired macro-view on traceability. Table 7 shows an adapted analysis conducted in the context of this research, which was in part published in the *Journal Computers & Industrial Engineering* [P8].

Table 7: Alignment of the technical requirements and blockchain features adapted from [P8]

Traceability requirements	**TR 6**	**TR 7**	**TR 8**	**TR 9**
Blockchain inherent features	Integration of business logic	Ownership visibility & liability	Longterm & immutable storage	Trustful collaboration
Maintain order of events		x		
Chaining of blocks		x	x	
Transfer & proof of ownership		x		
Timestamped transactions	x	x		
Digital signature		x		
Data hashing/merkle root		x	x	
Consensus on transactions		x	x	x
One-data truth			x	x

Distributed storage/no intermediary		x	x	x
Public keys for addresses		x		
Network propagation		x		
Attack resilience/no downtimes			x	
Transparent documentation	x	x	x	x
Integrity in decentral environments			x	x
Digital assets as tokens	x	x		
Integration of smart contracts	x			
Querying of events	x		x	
Cryptocurrency integration	x			

In contrast to conventional data storage technologies, blockchain further provides a set of inherent features, which make it a promising candidate for the macro-traceability use case. Product ownership visibility and tracking are, for example, ensured by the technology's ability to maintain a secure and transparent ledger of transactions through asymmetric cryptography and distributed storage. Data immutability is realized through data hashing, which creates fingerprints of data, while permanent storage is ensured through the non-erasability of data. Moreover, the consensus mechanisms allow peers to agree on one data truth. Furthermore, smart contracts integrate protocol-based conditions which transparently automate traceability processes in peer-to-peer interactions. The resilience and integrity of the network are ensured through distributed storage with no single point of failure, while the avoidance of a central authority further prevents power exploitation in the case of a recall and cuts the costs of intermediaries. For the TM_{SC}, the traceability database not only functions as the source to identify the affected trace objects but also as a risk management tool, which holds the sensitive data to solve recall issues, liability claims and to assign costs to supply chain partners [P7].

As shown above, the graph database thus provides the necessary traceability features for the TM_{MFG}, such as maintaining direct data connections as inherent features and its high capability to query complex data relations. Blockchain, in contrast, holds suitable features for the TM_{SC}, such as providing a transparent record in a decentralized environment and its ability to virtualize products as tradeable tokens across blockchain accounts.

While the features align to the addressed traceability use case, the technical capabilities need to be investigated in the further course of this thesis. The capability to operate in near real-time and scale with unlimited instantiations thus will be evaluated for both systems (TR10, TR11, TR13). Real-time refers to the ability to process the data and provide results within a predetermined time [122]. Real-time is thus not related to a specific value but needs to be defined with regard to the application [122]. Accordingly, the manufacturing traceability technology will require a higher performance than the supply chain technology. Additionally, links to off-system sources and files need to be integrated (TR12), for example by integrating uniform resource identifiers (URI) within both technologies.

While the reference model (type model) was completely defined as a graph, the implementations (instance models) are thus realized through different technologies for the TM_{MFG} and the TM_{SC}. As the TM_{SC} only holds the product structure and its statuses, the technical advantages of blockchain in a decentralized environment outweigh the advantages of maintaining the semantic and connected data through a graph-based implementation. The resulting architecture is shown in Figure 32. Following the steps for TM development, a set of technical design decisions are made for both data technologies to provide efficient and performant solutions. Neo4j was chosen for the TM_{MFG} as it is considered the world's leading graph database [123] with high technical maturity and superior performance results [124]. For the $TM_{SC,}$ a blockchain solution based on the Ethereum framework [125] was developed. The technical design considerations for the final Ethereum-based solution were taken iteratively based on prototype developments, which will be explained in detail in section 6.4. Neo4j and Ethereum are connected over a direct interface and maintain links to off-system documents, represented by the different document symbols in Figure 32.

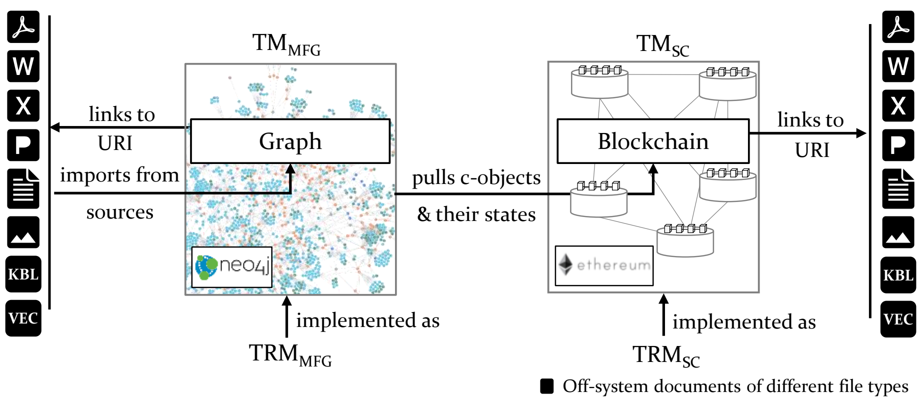


Figure 32: Technical implementation architecture for the traceability model

6.2 Extraction and simulation of the data

The first step in building the TM is the extraction and simulation of traceability data. In this thesis, an anonymized engineering data set for a middle-class vehicle produced in KSK logic is used. To build the TM, engineering data are collected and pre-processed to a uniform format. The data set includes design drawings, BOMs, BOPs, order logic including option codes, complexity lists, and detailed material libraries. All design and engineering data were available as CSV, Kabelbaumliste (KBL), vehicle electric container (VEC), XML, or PDF files. Moreover, the detailed process analysis conducted during the case study provided relevant resource data, such as machine types, tools, production areas, and line plans, as well as example files for process data such as machine, measurements, and testing logs. Resource data were mostly given as CSV or PDF files while process data were available as TXT-files. Additionally, the MES of the case study included the logic for order files. Using the complexity list and design data, 15 wire harness orders were simulated. During the simulation, the combination criteria for modules were taken into account, and a realistic composition of the wire harnesses was ensured. The resulting wire harness raw data sink built the base for all subsequent data extraction steps, as shown in Figure 33.

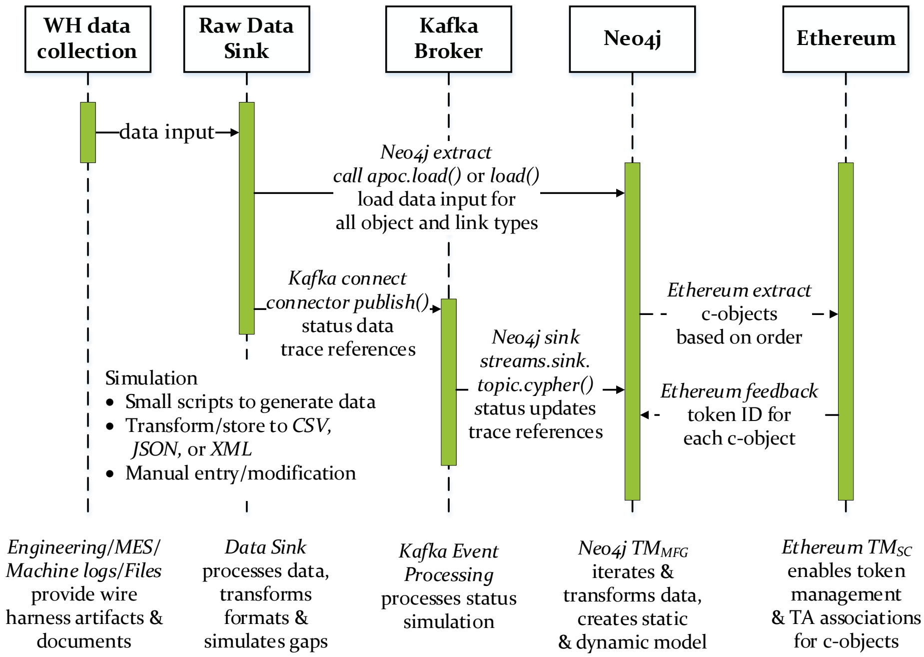


Figure 33: Data extraction for the traceability model in Neo4j and Ethereum

In the raw data sink, data was stored as CSV, JSON, or XML files to allow a direct read-in and transformation to Neo4j. For the simulation, small scripts were written that create or modify entries as needed. Moreover, measurement results were created manually as a CSV file for the 15 orders of interest. The extraction to Neo4j is realized for the static model directly from the sink (6.2.1) and for the dynamic model over a Kafka broker (6.2.2). Data for Ethereum is derived from Neo4j, which will be described in 6.2.3.

6.2.1 Extracting data for the static graph

The available data encompass static elements according to the TRM, such as trace objects and their characteristics, parameters, and indirect relationships of objects, which can serve to extract trace links. In the data sink (Figure 33), all collected files and objects were stored in a software-interpretable format. During this process, non-consistent entries were adapted, null-values were deleted, and different formatting styles were adjusted using small scripts, where possible, and manual adaptions if needed. Missing entries were created based on analogous data inputs, for example, missing timestamps were calculated based on other timestamps. From the pre-processed data sink, Neo4j imports the data using *call.apoc.load()* or *load()* functions. During import, parts of the TM_{MFG} can be directly created through suitable functions (e.g. *create*, *merge*), which translate the unstructured and non-semantic data entries into the conceptualized graph model.

Due to the structure of the data sink, domain-internal trace links (cc-, rr-, oo-, pp-links) can, in some cases, directly be generated based on the available sink data. Code 1 shows an exemplary command for data extraction from the sink to Neo4j. In Code 1, an SBOM-file is loaded, whereby empty columns are skipped (lines 1-4). Then, c-object trace nodes are created for production modules and their materials (lines 5-6). MERGE is used instead of CREATE to avoid data duplicates (TR11). The trace link between PM and material is generated using the TRM's defined semantics.

Code 1: Cypher command to extract data from external sources to the TM_{MFG} in Neo4j

```
1  LOAD CSV WITH HEADERS FROM
2  'file:///sbom.csv' AS row
3  FIELDTERMINATOR ';'
4  WITH row WHERE NOT row.productionmodule IS NULL
5  MERGE (p:PM {pmID:row.productionmodule} )
6  MERGE (m:Material {materialID:row.parts} )
7  MERGE (p) –[:CONTAINS_C_MATERIAL]→(m)
8  RETURN p, m
```

The data sink and import interface thus provided a significant share to generate the static TM_{MFG}, such as c-object and o-object structures based on SBOMs and MES-orders, r-objects based on machine and tool entries, or p-objects based on MES-recipes and assembly BOPs. Moreover, data gaps, which have not been identified in the data sink, were simulated directly in Neo4j, for example, missing process parameters or trace object attributes were simulated using CREATE functions.

6.2.2 Extracting data for the dynamic graph

Dynamic status updates are not available as files or database entries but need to be extracted in real-time and mapped to the model's trace references. To simulate dynamic status updates, an event-processing framework is needed. The event-processing framework listens to status changes of trace objects and distributes the information to the Neo4j database. The pipeline is realized for the trace objects with direct trace references using a Kafka Broker as a status update simulation and processing engine. Apache Kafka is a high-throughput and low-latency software platform for data messaging and streaming [126, 127]. It captures data in real-time from event sources like databases, sensors, machines, or software applications, to store the data for later retrieval, as well as to manipulate and route the event streams as required [126]. As it operates between data producers and consumers, it overtakes the function of a data broker while applying the principle of publish (receiving data) and subscribe (distributing data). Apache Kafka is chosen, as it is one of the most commonly used open-source data processing platforms with increasing importance in the field of manufacturing [127, 128]. As alternatives to Apache Kafka, Amazon Kinesis, RAITMQ, or ActiveMQ could be used to create the dynamic mode [129].

The software uses so-called topics to which status updates can be published. Each topic then functions similar to a folder or file system, storing the data and distributing it to relevant consumers through topic subscription. For each object domain, an individual topic stream is created, for which the status updates are published to the Kafka Broker. Neo4j then subscribes to the topic streams through the Neo4j Streams Plugin, extracts the relevant data, and creates the dynamic tracking model by mapping the status changes to the corresponding object's trace reference. Through the Kafka Broker, status updates can be simulated and written to the TM_{MFG} in real-time. The command specified in Code 2 shows the rationale of how the status updates are linked to their corresponding Neo4j object in their sequence of occurrence.

Code 2: Cypher command to dynamically create the status updates for the TM_{MFG}

```
9  streams.sink.topic.cypher.traceobject =
10 MATCH (t:Traceobject {rUID: event.RUID} )
11 OPTIONAL MATCH (t) - [:HAS_STATUS*0...] → (ts)
12 WHERE NOT (ts) - [:HAS_STATUS] → ()
13 CREATE (s:Stat {Stat:event.STAT, StatIndex: event.STATINDEX})
14 CREATE (ts) - [:HAS_STATUS] → (s)
```

First, the events are pulled from the topic of interest over the streams sink topic command (line 9). The MATCH functions (lines 10-11) then take over the role of identifying the corresponding trace reference in the database by filtering on the R_{UID} of the event message. The current status is created (line 13) and added to the latest one by creating a relationship ":HAS_STATUS", as defined in the TRM (line 14).

6.2.3 Extracting data for the Ethereum model

Neo4j, and also Apache Kafka, come as a user application with built-in functions and interfaces. Due to the novelty of blockchain technology, Ethereum-based applications are not off-the-shelf solutions but need to be holistically developed before data can be imported to them. This includes the specification and deployment of a TRM-aligned programming logic, which enables the virtual representation of c-objects. Moreover, functions that enable the defined TRM_{SC} need to be integrated. In this phase of TM_{SC} the general logic of data integration can be specified, as visualized by the dotted arrows in Figure 33. In an integrated traceability application, the o-object graph in Neo4j graph can serve to discover the c-object structure, which gets extracted by the blockchain. After issuing token identifiers in the supply chain network, the information then get pushed back to Neo4j to provide a bi-directional identification connection of c-object and token. The realization of that interface requires the development of the blockchain business logic and the full implementation of the Neo4j database, and will thus be described in sections 6.4 and 6.5.

6.3 Graph database manufacturing traceability model representation and implementation

The TM_{MFG} can now be fully represented and implemented in Neo4j. The following sections will demonstrate how the instance model TM_{MFG} effectively instantiates the type model TRM_{MFG}. Moreover, the alignment to the defined object and link schema including their semantics will be shown.

6.3.1 Neo4j trace object implementation

For each trace object type, a set of attributes were specified. Figure 34 shows how the TM_{MFG} instantiates r-objects ($ty_o = r$) and stores its attributes. Each r-object is represented by its own node as described in the TRM_{MFG}. In the visualized example, an excerpt of resource nodes is shown, such as formboards (FB1 and FB2) and CST machines (Alpha530, Alpha531, and CST12). All resource nodes are classified through labels, which represent the trace object type (*Resource*) and resource hierarchy (e.g. *Equipment*). The label determines the node color to facilitate interpretation by the user.

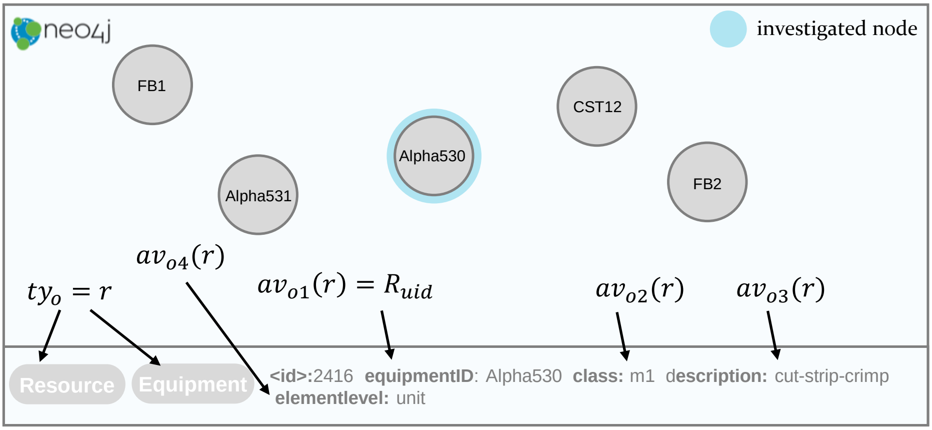


Figure 34: Implementation of resource objects in Neo4j

Moreover, attributes are implemented for each node according to the trace object schema. In Figure 34, the labels and attribute-value sets for the blue highlighted resource node *Alpha530* are shown. The first attribute corresponds to the equipment identifier (av_{o1}). The machine belongs to the machine class *m1* (av_{o2}) and can be classified as a *unit* (av_{o4}) according to the ISA-95 equipment element levels. Furthermore, a description is provided (av_{o3}). The <*id*> attribute refers to a Neo4j-internal node identifier. As this section focuses on the trace object representation, Figures 34 and the following Figures 35-37 focus on the respective trace objects while omitting other elements like trace links, references, or status.

For the product configuration, the wire harness c-objects are implemented through nodes representing *materials*, *assemblies*, *PM*, *FM*, and the final *product* as specified in the c-object schema. Figure 35 displays several c-object nodes, such as nodes virtualizing the final wire harness (*wh_121* and *wh_124*), connectors (*c78* and *c33*), and a tape node (*T77*). The highlighted example shows the product node with the identifier *wh_121* (av_{o1}).

Moreover, the vehicle identification number (*vinID*) (av_{o3}), and *class* (av_{o2}) are stored as node attributes. Analogous to the r-objects, the trace object type (*Configuration*) and hierarchy (*Product*) are given as labels.

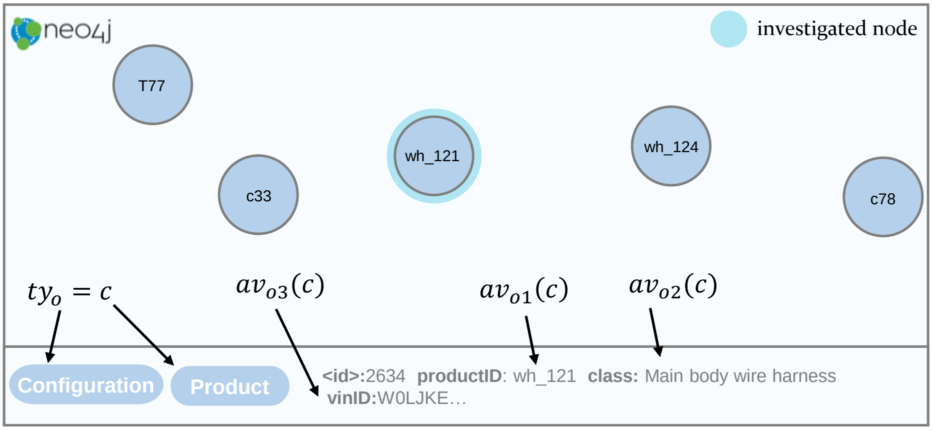


Figure 35: Implementation of configuration objects in Neo4j

As the third object type, processes ($ty_o = p$) are implemented in Neo4j, whereby Figure 36 displays an example for the recipe *r129* (av_{o1}). Analogous to the other objects, the p-object type and hierarchy level are given as node labels (*Process*, *Recipe*). For *recipes*, which define the logic for P1 wire processing, *To* (av_{o2}) and *From* (av_{o3}) specifications are stored as attributes, as defined in the schema in section 5.1. For the highlighted recipe *r129*, a seal (*71583329*) and terminal (*716451002*) are attached at one end of the wire, while the other side of the wire (From side) is produced without a seal or terminal (-).

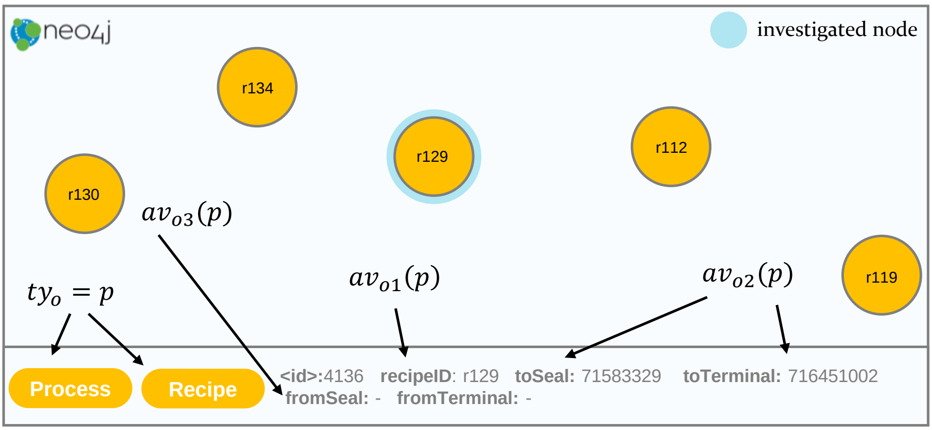


Figure 36: Implementation of process objects in Neo4j

The final object type are o-objects. In the given example in Figure 37, kanban order *o_k_11111* (av_{o1}) is shown, which is labeled with its object type (*Order*) and hierarchy level (*KanbanOrder*). As defined in the o-object reference schema, order-prioritization is given (av_{o3}) and schedules (av_{o2}) are included as inherent order attributes. Date and time data within schedule attributes are stored as temporal values in Neo4j to enable time calculations, such as duration and lead times. Moreover, the amount ordered is stored for each kanban order (av_{o4})

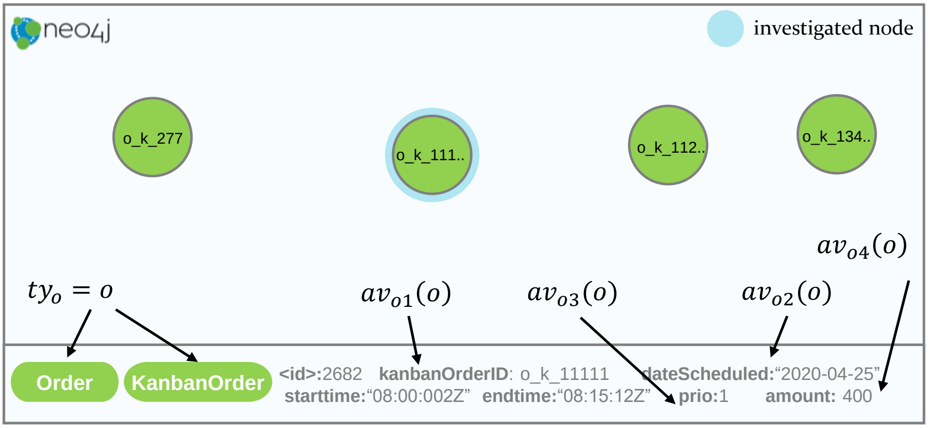


Figure 37: Implementation of order trace objects in Neo4j

All trace objects are implemented as shown in the four examples for the entire extracted data set. The object implementation can be conducted based on the extracted data through assigning attributes and transforming the data into the defined notation. The implementation of trace links is a much greater challenge since data connections were only implicitly available, if at all, and thus need to be created step by step to implement the envisioned connected traceability model.

6.3.2 Neo4j trace link implementation

The implementation of trace links can be realized by writing cypher queries that first identify a set of nodes based on given criteria, then map that set of nodes to other nodes, and finally create the trace link edge using the notation and logic defined in the reference schema. For the 15 wire harness orders, this already resulted in a high volume of 15.529 trace link instantiations, which shows how much the model relies on data connections.

In the TM_{MFG}, all defined trace links of the TRM_{MFG} are implemented. In the following, some chosen examples will be highlighted. The first example given in Figure 38 shows how configuration structures are implemented based on cc-links. As described previously, each c-object is provided with two labels, the *Configuration* label as shown in Figure 35 and a label corresponding to the hierarchy, such as *Material*, *Assembly*, *PM*, and *Product*. In the visualized query, only an extract of the full product is shown encompassing 33 configuration objects distributed over the different hierarchy levels and 32 links across those objects. To facilitate graph interpretation for the user, different colors are used according to their assigned labels (e.g. purple for *Product*, dark blue for *PM*, etc.). The colors can be modified for each query and thus have no other function than user interpretability.

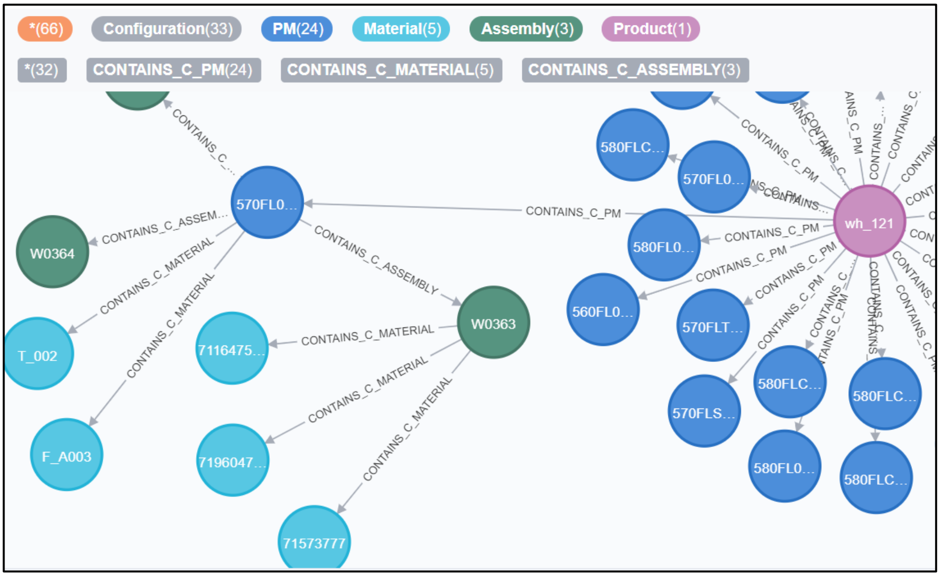


Figure 38: Implementation of cc-links in Neo4j to derive the product configuration

The visualized wire harness *wh_121* consists of several *PM* (":CONTAINS_C_PM"), which in turn relate to their integrated *assemblies* (":CONTAINS_C_ASSEMBLY"), such as *W0363* or *W0364*, as well as their integrated materials (":CONTAINS_C_MATERIAL"), such as tube *T_002* and the full tape *F_A003*. Further traversing the configuration graph, the wire assemblies are built of additional materials such as terminals and seals. As analyzed in the case study section, the product structure could not be holistically derived in conventional systems, as product configurations from the P1 area were maintained in different documents and formats than parts used during assembly in the P3 area. Through the cc-links in Neo4j, the entire product graph can be represented independently of where the sublevels are created. Moreover, as each c-object is stored unique, it can be very easily derived, which products integrate the same lower-level hierarchies by deriving all outgoing cc-links from a given c-object. The c-object sub-graph with its cc-links thus takes on the core function of representing complex product structures with all of their inner-hierarchy connections.

For resources, analogously implemented rr-links enable similar functionalities. As shown by the example in Figure 39, the resource hierarchy can be derived using ":CONTAINS_R_*" links. In the visualized example, resources are further labeled with their hierarchy levels, such as line, work station, and equipment.

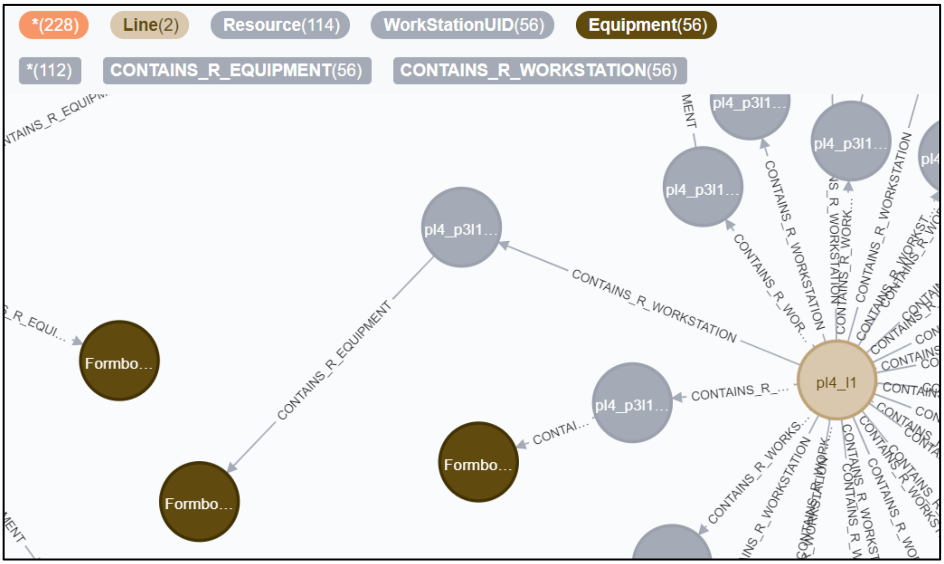


Figure 39: Implementation of rr-links in Neo4j to derive the resource hierarchy

The displayed line *pl4_l1* contains several *workstations* (grey nodes), which in their turn contain *equipment*, such as *formboards*, *machines*, *tools*, or *testing equipment* (brown nodes). The resource graph can thus be used to map other objects to the right resource element level. Exemplarily, a process segment can be directly connected to a machine or a product can be connected to a final assembly line. From those trace links, upper or lower resource connections can be further explored. As the rr-links are equally implemented for P1, P2, and P3, the resource hierarchy can be derived with a consistent logic for both, process- and product-driven production areas.

The next example in Figure 40 illustrates the storage of pp-links as well as cross-domain link types to achieve a holistic representation of more complex object connections. A given *machine* (brown node) can conduct a set of process *recipes* (yellow nodes). The practicable process *recipes* are connected over a ":PR_MAPPING". This link can, for example, be used to derive all machines, that can perform a certain process recipe.

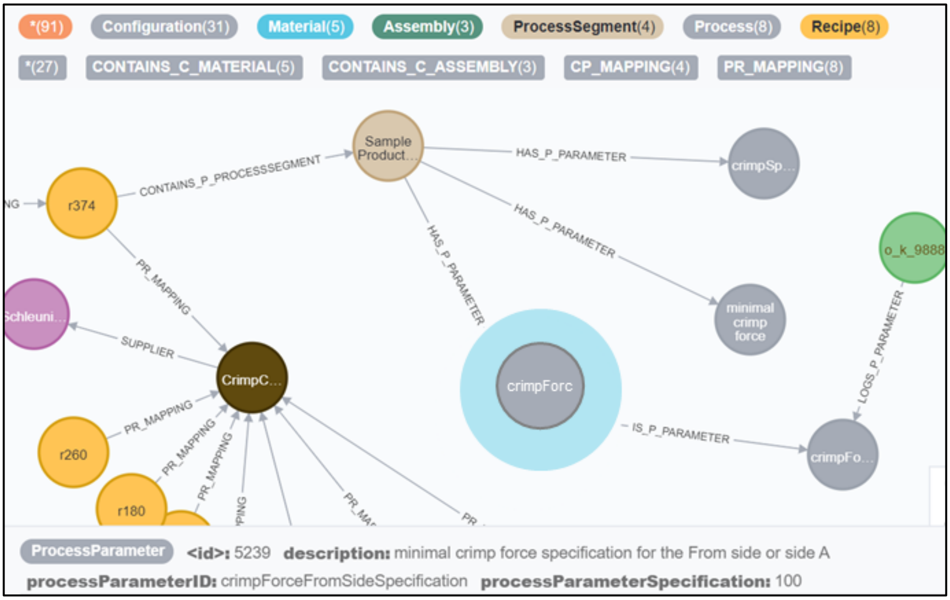


Figure 40: Implementation of cross-domain links in Neo4j for complex object associations

Accordingly, a *recipe* node can be connected to several *machine* nodes and vice versa. As defined for process-driven production flows, each *recipe* contains *process segments* (":CONTAINS_P_PROCESSSEGMENT"), which are specified through a set of process parameters (":HAS_P_PARAMETER"). The displayed process parameter node (grey outline) defines the crimp force at the *From-side* of the wire for the *sample production* process segment of recipe *r374*. The node's properties can be derived from the enlarged window at the bottom of the screenshot. The actual parameter measured is linked to the specification over an ":IS_P_PARAMETER" link. The more products are produced, the more actual parameters will be linked to the corresponding specification node. To be able to later associate a specific measurement with a specific product, the ":LOGS_P_PARAMETER" trace link is used. This link associates the actual results with a specific kanban order, which can then be used to derive upper-level o-objects and their corresponding c-objects. The given example thus illustrates the connections of different trace object domains and how specification data seamlessly integrates with actual measurements without creating duplicates. In contrast to non-graph storage, the edges thus point to the same node, making it very easy to derive which objects had the same parameters or which parts were produced at the same resource.

The role of oo-links and their function of logging data from non-referenced objects of other domains is further highlighted in Figure 41. The example shows an oo-link graph, in which the order *o_114* contains the sub-orders *o_s_1145*. When a sub-order is integrated into its superordinate order at the assembly line, the timestamp (*loggedAT*) and location stamp (*loggedON*) are stored as attributes of the highlighted ":CONTAINS_O_SO" link. This allows deriving when and where their related c-objects were merged.

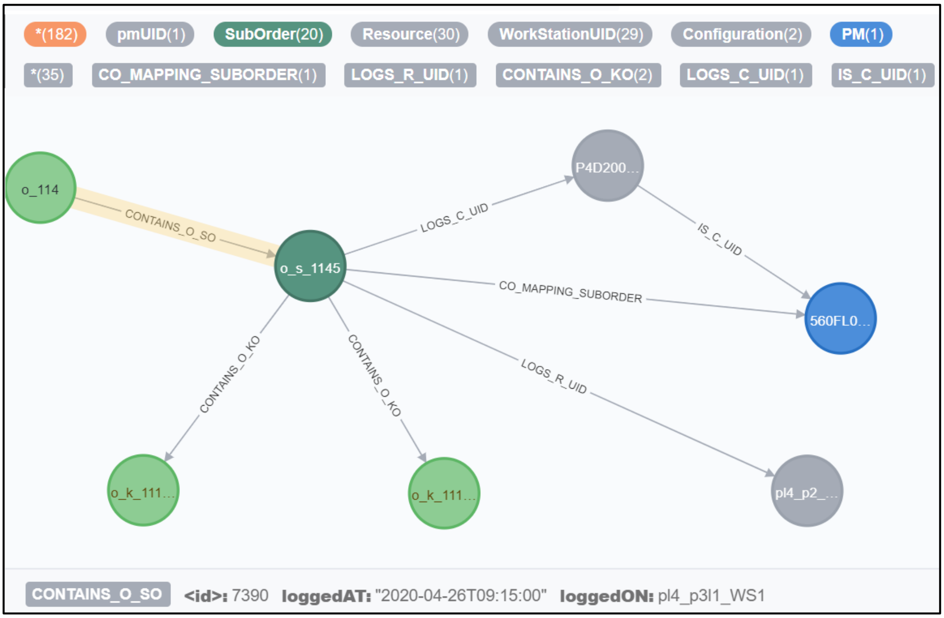


Figure 41: Order trace link implementation to overcome black boxes and to log data

The trace link ":CO_MAPPING_SUBORDER" points to the PM, which was ordered with this sub-order. The PM's trace reference is stored in a separate node and mapped using an ":IS_C_UID" link. Over a product's lifecycle, a PM will thus be linked to thousands of references, each representing a unique instantiation of that PM. To correctly assign each trace reference to c-objects of higher hierarchies, the reference is logged over its corresponding sub-order using ":LOGS_C_UID". Analogously, the resource trace reference on which that PM was produced is linked to the sub-order using ":LOGS_R_UID". As defined in the TRM_{MFG}, the order nodes, which are always referenced, collect other object's trace references and measurements.

6.3.3 Implementation of tracing functions

Tracing functions are implemented in Neo4j using cypher queries that combine a set of clauses to derive the aspired tracing result. They aim at answering the specified tracing function questions, such *as which part does a product consist of* or *on which resource was a product produced*. A typical tracing function includes the following clauses:

- A reading clause, such as MATCH or OPTIONAL MATCH, to specify a pattern (set of nodes or links) in the database.
- A reading sub-clause, such as WHERE, WHERE EXISTS, or SKIP, to add a constraint on the pattern.
- An aggregation function, such as COLLECT or COUNT, which group and aggregate patterns to a new result space.
- A projecting clause, such as RETURN…[AS], to return results.
- A projecting clause, such as ORDER BY, WITH, or UNWIND to refine the result set further.

During implementation, all specified tracing functions (Table 5, section 5.3) were implemented as local test scripts. This included testing the logic for traversing all inter-domain links (cc, oo, pp, rr) and tracing the relationships expressed through ":**_MAPPINGS" (co, cp, pr, or, op). Moreover, the defined xy- and xx queries such as ":LOGS_*", ":*_PARAMETER", and ":IS_*_UID" were developed. The following cc-link function in Code 3 allows tracing the product configuration of a given product identifier.

Code 3: Cypher command for a cc-link tracing function

```
   // What is the product configuration (children c-objects) for a certain c-object?
15 MATCH (o:Product)-[:CONTAINS_C_PM] → (s:PM)
16 WHERE o.productID = $productID
17 OPTIONAL MATCH (s) -[:CONTAINS_C_MATERIAL] → (n:Material)
18 OPTIONAL MATCH (s) -[:CONTAINS_C_ASSEMBLY] → (a:Assembly)
19 OPTIONAL MATCH (a) -[:CONTAINS_C_MATERIAL] → (m:Material)
20 RETURN o, s, n, a, m
```

The MATCH clause (line 15) combined with the WHERE constraint (line 16) derives all PM for a given product. Using a set of OPTIONAL MATCH functions, the materials and assemblies within those PM can be identified (lines 17-18). Moreover, the materials that are directly contained by the affected assemblies are derived (line 19). Exemplarily, if the productID in line 16 is set to *wh_115*, the RETURN clause yields 698 c-objects containing one product, 103 PM, 177 assemblies, and 417 materials, whose structure is represented through 1.002 cc links. The resulting graph is shown in Figure 42.

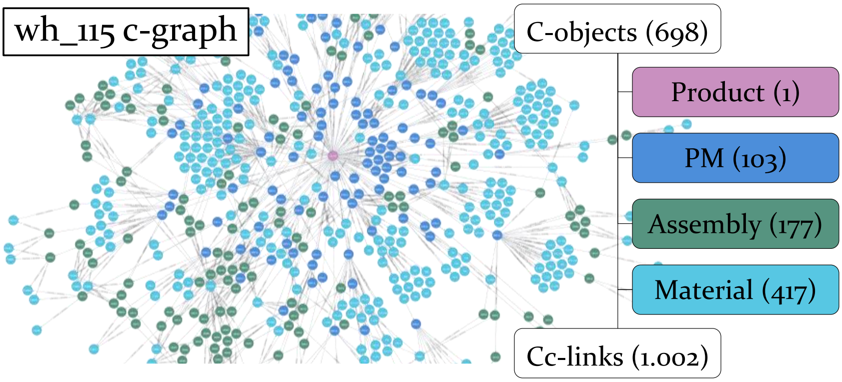


Figure 42: Neo4j result for the cc-tracing function specified in Code 3

As a second example, Code 4 shows the query for an order-resource link tracing function. The function constructs the resource history for a given order trace reference. Using a MATCH clause, all resources are derived for a given identifier (line 21). As a result, the query returns the timestamp and equipment identifier (line 22), whereby the results are sorted with an ascending timestamp (line 23).

Code 4: Cypher command for an or-link tracing function

```
   //Which r-objects can be associated with an o-object's trace reference?
21 MATCH  p = (o:Order{orderID:$orderID })-[r:LOGS_R_UID] → (t:Resource)
22 RETURN r.loggedAT, t.equipmentID
23 ORDER BY r.loggedAT
```

For a given orderID *o_111*, the tracing function identifies 28 workstations, which are displayed in their sequence of occurrence. Figure 43 shows an excerpt of the function's result. The first three workstations were logged in a takt time of 5 min. The function thus provides a table showing when (r.loggedAT) an order was passed by which resource (t.equipmentID). Accordingly, tracing functions can be outputted as a graph or table depending on the investigated use case. The results for this tracing function were visualized as a table to emphasize the chronological recording of r-objects. Tracing functions focusing on object relationships, however, could be outputted as graphs for ease of interpretation. The display and representation of tracing function results can thus be adapted to the use case independent of the underlying graph structure. Using the first timestamp and the last timestamp of the order log, the assembly cycle time and line downtimes can be calculated which can provide additional functionalities on top of the traceability model.

```
//Which r-objects can be associated with an o-object?
MATCH p=(o:Order{orderID:'o_111'})-[r:LOGS_R_UID]→
(t:Resource) RETURN r.loggedAT, t.equipmentID order by
r.loggedAT
```

	r.loggedAT	t.equipmentID
1	"2020-04-26T09:00:00"	"pl4_p3l1_WS1"
2	"2020-04-26T09:05:00"	"pl4_p3l1_WS2"
3	"2020-04-26T09:10:00"	"pl4_p3l1_WS3"

Figure 43: Result extract for an or-tracing function

Based on the tracing function schema in section 5.3, all tracing functions were tested on the Neo4j data set. For straightforward object connections, such as a product's origin, composition, or production parameters, the graph traversal queries were implemented similarly to Code 3 and 4. As the TM_{MFG} was built to align with the tracing function schema, the ease of deriving trace object associations is an inherent feature of the model. In a relational database, these queries would either require consistent c-object trace references to which all data are collected or several JOIN statements that aggregate the data according to the function. The JOIN statement design would further require a good knowledge of the underlying data model and its data keys to associate the data correctly. In a document-based system, these functions could only be enabled if parts of the typical function results were already stored within a document. Because of the heterogeneity of possible tracing functions, this would be difficult to incorporate.

While single tracing functions for straightforward object connections could be very easily implemented in the graph TM_{MFG}, complex tracing functions, which are combinations of multiple functions, also need to be considered. In a combined query, sub-graphs need to be derived and stored as an intermediate result space from which traversals across heterogeneous link types can be conducted. Those functions are needed in the case of a recall to detect all affected products or to identify the root cause of a failure. To illustrate the challenge of a combined tracing function, it will be assumed that all products containing a certain material R_{UID} need to be recalled.

Figure 44 shows the recall challenge for the example R_{UID} *TF68012004*, which refers to the material ID *7116690306*. This material ID is integrated into several assemblies, such as *W0186*, *W0160*, *W0258*, and *W0185*, visualized as dark green nodes. Starting from that material's R_{UID}, the c-structure can not be holistically derived using a single cc-link tracing function, as c-objects without references hinder mapping lower levels to the final wire harness trace reference.

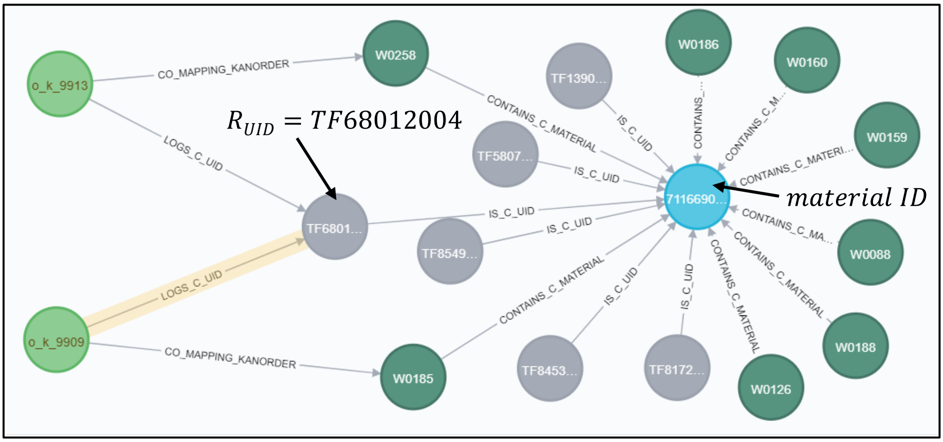


Figure 44: Recall challenge for a combined tracing function

To derive a tracing query that can securely bridge non-referenced c-object sub-levels, more advanced graph analytics algorithms based on order sub-graphs are created. Using the *Neo4j Graph Data Science Library*, Code 5 shows an implementation of a combined tracing function. The algorithm creates a projection of the Neo4j property graph model, which contains only the relevant nodes and properties (lines 24-33). The sub-graph is generated by traversing the relationship that enables the aspired node associations, whereby the orientation indicates whether a relationship is to be traversed along its edge (natural) or in the opposite direction (reverse). The projected sub-graph is stored in a compressed structure and can be called over its name *ORDERTREE* for function instantiation. Then the algorithm identifies the c-object's trace reference and indicates that node as a starting point (lines 34-35). From that starting point, the ORDERTREE is traversed using a breadth-first search algorithm (lines 36-37), and the traversal path from the reference over the order structure to the final wire harness trace reference is identified. Last, the resulting wire harness trace references are outputted and sorted in ascending order (line 38). For the given example in Figure 44, Code 5 identified eleven wire harnesses, in which the batch of interest was integrated. The identification of the product identifier over the co-links thus provides a promising application of tracing functions for customized and complex product structures. It can be called for any type of material or assembly R_{UID} and is thus applicable for all TM_{MFG} that were generated in alignment with the defined reference model.

Code 5: Cypher co-tracing function to derive c-objects from order sub-graph

```
// Mapping from an o-sub-graph to a c-structure
CALL gds.graph.create('ORDERTREE', ['AssemblyUID', 'MaterialUID',
'Order', 'SubOrder', 'KanbanOrder', 'Product'], {
CO_MAPPING_ORDER: { type: 'CO_MAPPING_ORDER',
orientation: 'NATURAL'},
CONTAINS_O_SO: { type: 'CONTAINS_O_SO', orientation: 'REVERSE'},
CONTAINS_O_KO: { type: 'CONTAINS_O_KO', orientation: 'REVERSE'},
LOGS_C_UID: { type: 'LOGS_C_UID', orientation: 'REVERSE'},
CO_MAPPING_KANORDER: { type: 'CO_MAPPING_KANORDER',
orientation: 'REVERSE'}})
YIELD graphName, nodeCount, relationshipCount;
MATCH (a:MaterialUID{materialUID:$materialUID})
WITH id(a) AS startNode
CALL gds.alpha.bfs.stream('ORDERTREE', {startNode: startNode})
YIELD path UNWIND [ n in nodes(path) | n.productUID] AS WIREHARNESS
RETURN DISTINCT WIREHARNESS ORDER BY WIREHARNESS
```

6.3.4 Implementation of dynamic elements through Neo4j and Apache Kafka

The dynamic TM_{MFG} is implemented through a simulation of real-time events and status changes as described in 6.2.2. Figure 45 shows how Neo4j and Apache Kafka interact to create the dynamic model. The given example demonstrates status updates for c-objects which are published to the topic "PRODUCT_STATUS_CHANGE". The status are sent to Apache Kafka, appearing in its event log of the broker. From Kafka, other systems can subscribe to their topics of interest to consume the data. As shown in the example, Neo4j filters and extracts the status updates for product changes in real-time. It then creates the dynamic status path by matching the first status update *s1* to its corresponding static node. The trace object's R_{UID} thus functions as the initial node of the status path. All following status are added chronologically using the ":HAS_STATUS" link (*s2*, *s3*, etc.). Neo4j is able to assign each status correctly by mapping the database's R_{UID} with the information given in the Kafka event message log (black boxes).

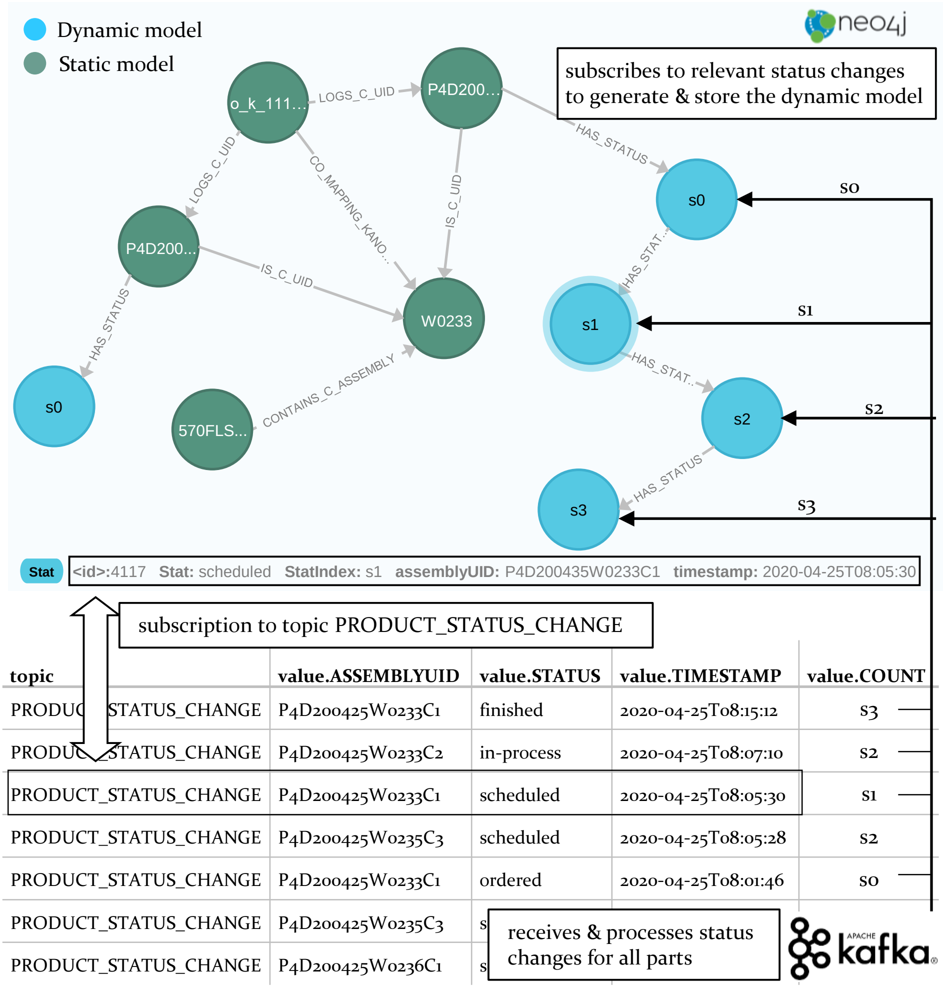


Figure 45: Implementation of dynamic status changes using Neo4j and Kafka

Each status node in the resulting dynamic model has a status index, which can be used to derive the sequence of status changes. It is further attributed with its status description, such as *scheduled*, *ordered*, or *in rework*, as well as the timestamp of the status' occurrence. In this thesis, example status changes were simulated in Apache Kafka (value.STATUS) based on the available data, however, they are not extensive and can be set without restrictions according to the use case. In relational or document-based systems, status updates are often integrated as attributes within the model's main entities. The chosen implementation in the TM_{MFG} represents status as their own nodes, which enables a scalable and flexible addition of status updates. The chosen labels allow to easily separate the dynamic from the static model, if needed, whereby the two models remain connected through the ":HAS_STATUS" link path. Moreover, this link path enables the implementation of tracking functions, which dynamically follow the status changes for trace objects. Analogous to the visualized example, status updates for c-objects, o-objects, and r-objects were implemented, leading to dynamic sub-graphs for each referenced trace object in the TM_{MFG}.

6.3.5 Implementation of tracking functions

To observe the status changes of an object, the graph model can be traversed along its ":HAS_STATUS" relationships. In this thesis, tracking functions were implemented using a spanning tree algorithm (apoc.path.spanningTree), which traverses the graph from the initial status to the final status through the ":HAS_STATUS" relationship filter as shown in Code 6, lines 39-40. The algorithm allows outputting the entire status path (line 41), whereas the root of the path corresponds to the trace object's most recent status. The tracking algorithm can be instantiated for any object with a trace reference (line 39), allowing status tracking for all identifiable objects.

Code 6: Cypher command for an assembly status tracking function

```
   //What is the o-object's trace reference?
39 MATCH (a:AssemblyUID {assemblyUID: $assemblyUID})
40 CALL apoc.path.spanningTree(p, {relationshipFilter: "HAS_STATUS"})
41 YIELD path RETURN path
```

6.4 Blockchain-based product configuration and supply chain model implementation

As shown in 6.1.2, any blockchain-based TM_{SC} implementation profits from the technology's general features, such as providing a transparent record of events in a decentralized environment. However, decisions about its concrete technical architecture will ultimately affect the solution's ability to address the stated technical requirements (section 6.1) and its ability to instantiate the aspired TRM_{SC} (section 5.5). For non-complex parts, like packages or containers, blockchain-based traceability applications are comparably mature (e.g [120, 130]). For complex assemblies, on the other hand, blockchain-based virtualization remains a great challenge [P7]. Especially the direct representation of c-objects (TR6) for complex product configurations has not been focused so far and requires novel proof of concepts.

6.4.1 Technical design based on tests and prototypes

For the technical implementation, a set of design decisions need to be taken to derive a suitable blockchain solution. As depicted in Figure 46, the design considerations encompass the network structure, consensus algorithm, and environment as well as business logic and asset virtualization.

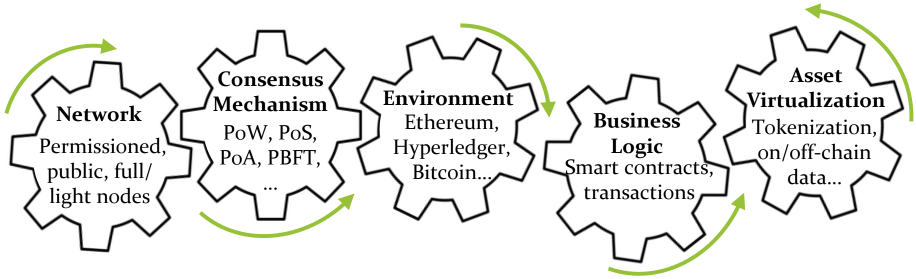


Figure 46: Technical design considerations for the blockchain solution

For the network structure, this thesis aims at a fully distributed concept with equal power distribution of all participating companies. For the application of supply chain traceability, a permissioned chain is preferable. In contrast to public blockchains, in which everyone can participate, permissioned blockchains are used for enterprise solutions with a pre-defined user group [131]. In the permissioned architecture, all companies should provide a full node to the network ($\sum nodes \geq \sum TA_{Company}$), whereby companies with several factories ideally provide one full node per factory ($\sum nodes \geq \sum TA_{Factory}$). Full nodes participate in block mining and consensus, smart contract creation, and approval, as well as storage of the entire blockchain. Furthermore, full nodes validate transactions, reject incorrect transactions, and add blocks to the ledger [P8]. Moreover, a set of light nodes trace actors (TA_r) need to be defined per factory, which could be an MES, machine, or resource area connecting to the blockchain over a light client. Light nodes can submit new transactions, verify relevant transactions, and interact with the smart contracts of the network.

The second technical design consideration concerns the consensus mechanism. The consensus mechanism describes the method or algorithm of how to agree on the truth of data and thus transaction history [109]. In open networks, such as Bitcoin, energy-intensive consensus mechanisms such as Proof of Work (PoW) are applied, which require the nodes to participate in a hash search race with a defined difficulty level [109]. An increasingly popular alternative is Proof of Stake (PoS), in which the relative stake in the network determines the likelihood of participants to add the next block as it is assumed that nodes with a great value share are less probable to compromise the network [132]. In the very similar Delegated Proof of Stake (DPoS), delegates are elected to mine the blocks [132]. Proof of Authority (PoA) is a more energy-efficient consensus mechanism [133, 134]. In PoA, the mining node is randomly selected from a set of authorized nodes [133]. The algorithm is safe for permissioned networks as long as 51 % of all authorized nodes do not conspire to manipulate. Another energy-efficient consensus algorithm for permissioned networks is the Practical Byzantine Fault Tolerance (PBFT) algorithm. In PBFT, a mining node is elected to create the next block. The block then needs to be confirmed by two-thirds of all eligible miners [132]. The participation openness and consensus protocol largely determine the performance and scalability of a blockchain. When choosing the consensus, the tradeoff between ease of participation and achievable data throughput needs to be considered. Moreover, the block size, block time, and thus transaction rate need to be applicable for the chosen use case. In the public Ethereum network, a block is mined every 12 s, whereby the transaction throughput is restricted by the gas limit. The gas limit is an upper bound for the computing capacity available per block and is used to protect the network from denial of service attacks [110]. Public networks are very scalable regarding new participants, while permissioned networks applying PoA or BPFT are more scalable regarding data throughput with limits to participation [135].

In addition to those general architectural considerations, the availability of existing frameworks needs to be considered. Existing frameworks, like Bitcoin, Ethereum, or Hyperledger Fabric, facilitate the implementation through a pre-defined and continuously improved development environment, available clients, as well as standards for application development and integration. Moreover, they determine the ease of integrating user-defined program logic such as smart contracts, which are needed to virtualize the business logic for a specific industry application or use case.

For the wire harness TM_{SC}, the choice falls on a *permissioned* blockchain with an *energy-efficient consensus* that allows *smart contract* development due to the pre-defined group of participants, high data throughput requirements, and the necessity to integrate business logic on-chain. To find the optimal architectural fit within these general boundaries, an iterative approach of prototype development and assessment was applied. After general feature considerations [P10, S4], the first blockchain traceability prototype was built in Hyperledger Composer [S5, S6]. Hyperledger Composer is an enterprise development tool on top of Hyperledger Fabric, which supports permissioned blockchains using a voting-based consensus similar to

PBFT and permits business logic integration through *chain code*. In the wire harness prototype network, a set of participants were defined, and processes like building and shipping the wire harness were simulated. Moreover, c-object states could be integrated into the transactions (e.g. *received*, *tested*, etc.). The prototype worked well to monitor changing TAs for defined c-objects [S5, S6]. TA control could be implemented and tracked, as only TA with asset control rights could transfer and manipulate the given c-objects [S6], however, the first prototype yielded difficulties managing the complex configuration structure of wire harnesses, which could not be efficiently virtualized to the network.

The next development phase thus focused on object representation. The hypothesis was that c-object representation could be improved through blockchain tokens similar to the first concepts emerging in the literature [120]. In the second prototype, blockchain tokens were hence applied to directly virtualize c-objects in an SCM context [P8]. The solution included a smart contract that allowed c-objects to be directly transferred between TA, the monitoring of control, and the tracking of status updates. Its functions inherited from the ERC standard set (Ethereum request for comment). The ERC 20 standard, exemplarily, manages assets as fungible and non-distinguishable tokens for cryptocurrency applications [136] and is only able to support applications in which all assets are interchangeable. In contrast, the ERC 721 standard manages objects as non-fungible or unique tokens [137]. The ERC 721 was thus chosen for the second prototype and proved to be highly applicable to track a final product throughout the supply chain. The development of traceability logic was facilitated through the standard's built-in functionalities of ownership mappings and token transfers. While the usage of tokens in the second prototype optimized the solution's ability to digitally manage physical assets, it was again not able to represent c-objects in their multi-hierarchical structure consisting of unique and batch-type parts. Moreover, the prototype was still based on the Ethereum main network with PoW consensus so that the actual throughput and performance could not be realistically simulated.

Due to its flexible smart contract development and the opportunity to integrate tokens, Ethereum was also chosen for the third prototype as the most applicable framework to address c-object virtualization. As some c-objects are unique, they need to be represented as a unique token with an instance of one, while other assets, such as batches, are digitally identical, which necessitates a token with multiple instances. Hence, another token standard that fulfills those requirements had to be found [S7]. The search yielded in the recently published ERC 1155 standard by Enjin [138] that had been developed for the gaming industry to manage the players' fungible and non-fungible virtual items within one contract. The idea was to transfer this capability of the ERC 1155 standard to the traceability challenge of managing unique parts and batches in a holistic c-object wire harness representation [P9, S7]. Moreover, the solution should enable the throughput for a realistic wire harness value chain. To improve from the second prototype, this was to be realized by optimizing the transaction rate and volume. Accordingly, the third prototype was developed as a permissioned blockchain with a PoA consensus based on the Ethereum framework [P7, P9, S7]. In this prototype, an Assembly Token Manager (ATM) smart contract for c-object virtualization based on the ERC 1155 was integrated. The iterative development until the final prototype is shown in Figure 47. The final prototype thus provided a superior asset representation compared to the other two prototypes while also aligning well to the business requirements. In the following sections, the resulting implementation of the TM_{SC} based on the third prototype is described as well as its integration into the holistic traceability application.

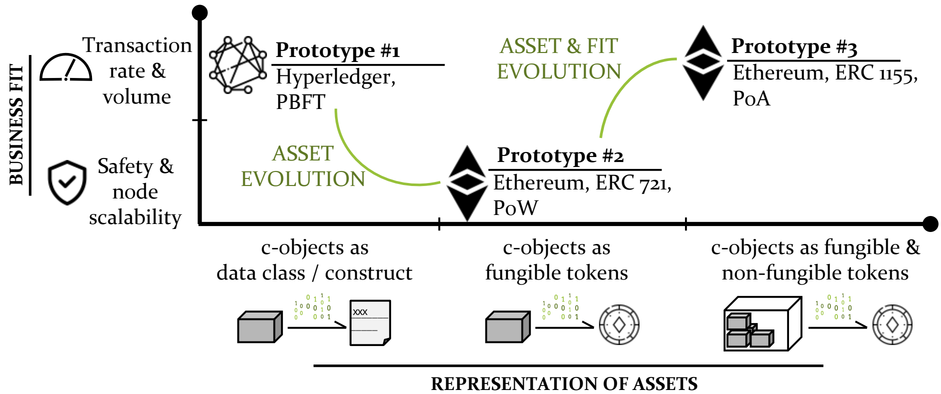


Figure 47: Technical development to the final Ethereum implementation

6.4.2 Network and consensus implementation

The permissioned Ethereum network was set up in the Go Ethereum (*Geth*) client [125]. Geth runs the Ethereum virtual machine (EVM), which manages the deployment and execution of the smart contract and is imperative for its consensus. For the TM_{SC}, the accounts for two full nodes ($TA_{CompanyA}$) and ($TA_{CompanyB}$) were implemented, and a genesis file for the initial configuration was created. Furthermore, the PoA consensus was implemented through the *Clique* protocol [139]. This adaption of PoA for the Ethereum ecosystem was chosen for TM_{SC} in alignment with the Benchmark conducted by De Angelis et al. [134], who demonstrated its performance in permissioned blockchains. Moreover, light nodes (TA_r) were created and the ATM smart contract was deployed to the EVM. The resulting network implementation for the TM_{SC} is shown in Figure 48.

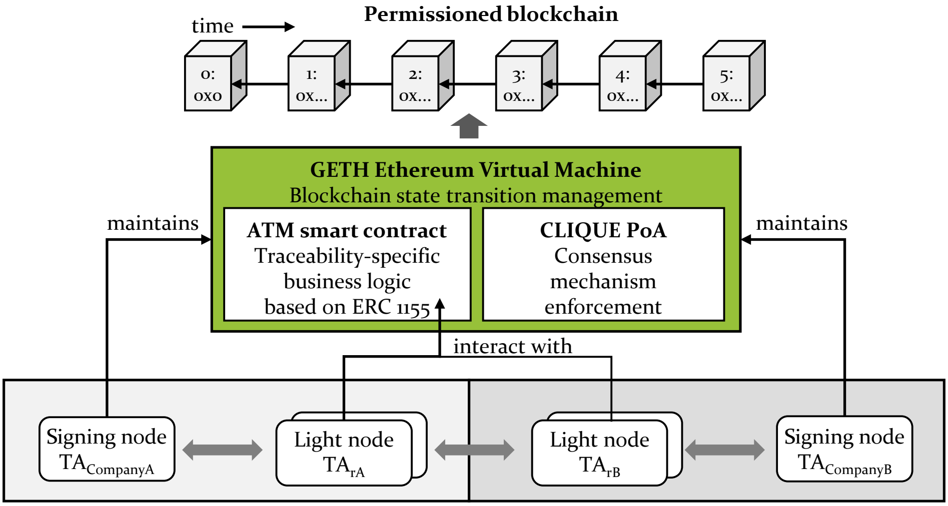


Figure 48: Ethereum network set-up in Geth with Clique PoA adapted from [P7]

6.4.3 Development of the ERC 1155 token contract

The business logic to manage and store the complex c-structure of wire harnesses was enabled through the ATM smart contract, which is written in *Solidity*, the Ethereum-native and object-oriented programming language for smart contracts development. The ATM was developed in [S7] and was then developed further for publication *in Procedia CIRP* [P9] and the *Journal of Manufacturing Systems* [P7]. The ATM inherits from the ERC 1155 reference implementation [138] and holds the logic to create, transfer, merge, and update virtual tokens representing physical c-objects. The key functionality of simultaneously maintaining unique and interchangeable tokens within the ATM is shown by a double mapping, which assigns each token ID to a map of accounts [138, P7]. This mapping can be used to represent unique and batch c-objects as well as their TA affiliation as visualized in Table 8. In the TM_{SC} batch c-objects are thus represented by a token ID with token instances greater than one, as shown for ID1 and ID2. As batches can be integrated into different c-objects and managed by different resources along with their value flow, the mapping allows the association of their instances with different accounts, as shown for ID2. On the other hand, a unique c-object is represented by a non-fungible token with an instance of one (ID3). If c-objects are assembled or scrapped, these objects are no longer physically available, but their IDs are not reused and remain traceable, as shown by the last example with ID4.

Table 8: Visualization of an example token balance mapping adapted from [P7]

Token ID	TA Accounts 0x1	0x2	0x3	0x4	Traceability scenario
ID1	5	0	0	0	A fungible token, whose 5 instances are owned by one account representing a batch c-object being processed by one resource.
ID2	1	2	1	1	A fungible token, whose 5 instances are distributed across multiple accounts representing different instances of a batch being processed by different resources.
ID3	1	0	0	0	An effectively non-fungible token whose single instance is owned by one account representing a final wire harness owned by one resource.
ID4	0	0	0	0	A burnt token, which does not have any live instances but is kept for traceability purposes representing a consumed or assembled c-object.

Each TA account can update the data of its owned tokens and initiate their transfer to other accounts. The affiliation with an account at specific points in time builds the base for tracing location and responsibility changes in distributed supply chains [P7]. To realistically virtualize different levels of TA control, the ATM integrates a mapping of token IDs to TA accounts with control rights [P7, P9]. The resulting public *controllers* enable certain TAs to act on behalf of operating TA accounts, such as an MES overtaking token control for its subordinate resources. These control rights are verified for each of the ATM's main functions (e.g. controllers[_id][msg.sender] = *true*) to ensure that data sovereignty always remains with the responsible TA. The main functions in the ATM to re-create the c-object traceability logic can be classified into the categories c-object control management, existence management, and status and metadata management [S7]:

C-object TA control management

- The *is controller* function verifies the controlling TA for a c-object.
- The *add controller* function assigns c-object control to a new TA.
- The *renounce control* function renounces control over a c-object from a TA.

- The *transfer control* function assigns control over a c-object to a TA and strips the current controlling TA of its controlling rights.

C-object existence management

- The *create* function enables c-object virtualization through token formation.
- The *craft* function enables merging c-objects by consuming a set of input tokens and emitting an output token with a given quantity.
- The *burn token balance* function removes the c-object from a given TA account.
- The *single* and the *batch transfer* function enables TA allocation changes for c-objects through token transfers between accounts.

C-object status and metadata management

- The *new status* function enables c-object updates by emitting status events for a token.
- The *new URI* function sets a new link to off-chain documents.
- The *new* R_{UID} function updates the trace reference information for a given c-object.
- The *update metadata* function allows simultaneously updating status, URI, and R_{UID}.

Figure 49 provides an example of how the developed prototype holistically implements the TM_{SC} through the defined smart contract functions. The use case displays shifting TA allocations well as status updates as products move through their value flow. Each TA is represented by a blockchain account and can interact with the ATM directly or rely on the MES, which is listed as an authorized account ($TA_{Factory\_MES}$ is the operator of all TA_r). C-objects are virtualized through c*reate* and *craft* functions along with the product transformations that occur in the physical production flow. Incoming parts can be virtualized through previous supply chain partners or registered to the blockchain in the inbound logistic area. When the c-objects move between TAs, their tokens are transferred between their respective accounts. When the product leaves the sovereignty of the factory ($TA_{Factory\_MES}$), not only the token but also the token control is transferred to the next responsible TA account (e.g. TA_{OEM}).

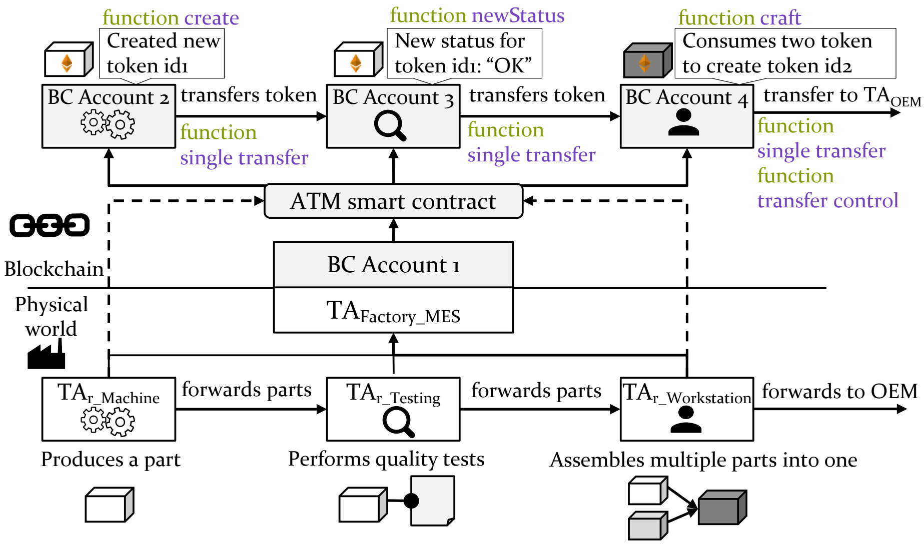


Figure 49: Use case-based demonstration of the ATM's functions adapted from [P9]

Due to the multitude of functions, only a few selected functions are presented in more detail. Figure 50 displays the code flow chart for the *transfer control* function. The function inherits from the ERC 1155 standard and can only be called by the current controller of the c-object's token ID, which is ensured over the modifier *controllerOnly*.

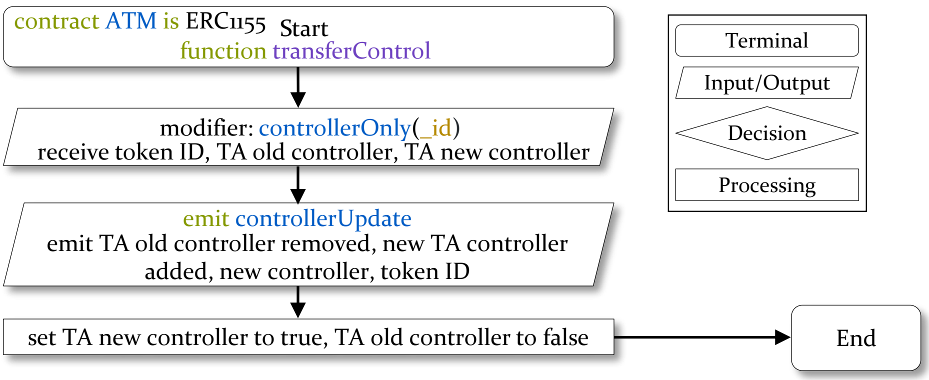


Figure 50: Code flow chart for the transfer control function

The function receives the current controller TA and the new controller TA and transfers the control. To store the control change in the ledger, the two TAs and the token ID are logged (*controllerUpdate* event). The function is called whenever c-objects are transferred between companies or factories, effectively transferring the data sovereignty and control in alignment with the physical c-object responsibility.

To virtualize c-objects as tokens and to recreate batch logic, create and craft functions are used. Figure 51 shows the create function, which receives the token quantity, a link to external traceability documents, the TA account, and the c-objects R_{UID}, which builds the link to the representation of c-objects in the Neo4j TM_{MFG}.

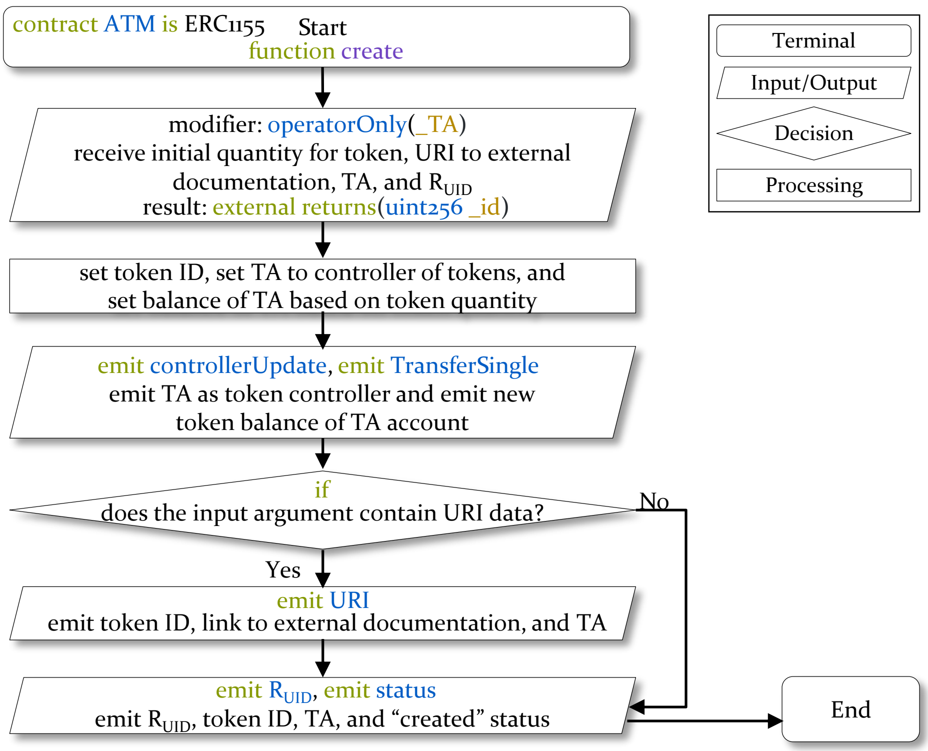


Figure 51: Code flow chart for the create function to virtualize c-objects to the blockchain

The *create* function sets the token ID and the token controller. Moreover, it updates the TA's account balance based on the token quantity. If external documents are provided to the function, a URI event is emitted to store that link address to the database. The function further logs the R_{UID} of the given token ID and sets the first status information to *created*. By setting the quantity to one, unique c-objects can be virtualized, while a quantity of more than one allows representing batches. Exemplarily, the assembly of a unique sub-module in the P2 area would trigger a create function with a quantity of one, while the production of a wire assembly in the P1 area would the quantity to the batch size (e.g. 400).

As shown in Figure 52, the *craft* function is one of the more complex functions of the ATM and provides the logic to recreate assembly processes. The function receives the input IDs and quantities of the c-objects to be consumed as well as the aspired output information of the resulting c-object including its URI, TA, and R_{UID}. The function needs sufficient input quantities for every token. Moreover, it requires the function caller to have sufficient funds and to be listed as the controller of all input token IDs. If one of the conditions is not met, an error message is given, and the crafting process is halted. This prevents TA's from using tokens for which they do not have physical responsibility.

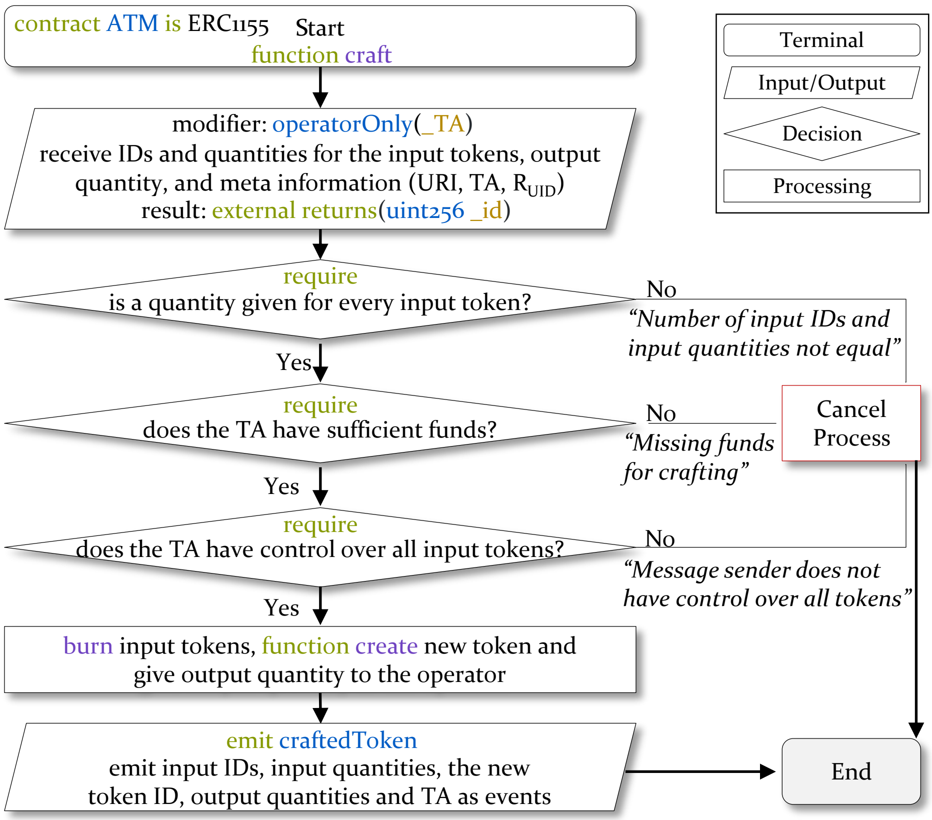


Figure 52: Representing assembly logic through the crafting function adapted from [P9]

If all conditions are met, the tokens are burned, meaning that the TAs balance is reduced by the input quantity, and the output c-object is created analogously to the create function described above. The *craftedToken* event emits all relevant data for storage, enabling comprehending the product composition (cc-links) using tracing functions.

As a last function example, the *new status* function is highlighted. The new status function allows setting quality-relevant status updates and enables the implementation of the dynamic TM_{SC}. Starting from the initial status *created*, any status information can be assigned to the c-object, such as *blocked*, *shipped*, or *in production* allowing to virtualize dynamic status changes for all c-objects in the supply chain. The code flow chart of the *new status* function is visualized in Figure 53. New status for a c-object can only be provided by the controller of a token to ensure sovereignty and data liability in the distributed environment. The logs are stored as events. This enables the status changes to be monitored using tracking functions.

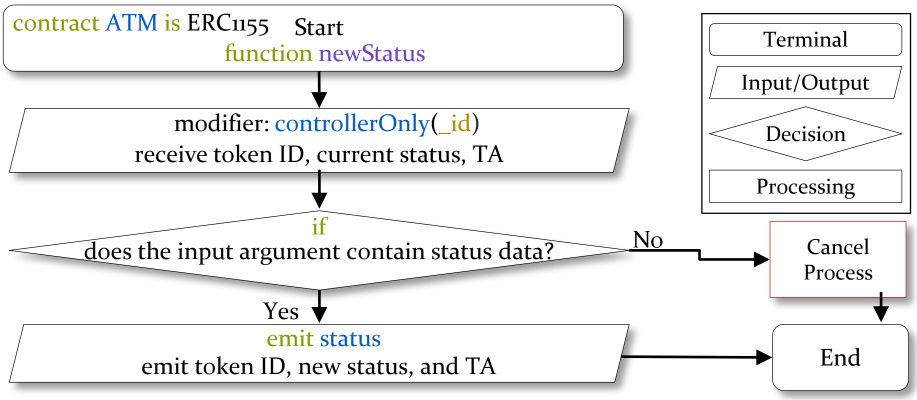


Figure 53: Code flow chart for the new status function to build the dynamic TM_{SC}

6.5 Integration of Neo4j and Ethereum in one traceability application

To automate importing c-object information to the presented ATM smart contract, a direct connection to the Neo4j database is needed. As already shown in Figure 33, the Ethereum-based TM_{SC} can then extract the c-object information from the Neo4j-based TM_{MFG}. While the smart contract's functions were previously called with placeholder data, this chapter demonstrates the full implementation for the wire harness traceability records as defined in section 6.2. To interact between the two solutions in a user-friendly way without needing to rely on console commands, a decentralized web application (DApp) was developed. The DApp automates and facilitates the integration of the two traceability models while simultaneously providing an accessible visualization platform. Moreover, it demonstrates the technical and informational compatibility of the two TMs. The data for the DApp are hosted by each TA in the network, while the DApp itself could be provided as a service by an independent traceability software company.

6.5.1 Development of the web application

The DApp's development environment is based on the JavaScript framework Vue.js for library modularization and Node.js as the java runtime environment. Moreover, web3.js libraries allow connecting to the blockchain network and Neo4j-driver provides the interface to Neo4j. Bootstrap and ECharts are further used for visualizations. As shown in Figure 54, the application consists of five main components, which are order management (1), account information (2), token profile (3), production view (4), and performance visualization (5). The DApp was developed in [S8], and its interaction with Neo4j was further refined and optimized for publication in the *Journal of Manufacturing Systems* [P7].

Orders form the entry to the TM_{MFG}, which can be selected from the dropdown menu in (1). Over the produce button, the application can interact with the ATM to create and craft the tokens of c-objects that are connected to that order. A user can thus easily create a token record for a wire harness without requiring a deep knowledge about blockchain. In (2), general account or blockchain information can be given. The third application component (3) provides traceability data based on a token ID, such as the corresponding c-object's trace reference, links to off-chain data, or the current status of the token, which facilitates status tracking for c-objects. In (4), the production view is given. For the chosen use case, the production view is divided according to three main production areas that function as trace actors (TA_{P1}, TA_{P2}, and TA_{P3}). This view was chosen due to the scope of the available c-object data, which encompasses 15 full products for one wire harness project. If the application was implemented for an entire supply chain, for example by using c-object data for the vehicle integration or by integrating different wire harness producers, the TA view could be extended to a cross-factory level. In that case, more than one contributing Neo4j TM_{MFG} model could be connected to the application.

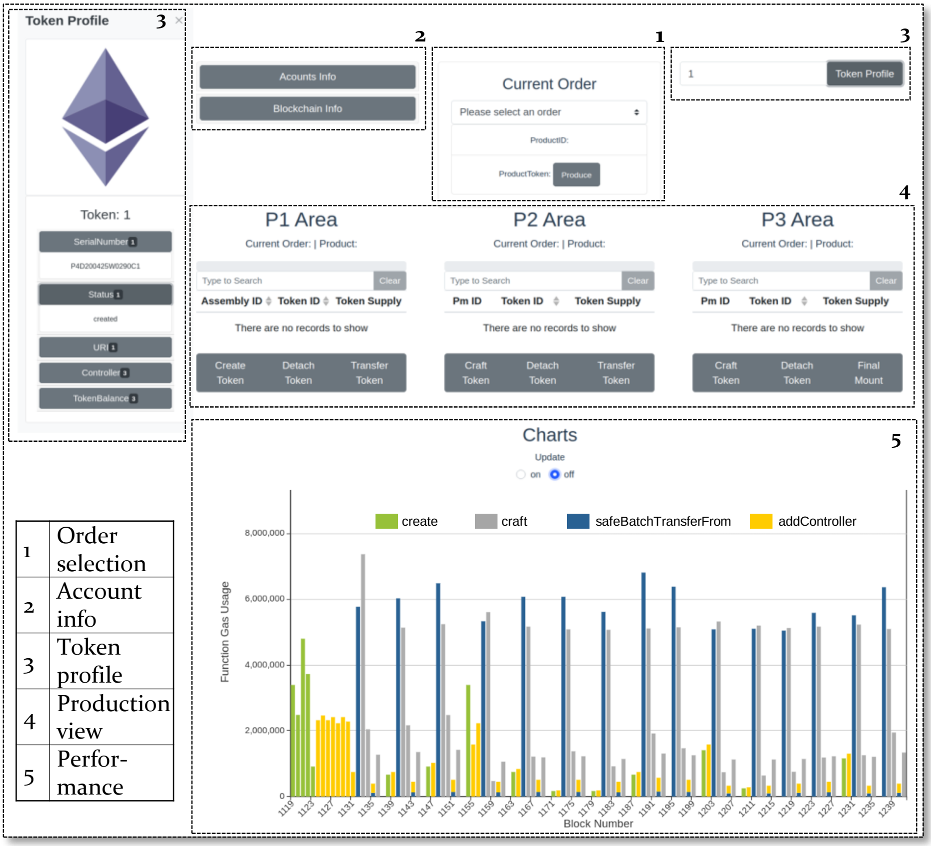


Figure 54: User interface of the DApp adapted from [S8]

Figure 55 shows a detailed view of the production window (4) for the order *o_111*. In the process bar, the user can see the progress of token creation per area. Using a filter, specific assemblies (TA_{P1}), pre-fabricated PM (TA_{P2}), or PM (TA_{P3}) can be analyzed in detail. The function panel can be used to initiate creating, crafting, transfer, or burning functions in the ATM. Exemplarily, the create button initiates a loop over all P1-items listed in the TA's production view based on the given filter. The items are collected in a list and the ATM's create function is initiated over the web3.js interface. During creation, the token supply is set to the listed batch size per c-object and the c-object's R_{UID} is stored for each token. After token creation, it retrieves the transaction receipt and logs the token's identifier, quantity, and timestamp. Using the Neo4j-driver interface, the token ID and timestamp get published to the Neo4j database. [P7, S8]

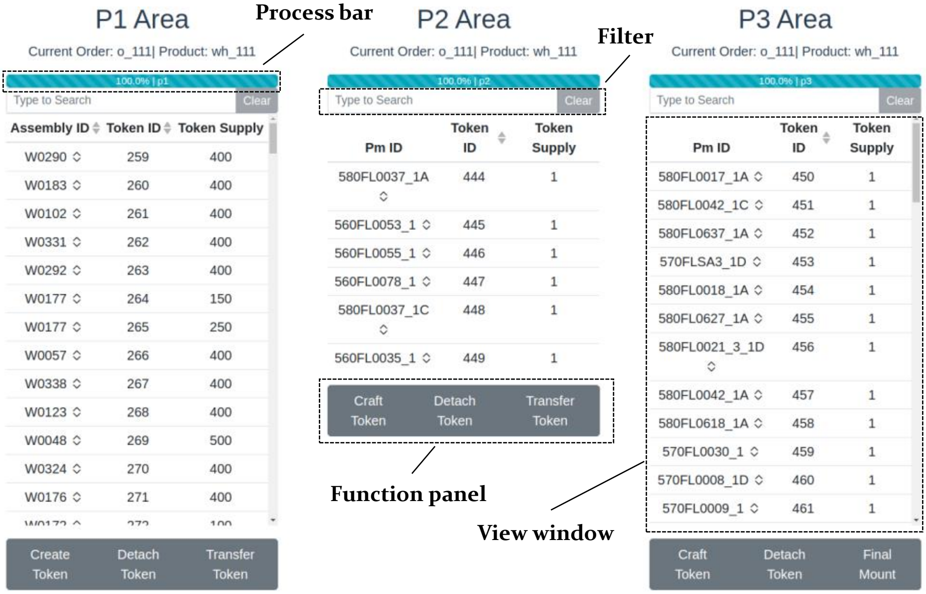


Figure 55: Detailed view of the production window in the web application [S8]

In section (5) of the DApp, the visualization based on the ECharts library allows assessing the performance of the solution through the Ethereum-native performance indicator *gas*. This section can be used to monitor the gas usage for a defined block range or visualize gas consumption per TA or per order. Moreover, different functions can be selected for the bar charts to show their respective share of the overall computational costs. This part of the DApp facilitates performance assessment and is be examined more closely in section 7.

6.5.2 System synthesis and interaction

Each of the functions described in the previous section requires the application to derive the c-object structure for a given product or order from the Neo4j database. The connection to the Neo4j database is realized by importing the Neo4j- driver, which allows linking the web application with the database. Exemplarily, the application runs a MATCH command for the filtered order in the Neo4j database and uses the defined tracing functions (e.g. Code 3 and Code 5) along with the co- and cc-links to derive the c-configuration and its properties. Moreover, the implemented cr-links allows retrieving c-object data for a specific TA by filtering the Neo4j model to objects in the P3 area. The application derives its functionalities using the standardized semantics of the defined object and link schema. The preceding modeling work thus significantly facilitates subsequent application and function development, e.g. by using the semantics of links or objects as function filters.

From a system integration perspective, the TM_{MFG} and the TM_{SC} both need to store links to the other system to form bi-directional model entry points. The shared product perspective in the blockchain thus needs to include the c-objects' R_{UID}, which is logged to the token during token creation (see Figure 51). The R_{UID} can be easily retrieved from the Neo4j model through MATCH functions or even read from the c-object itself through scanning. Moreover, the token ID should be pushed back to Neo4j as the token ID provides the c-object's secondary trace reference as maintained by the TM_{SC}. If Neo4j stores the token ID to its c-object graph, tracing functions across systems are easier to implement. If the TM_{SC} identifies a problem with a c-object in its supply chain, the status of that token ID could be set to *blocked* using the *new status* function. Then, the token ID can be used as an entry to the Neo4j model to traverse the detailed traceability graph, possibly containing data about the root cause of the quality problem. As the detailed process, resource, and order data are not stored in the blockchain, a clear link between those systems is thus needed to dive from a higher level traceability problem to more details available in the TM_{MFG}. Code 7 shows a function developed for the DApp, which pushes the token identifier from the blockchain storage to the corresponding c-objects in the Neo4j database. The function identifies the right c-objects through the R_{UID} in the token argument (lines 42-43) and matches the corresponding nodes using the ":IS_C_UID" trace link (lines 44-45).

Code 7: Expression to push token information to the Neo4j database [S8]

```
42 updateAssemblyTokens: async function ({Ruid, tokenID
43         tokenSupply, timeStamp}) {
44   const expression = `MATCH (a:AssemblyUID {assemblyUID:
45         '${Ruid}'}) - [:IS_C_UID] - (b:Assembly)
46         MERGE (b) - [:HAS_TOKEN {timeStamp:'${timeStamp}'}] →
47         {tokenID :'${tokenID}', tokenSupply: ${tokenSupply}}) -
48         [:HAS_WUID {timeStamp:'${timeStamp}'}] → (a)`
49   const query = (tx) → {
50     return tx.run(expression)
51   }
52   const session = driver.session()
53   try {
```

```
54      await session.writeTransaction(tx → query(tx)).then(() →
55          session.close())
56    } catch (err) {
57      console.error('Update Assembly Token failed: ', err)
58    }
59    return true
60  }
```

After identifying the right c-objects, a new node for the token is created using the MERGE command and linked to the c-object over a ":HAS_TOKEN" link. The token ID, the timestamp of token creation, and the token quantity are set as properties (lines 46-48). The transaction is then written to the Neo4j database (lines 49-55). If an error occurs, for example, if an R_{UID} cannot be found in the database, a console log is created and an error message is fired (lines 56-58). The result of the query is shown in Figure 56.

Figure 56: Neo4j graph before (a) and after token creation (b) adapted from [S8]

The DApp's integration functionalities can be summarized as follows:

DApp-Neo4j-driver-Neo4j interface

- Requesting data: The DApp requests a Neo4j connection and runs a database query to identify the objects and links of interest.
- Returning data: If the objects or links are identified successfully, the requested data are outputted, otherwise, an error message is logged.
- Writing data: The DApp calls a write-transaction (e.g. Code 7) to create token nodes and link them to their corresponding c-object.

DApp-web3.js-Ethereum interface

- Sending transactions: The DApp sends transactions to the ATM to issue create, craft, transfer, or new status functions.

- Returning receipts: The DApp subscribes to blockchain events and feeds its objects by extracting data from the transaction receipts.

6.5.3 Analysis of the wire harness data through tracking and tracing functions

Through the DApp, c-object tracing functions, as well as status tracking functions, can be implemented for the TM_{SC}. The c-object structure from a supply chain perspective can be identified through the *craft* logs, while status changes are retrieved from the *new status* logs. To build tracking and tracing functions, it is thus essential to understand how the traceability data are stored in the distributed ledger. The following event log shows the data storage created through a *craftedToken* event. The event log in Code 8 shows how data are stored for the *craftedToken* event. The log enables the retrieval of the TA, the input token IDs, the input quantity per ID, and the output token IDs including the quantity.

Code 8: JSON event log where redundant information were omitted for clarity [P7]

```
address: "0x9Be12Be81e425e3d26480609d1EA2668829cBC98" //ATM address
blockHash: "0xc708b22ab778788..." //can be used to verify the block
blockNumber: 88 //can be used to get the timestamp
event: "craftedToken"
_actor: "0x994af70700..." //address of the TA
_inputIds: ["84", "83", "86"] //IDs of input tokens
_inputQuantities: ["1", "4", "12"] //quantities for each input token
_outputId: "301" //ID of the output token
_outputQuantity: "1" //quantity of the output token
transactionHash: "0x63f8bc3045e2....”//can be used to verify the transaction
```

To build the c-object structure over the entire supply chain, an algorithm querying over the event logs of Code 8 is implemented. The logic of the algorithm can be stated as follows [P7]:

- Identify the token that belongs to the c-object R_{UID} of interest.
- Retrieve all status events for the token.
- Filter to the *craftedToken* event.
- Log all input token IDs listed in the *craftedToken* event.
- Recursively repeat status retrieval, filter, and token ID logging until the source of the token-represented c-object graph is reached.

The resulting cc-tracing algorithm for the TM_{SC} is visualized in Figure 57 for the example *Token 19*. The algorithm derives all status events for the given token. The crafted events, visualized in green, include the identifiers of the consumed input tokens, such as *Token 13*, *Token 18*, or *Token 11*. The c-object tracing algorithm identifies all successive events until the first event is reached. The implemented cc-link tracing function thus enables deriving the product configuration structure over the entire supply chain.

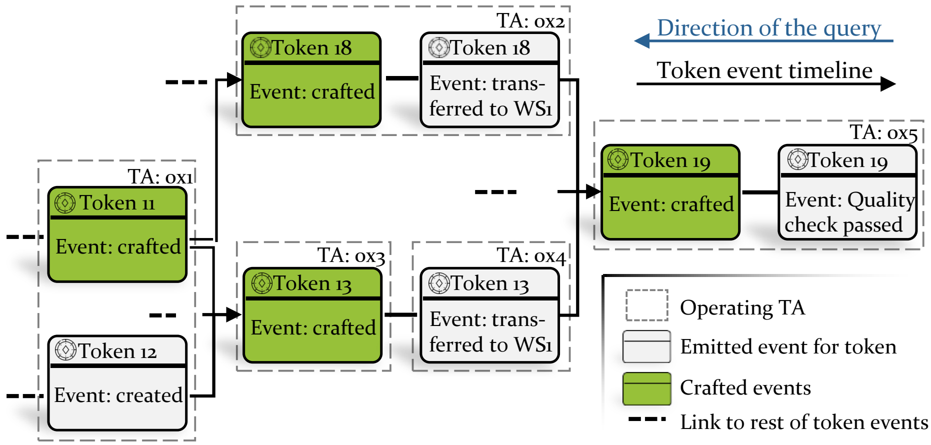


Figure 57: Tracing algorithm to derive the c-object structure from craftedToken events

In addition to cc-link tracing, status tracking needs to be implemented to monitor the dynamic TM_{SC} analogous to the TM_{MFG}. Status tracking thus refers to the c-object changes observable through status events emitted by the *new status* function. The *getPastEvent* and *tokenStatusProfile* queries are used to derive the status data. As shown in Code 9, the function allows filtering on the token ID of interest (line 66) and enables specifying the block range to be queried (lines 67-68). The status values are then retrieved and returned to the DApp's user interface (lines 70-73).

Code 9: Java script status tracing function for a given c-object [S8]

```
61 async getPastEvents (eventName, options) {
62 return await this.contract.getPastEvents(eventName, options)
63 }
64 async tokenStatusProfile () {
65    const options = {
66       filter: { _id: this.tokenID },
67       fromBlock: 0,
68       toBlock: 'latest'
```

```
    }
     this.loading = true
     const st = await app.getPastEvents('status', options)
     this.fields.Status = st.map(e => e.returnValues._status)
}
```

For both implementation technologies, it is thus possible to monitor and track dynamic status changes. Whereby the Neo4j implementation allows monitoring status updates for all object types, the blockchain implementation focuses on product status updates as specified by the reference model. Building upon this architecture, tracing functions were implemented through appropriate algorithms, as shown by the example implementations of Code 3-6 and Code 9. In the detailed manufacturing database, tracing across all link and object types was realized, such as deriving product configuration, process and resource histories, or details on order associations. In the blockchain implementation, tracing functions for product configurations were fully implemented using tokens, which allows reconstructing the final product composition throughout a decentralized supply chain. Moreover, the integration of order objects in the DApp's user interface enables a seamless entry to the corresponding manufacturing model in Neo4j. This allows initiating in-depth analysis within the manufacturing database based on order or product trace references.

7 Evaluation and contribution

In this thesis, three research questions were raised. To address RQ1 (*Are modeling methodologies from other research fields transferrable to systematically develop traceability systems in manufacturing?*), a traceability modeling methodology was developed (chapter 4). The methodology was abstracted from software modeling methodologies and aimed at providing a new systematic approach to manufacturing traceability solutions, which are often characterized by non-transferable ad-hoc implementations for specific use cases. The second research question RQ2 highlights the lack of available data models for complex manufacturing processes (*How does a data model need to be designed to enable tracking and tracing in complex manufacturing?*). The proposed TRM in section 5 demonstrates a new graph-based traceability reference model in which process- and product-driven areas are aligned and associated using uniform semantics, appropriate trace links, and order-based sub-graphs. The third research question RQ3 addresses new traceability technologies, which fit the proposed data model and use case (*Can graph databases and distributed ledger technologies effectively realize the data model and provide complete traceability in complex manufacturing?*). In section 6, a traceability system based on graph- and blockchain-based was developed for the wire harness industry. In this chapter, a full analysis and evaluation of the proposed method, data model, and technologies are given. This allows evaluating the achieved results for each research question and to verify the theoretical and practical contributions of this work. Figure 58 shows the areas of evaluation. In section 7.1, the technical performance of the proposed solution is assessed. The technical evaluation mainly refers to RQ3.

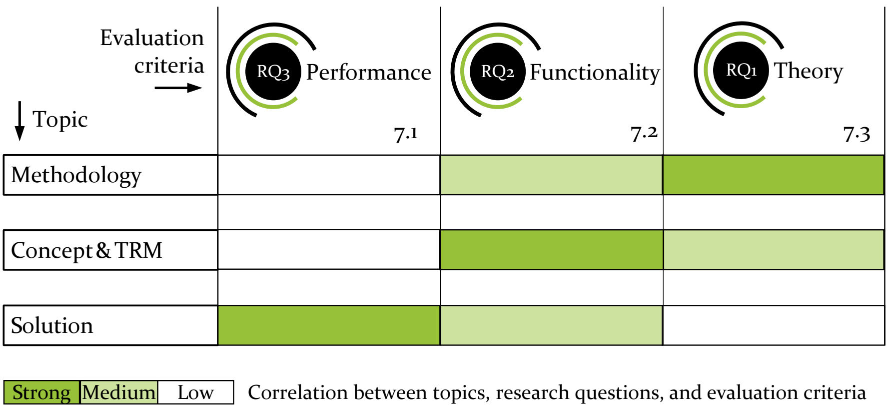


Figure 58: Scope of the evaluation section

The functional assessment in section 7.2 evaluates the proposed data model and the solution's practicability for the wire harness use case, which primarily addresses RQ2. In section 7.3, the theoretical contribution is discussed by analyzing the proposed methodology, the achieved extension of the state of the art, and the fulfillment of the overall research goals. The theoretical assessment thus primarily refers to RQ1.

7.1 Performance and technology assessment

To assess the technical performance of a traceability solution, several evaluation criteria can be applied. Cheng and Simmons [37] propose assessing a traceability system based on the retrieval's functions accuracy, completeness, and speed. The implementation of tracking and tracing functions in chapter 6 has already shown that accuracy and completeness can be fully guaranteed for the developed traceability model. Accuracy can be defined as the share of correct hits that a query generates, and completeness as the holism of the matched result space. As each function was specifically designed and optimized until it safely determined the aspired traceability result space, accuracy and completeness are near 100 % for this thesis. However, in a realistic manufacturing database, non-consistent data entries or changes would negatively influence the two indicators.

In an actual production environment, however, those two assessment criteria need to be carefully monitored as erroneous data entries could lead to incomplete or incorrect track and trace results. All developed algorithms rely on the semantics and building blocks defined for the TRM, which they apply as filters and mapping criteria to derive the aspired sub-graph or token data. Data could be stored slightly differently than specified by the TM_{MFG}, for example, due to a machine that does not issue the data with the appropriate labels or due to a TA who provides c-object data without the token association to the TM_{SC}. Then, the track and trace algorithms would be unable to identify all corresponding objects and links leading to incompleteness and accuracy losses. The fulfillment of those two criteria thus highly depends on the discipline and capability to feed the data as defined by the reference data model. Machine inabilities, for example, can be overcome using cypher queries that transform the data into the right format during import, as was demonstrated in the extraction phase in section 6.2.1. However, failures introduced through the physical process are more difficult to detect, as the correctness of the traceability record can only be ensured through consistent process management. Typical issues that generate inaccurate traceability data could be inconsistent batch management (mixed or exchanged batches), scanning of the wrong identifiers (e.g. built R_{UID} and logged R_{UID} differ), or the general loss of data through middleware downtimes and communication problems. Furthermore, deliberate data manipulation such as manual entry of incorrect data (TM_{MFG} and TM_{SC}) or subsequent modification (TM_{MFG}) of the data can not be prevented with the current system. Completeness and accuracy will thus be a pending challenge during field production, however, in the demonstrated use case, which was holistically and truthfully implemented as defined in the TRM, accuracy and completeness are both guaranteed.

As the presented solution focuses on a previously unaddressed industry in terms of traceability and employs novel traceability technologies, the systems' performance needs to be assessed regarding the technical requirements that were derived in section 6.1. The technical performance will be measured through *transaction latency* (TR10: Ability to operate in near real-time for dynamic objects with a latency below ten seconds and data retrieval below one second), *data scalability* (TR11: Ability to provide an unlimited and non-redundant amount of instantiation options for KSK wire harnesses), and *system capacity and speed* (TR13: Ability to provide fast queries and sufficient system capacity). The performance criteria introduced by Cheng and Simmons [37] are thus extended by latency, scalability, and capacity. Each of those parameters will be evaluated for both; the TM_{MFG} in Neo4j and the TM_{SC} in Ethereum. Performance tests were done with a standard laptop with an Intel Core i7 2.8 GHz processor with 16GB RAM and a 64-bit Windows 10 operating system.

For Neo4j, transaction latency can be evaluated when importing records to the database. During LOAD, CREATE, or MERGE operations, latency or time delays between actual occurrence and data appearance in the database can be derived from the Neo4j logs. Accordingly, the latency was measured for the used data sets from section 6.2 during all load and write operations. The measurement yielded 217 ms for the smallest available data sets, which yielded node or link creation in the range of 10^2, and 880 ms latency for the largest data set, which encompassed data entries in the range of 10^4. The remaining data extraction procedures ranged in between those values and were all below the stated threshold of one second. As the data sets used can be considered representative for the industry the solution is sufficiently fast for a manufacturing application with cycle times in the range of seconds (P1 area) or takt times in the range of minutes (P2 and P3 area). A more critical use case to measure latency in the TM_{MFG} is the dynamic graph. For the dynamic model, frequent and time-critical status updates are distributed over the Kafka Broker and need to be logged near-real time to provide accurate tracking functions in alignment with TR10. For the dynamic data imported to Neo4j, an average latency of 630 ms between status occurrence and storage was measured, while a standard deviation of around 300 ms showed that data are stored below one second. According to [118], a latency under 1000 ms is desirable for user applications, which can be achieved. As with any database, load and query performance in Neo4j decreases with database size, also shown in the benchmark of Dominguez-Sal et al. [140]. When deploying a traceability database, counter-measures should thus be taken in time, such as reserving more memory space, optimizing queries, or decentralizing the data to clusters [107]. As performance decreases only become measurable for large-scale applications (e.g. one million nodes) [140], it could not be investigated in this thesis.

As a second performance measure, data scalability is considered. As Neo4j was specifically developed to enable highly scalable Big Data applications, its arbitrary upper limit on the graph size is listed at around 10^{10} data entries [123] or even up to trillions of nodes [141]. Due to the extremely high volume of needed data to verify that limit, the scope of this thesis does thus not allow or require testing data scalability for the TM_{MFG}. It can be safely assumed, however, that even with increasing automation and rising complexity, the graph-based storage in Neo4j will enable sufficient scalability.

Another advantage can be identified by comparing the solution to relational systems. If a high amount of JOIN statements are needed, which is the case if data relations across domains need to be derived and aggregated, RDBMS have shown to significantly decrease in performance [99, 104, 141]. The comparison of the proposed graph-based solution with an RDBMS is visualized in Figure 59. In a typical industry RDBMS, traceability would be kept in separate, domain-based tables, which are then aggregated and combined to answer a specific track or tracing function [59]. While simple associations can be derived fast, recursive JOINs for more complex patterns highly influence the system's performance and scalability [104]. Moreover, the data are then only available at the time of querying, which makes it more difficult to identify other affected trace objects within a recall. The graph-based model maintains the data in its connected structure with links as inherent elements of the model. They can be directly retrieved with MATCH statements using the available labels on nodes and edges, which simultaneously provide a visualization to the user of how the data are related. Based on the use case of storing trace objects and trace links with equal information contribution to the overall traceability model, the graph provides a direct representation of the aspired structure.

S		D	
supplier	part	part	order ID
Cable AG	p1	p1	o-111
Cable AG	p3	p3	o-112
Electric Cpy	p8	p8	o-111

JOIN

S*D

supplier	part	order ID
Cable AG	p1	o-111
Cable AG	p3	o-112
Electric Cpy	p8	o-111

RELATIONAL

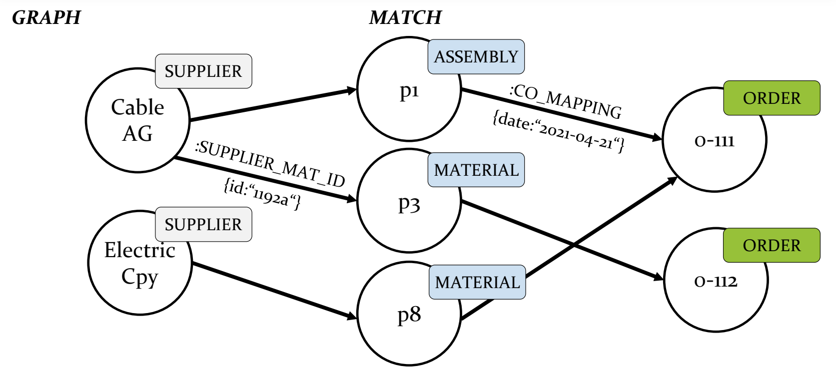


Figure 59: Comparison of JOIN statements in RDBMS and MATCH statements in graphs

To verify the performance for any given tracing query, tests measuring the query *speed* need to be conducted. For the TM_{MFG}, simple queries, such as identifying a node based on its properties or labels, are computationally fast, while more complex queries, which require deriving sub-graphs, are slower and apply iterative computation [123]. For the investigated use case, both query types are common and need to be evaluated. Accordingly, a random testing set of 20 tracing queries was defined. For each query, the size of the result graph (G) was determined by counting the amount of hit trace objects (O_{tr}) and trace links (l_{tr}). Moreover, the query speed q [ms] was measured for each query. To get a representative range of tracing queries, each trace link type was queried through a simple mapping that uses single connections and through a complex query that included multiple mappings and the deviation of sub-graphs. The algorithm developed for Code 4, which derives all r-objects associated with an o-object's trace reference, would be a representative example of a single *or*-connection. Code 5, which derives the c-objects from an order sub-graph, would be an example for a complex *co*-mapping, as it requires the deviation of sub-graphs before traversing to the objects of interest. The result of the testing series, which was also published in [P6], is shown in Table 9. The testing result shows, that linear queries like *which resources were equipped at a workstation* (rr-link) or *which wires are contained in a PM* (cc-link) perform well with a query speed mostly below 1 ms. More complex queries require subgraphs calculation and also lead to a larger result space. The complex cc-query, for example, yielded 1023 trace links, while the simple cc-query resulted in 114 trace links. Nevertheless, even for complex queries, the query speed ranges between 23 and 280 ms. Response times under 100 ms appear instant to humans and are thus desirable for data retrieval applications [118]. It can thus be said, that the graph-based TM_{MFG} fulfills all technical performance requirements for providing traceability in complex and customized manufacturing flows. It is highly suitable to store the manufacturing traceability data model for the wire harness use case. As stated before, growing data size will reduce performance and thus query speed. Analogous to any database application, the query speed should thus be monitored and performance measures taken in a timely manner (e.g. [107]).

Table 9: Performance analysis for simple and complex queries in Neo4j adapted from [P6]

	$G\ (O_{tr}, l_{tr})$ / Trace link connection type	Linear			Complex		
		O_{tr}	l_{tr}	**q [ms]**	O_{tr}	l_{tr}	**q [ms]**
CONTAINS_ty_{li}	**CC**	115	114	1	716	1023	280
	OO	9	8	1	284	369	49
	RR	5	4	1	273	535	29
	PP	7	6	1	629	685	111
$ty_{li}ty_{li}$_MAPPING	**CO**	49	72	4	126	337	30
	CR	2	1	1	211	291	27
	CP	5	4	1	70	111	22
	OR	2	1	1	196	363	130
	OP	6	4	1	155	194	80
	PR	10	9	1	85	107	23

Analogous to the TM_{MFG}, the TM_{SC} is assessed regarding latency, scalability, system capacity, and speed. For permissioned networks, the ability to operate with low latency can be ensured by choosing fast consensus algorithms and setting the blockchain's performance parameters, such as block size, block interval, or gas limit in alignment with the manufacturing speed of the use case. The time to assume a near real-time operation differs for the manufacturing-based traceability system and the blockchain-based traceability system. While in manufacturing, status updates may be critical in the range of seconds or even milliseconds, the cross-company c-configuration and related status updates offer a macro perspective on traceability, which can have a latency of several seconds or even minutes to be still considered a real-time operation.

To assess the performance in more detail, a simulation for the production of the 15 wire harness orders was run, which were holistically virtualized to the blockchain using the ATM's *create*, *craft*, *transfer*, *add controller*, and *new status* functions. The simulation, which was also published in parts in the *Journal of Manufacturing Systems* [P7], yielded 13.226 c-objects that were virtualized to the blockchain. Moreover, it generated 15.000 blockchain transactions and 30.000 event logs. To fulfill the near real-time operation requirements, the block time was set to a low level (5 s), while the gas limit was set to 6.000.000 gas. As this research is the first to present a blockchain-based solution for tracing complex assembly structures, there are no benchmark values for performance assessment, so that the solution has to be evaluated based on industry knowledge.

The goal of the first test set was to verify that the chosen architecture has sufficient capacity to provide traceability to a wire harness supply chain. Figure 60 shows the cumulated gas usage logged during the simulation. It can be observed that the first order *o_111* generates twice as much gas as the other 14 orders. This is due to the basic materials that need to be virtualized through token creation and controller associations, whereby the subsequent orders can craft assemblies out of that available material pool. In a real production setting, incoming material could be crafted evenly distributed or before the production start so that o_112 to o-125 represent a more realistic capacity use case. For those orders, the simulation shows a cumulated gas usage between 17.6 and 25.2 million gas. [P7]

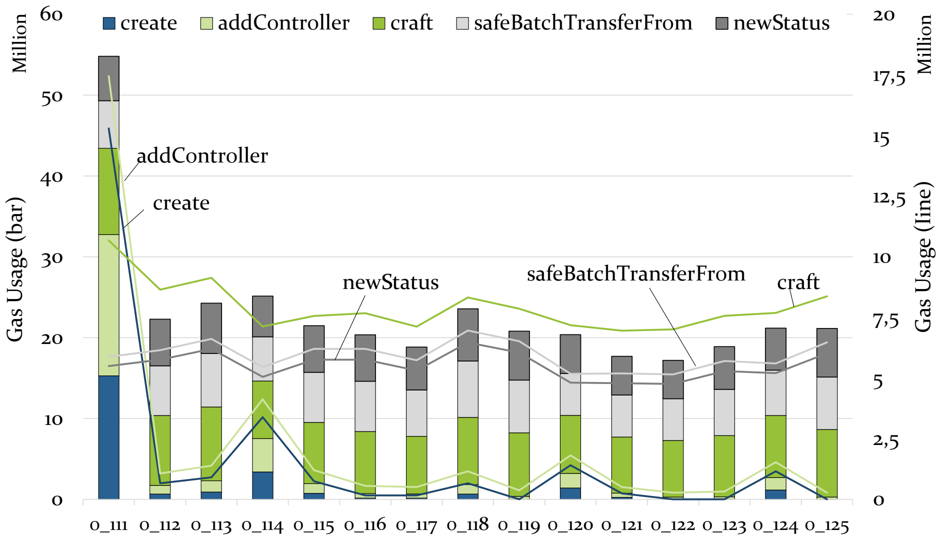


Figure 60: Gas usage for a simulation of 15 wire harnesses adapted from [P7]

The simulation further shows that the craft function is the most common and computationally intense function, averaging around eight million gas per wire harness. This goes in line with the traceability challenge observed for complex manufacturing flows that require the virtualization of frequent product merges and assembly sub-hierarchies. Additionally, the transfer function yields a high amount of gas (around 6 million gas per wire harness) as this function allows monitoring shifting TAs when the wire harnesses move through the supply chain. If c-object virtualization was distributed uniformly across all wire harness orders each final wire harness would induce around 20 million gas. As gas is not connected with a fee in the private network, it only allows indicating which functions are computationally expensive and should be, for example, grouped, optimized, or conducted outside the shift to save computing capacity. Nevertheless, in comparison with other applications, the developed solution indicates a high performance. In [119] exemplarily, the enrollment and shipment of a single product alone cost approximately 800.000 gas. A cumulated 20 million gas per order (consisting of 850 parts on average) thus yields 23.500 gas costs per part, which demonstrates the competitiveness of the solution.

To fully evaluate the capacity utilization, the maximum possible throughput of orders needs to be derived using the block time, order simulation, and gas limit. Given a typical takt time of 5 min and a typical line set-up of operating five lines in parallel with a gas limit of 6 million gas, the system would operate at a capacity utilization of around 33 %, as shown in

Figure 61, part a. This means that for that parameter setting, the throughput could be tripled until encompassing three large companies that each perform equally complex calculations for product configurations. As it can be assumed that a typical wire harness supply chain encompasses a handful of complex manufacturing companies (e.g. two to three wire harness producers and one OEM) and also some non-complex manufacturing companies (second-tier provider with lower product complexity), the system's capacity should be further optimized starting from the initial set-up. Accordingly, the gas limit was gradually increased to 10 million, 15 million, and 20 million gas, while block time was kept stable at 5 s to not comprise the solution's real-time capabilities. For each setting, the simulation was run again. The analysis showed that with increasing gas limit, the system could process more orders per minute. Accordingly, the throughput could also be increased from the initial 15 to 25, 37.5, and 47.37 wire harnesses that can be processed per minute as visualized in part b. Through the increase of computing capacity per block, the blockchain's capacity utilization rate drops from 33 % to 11 % for the highest throughput case (part a). Accordingly, for the highest gas limit, the TM_{SC} could virtualize almost 50 complex manufacturing lines simultaneously, which provides sufficient capacity and scalability for the wire harness use case. [P7]

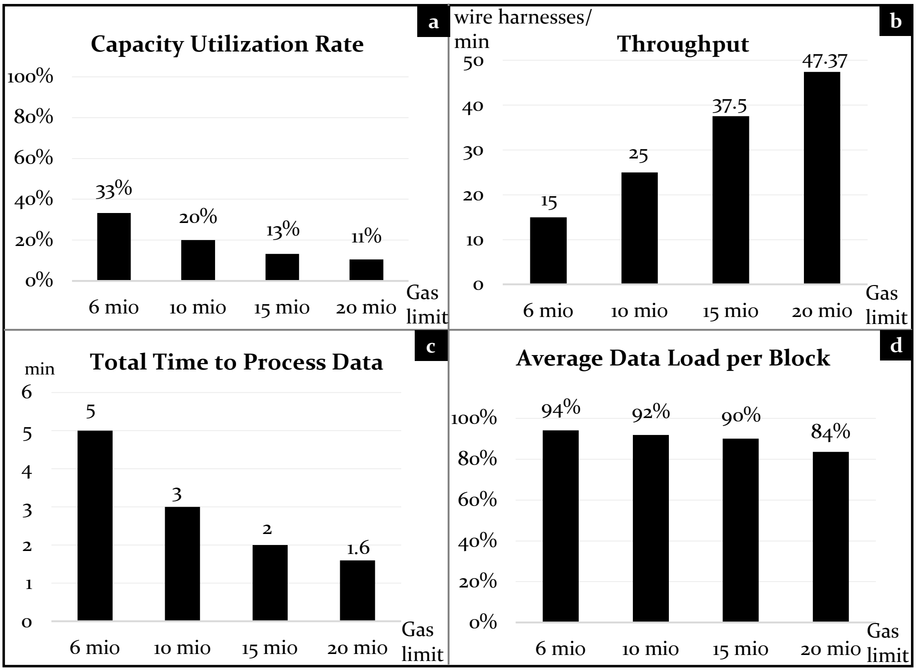


Figure 61: Correlation of capacity utilization, load, throughput, and total process time

In addition to throughput and capacity utilization, the total time to process the 15 wire harness orders in the blockchain as well as the data load per block were analyzed (part c and part d). An increase in gas limit has shown to shorten the processing time significantly from 5 min for the 6 million gas scenario to 1.6 min for the 20 million gas scenario. This can be explained by the fact that the higher gas limit means that each block can hold more transactions, leading to the orders being processed in fewer blocks. At the same time, the data load rate per block drops from 94 % to 84 %, as the provision of that extra data leads to unused block capacity. The simulation has shown, that the system is capable of operating a supply chain that builds high-volume and complex product structures, whereby the gas limit is one of the most relevant indicators to increase the system's performance. However, field tests are needed to derive the optimal gas limit and thus trade-off between throughput and computing capacity provision. Options to increase computational power or optimizing the ATM's functions regarding performance could further be considered, nevertheless, the solution's overall performance capability was verified.

As the last performance indicator, the speed to retrieve the data needs to be ensured. Four test queries were conducted for each function, and the query speed (q_1-q_4) was measured [S7]. Table 10 shows that the average query speed to derive the data from the blockchain and provide it to the user interface is higher than for the Neo4j-based TM_{MFG}. Querying the TM_{SC} is thus a bit more computationally intense, as the event tree needs to be retrieved throughout all stored blocks. Nevertheless, the average query speed is below 400 ms for all functions, which is sufficiently below the stated threshold of 1000 ms. Answering RQ3, both solutions adhere to all requested technical performance and are applicable to provide full traceability to the wire harness use case. Moreover, through their general-purpose design, they apply to analogously-characterized complex manufacturing industries.

Table 10: Query speed for the TM_{SC} adapted from [S7]

Functionality	**q_1 [ms]**	**q_2 [ms]**	**q_3 [ms]**	**q_4 [ms]**	**∅ q [ms]**
Metadata history	297	303	394	340	333
Transfer history	323	351	332	348	338
Controller history	102	156	112	119	122
Token structure	166	105	217	92	145

7.2 Functionality and practical contributions

The functional assessment aims to evaluate the ease of traceability and the solution's practical contribution compared to existing solutions. One essential aspect of functional traceability assessment is the visualization of queries [54], as the result view contributes to the system's usability [123]. Users interact with the system in two ways; the detailed TM_{MFG} can be analyzed through the Neo4j browser interface, while the integrated supply chain perspective can be accessed with the DApp as shown in Figure 62. The visualization as graphs offers an intuitive approach to traceability data since objects and links are displayed as they are stored. This provides a built-in advantage compared to other database systems that need to aggregate the data for presentation and put it into a new context. The view on track and trace results thus shows the data in its native graph structure. The readable semantics make the connections directly interpretable by the user, which constitutes a significant difference to other database systems, in which semantic links are only available during modeling but usually not during application [104]. Starting from a quality result, the user can then iteratively expand the graph in the Neo4j browser to objects of interest, which provides a user-friendly way to further explore the traceability data and discover patterns. Through the integrated function buttons and result display, the backend functions are hidden from the user, who does not need to use the console to interact with the solution. Both, the Neo4j browser and the DApp thus effectively visualize traceability results.

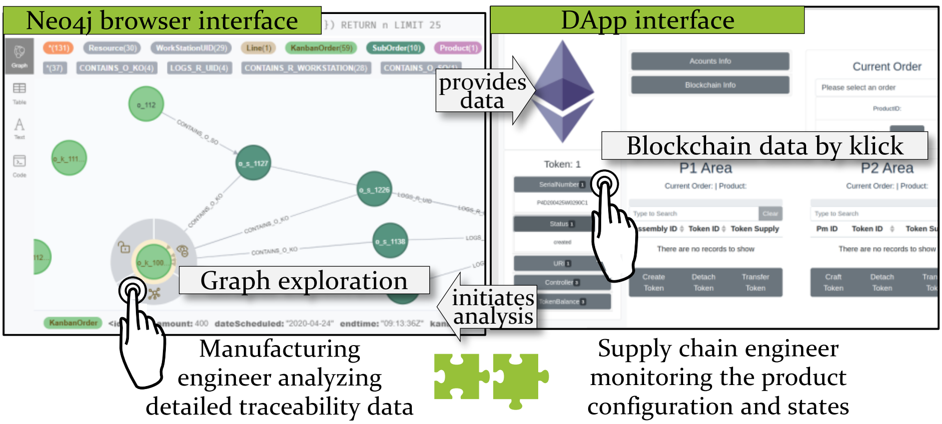


Figure 62: Usability of Neo4j and DApp interface

In addition to the visualization, the solution needs to be discussed concerning its ability to overcome the use case's traceability shortcomings. The proposed graph model allows directly connecting and semantically structuring heterogeneous data domains. Within this model, the order-based trace object and link structure makes the solution more robust against non-consistent product marking and facilitates the connection of the hybrid production logic. By connecting the orders to an order sub-graph and maintaining their relationship to corresponding c-objects, the wire harness industry can maintain recipe-driven and product-driven parts within one model, as shown in Figure 63. In the new data model, the process-driven and product-driven areas are structured analogously, building a holistic picture across all production areas. The only differences that remain are that process-driven segments are grouped into recipes and generate more actual parameter feedback than the manual assembly, which has fewer actual parameters stored but links to more c-object nodes. With increasing automation, these two sub-models will further align.

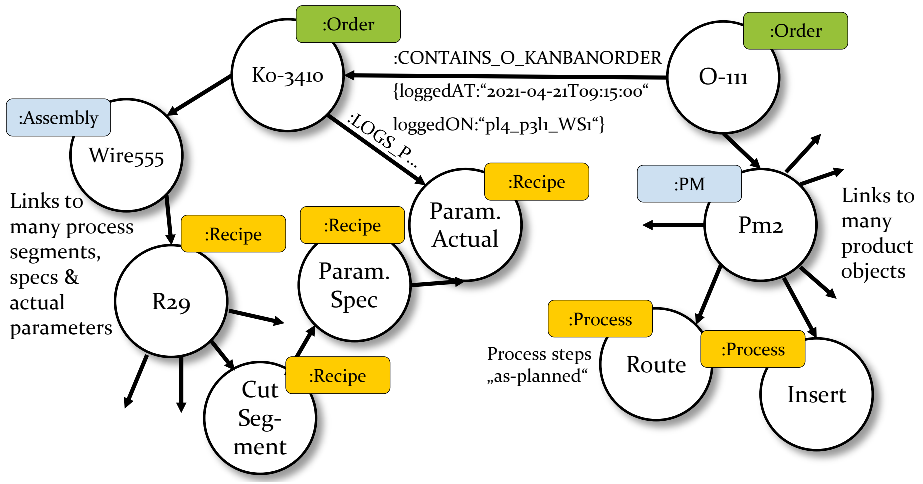


Figure 63: Connection of the hybrid production data through order-objects

The presented order sub-graph aligns well with this customized industry, which already manages process-driven areas through kanban orders and product-driven areas with (sub-)orders. Software providers can use the proposed data model to build MES that collect and connect the data with reference to the existing order structure. While the order-driven model allows connecting the data from a software perspective, the maintenance of its underlying physical integrity of order mappings will be a great challenge. Kanban orders virtually mapped to a sub-order could be exchanged or mixed in the physical flow. When applying this data model, measures for consistent order management need to be considered, for example, through FIFO consumption of materials, picking support systems, or supervised logistic flows that safely distribute the orders as virtually destined. If those measures are considered, the proposed solution overcomes the current shortcomings and enables all required tracking and tracing functions (RQ2).

In addition to the improvements realized through the data model, the two implementation technologies Neo4j and Ethereum need to be evaluated with regard to their functionality within the use case. For the manufacturing traceability data model, a technology was chosen which allows data granularity, data interconnectivity, and volume scalability. The storage of trace objects as nodes and trace links as edges facilitates the instantiation of thousands of product configurations without data duplicates. This way, KSK logic can be efficiently recreated, as objects contained in several products can be represented by the same node, from which different links to specific references or process parameters originate. As new parameters and objects can be unlimitedly added to the storage, the solution is highly applicable for growing traceability databases. While Neo4j was used for its wide application and technological maturity, other graph databases, such as OrientDB, InfiniteGraph, FlockDB, AllegroGraph, ArangoDB, Titan, or DEX, should be equally applicable [123, 124]. However, as each of them comes with its strength and weaknesses, such as read performance or write performance [124], this applicability would need to be tested, and the implementation and performance assessment would have to be repeated.

While the manufacturing model maintains detailed production data, the supply chain solution documents the macro traceability perspective. For the use case, a solution capable of operating in a decentralized environment was required with long-term storage capabilities that adhere to the long recall cycles of the automotive industry. The Ethereum-based system allows fulfilling the requirements of providing a consistent, trusted, and binding database with shared responsibility and liability for all parties contributing to the final product. The usage of the ERC 1155 standard efficiently virtualized the product histories across different trace actors, which addresses the industry's most common failure of faulty product configurations. These are now detectable, as every participant has a copy of the blockchain and also assignable, as responsible TAs are logged. Analogous to the graph's substitutes, the blockchain solution could also be implemented using another framework, such as Hyperledger Fabric or IOTA, however, due to the non-availability of the ERC 1155 standard, the implementation would require to recreate the applied assembly token logic.

Despite the high applicability and advantages of the chosen technologies, some shortcomings should be discussed. Both technologies are comparably new and have only recently been suggested for traceability use cases. Accordingly, they are just reaching business maturity and require difficult-to-acquire experts for implementation and maintenance. Moreover, the implementation of a blockchain-based solution would imply a technical consortium of OEM, wire harness manufacturers, and second-tiers who participate on an equal footing in setting up and maintaining the technology. At this point, it must be said that the solution only makes sense if all companies develop, maintain, and contribute equally, which might pose a major challenge for the hierarchical automotive industry. An independent auditing party or traceability software provider might be needed to moderate the consolidation process and develop the functions.

7.3 Theoretical contribution

In this last section, the theoretical contribution of this thesis is outlined. In alignment with the shortcomings highlighted in the state of the art, the theory discussion is clustered into method discussion, model contribution, and technological novelty.

From a methodological point of view, the state of the art has shown that manufacturing traceability systems are developed as technologically tailored solutions for narrow-defined domains. These solutions often lack transferability and are difficult to adapt to new requirements and trends. Learning from the more mature field of software traceability, this thesis developed a systematic traceability modeling methodology. The methodology starts with the definition of the traceability building blocks, such as trace objects, trace links, trace references, tracking and tracing functions, as well as trace actors. As traceability systems include very different data types, actors, and functionalities depending on the traceability goal and use case, the building blocks provide a common terminology and structure for systematic conceptualization and implementation. Moreover, as the developed methodology separates traceability modeling (TRM) and implementation (TM), the presented data model can also be implemented with other technologies. Exemplarily, an RDBMS for manufacturing data and document-based storage for SCM data could theoretically be used, which offers more flexibility during implementation. As this methodology was developed based on software best practices and manufacturing-specific requirements, it allows a uniform description of traceability terminology and the use of systematic abstractions mechanism to model different traceability aspects. The modeling method itself is thus applicable to any manufacturing industry.

In addition to the methodological contribution, the proposed data model extends the literature through its graph structure and its novel use case. Complex manufacturing flows have not been the focus of traceability research insofar as available traceability solutions mostly addressed linear and deterministic process flows, in which all data are linked to a pre-defined product trace reference. Accordingly, the developed data model demonstrates how complex manufacturing flows can be efficiently tracked and traced. Based on the identified importance of order objects, the commonly used PPR model should be extended to a PPRO model for customized and complex manufacturing flows. Moreover, this thesis revealed the potentials of graph-based modeling and implementation for the traceability use case, which has not been focused on in manufacturing research at the time of writing. In contrast to relational implementations that require a translation between type and instance models, the graph type and instance model are much closer related so that the used semantics can be applied during modeling and implementation. While in the more commonly used ER-modeling techniques, relations only exist in the concept phase but are not inherent to the implemented model, the here proposed data model could benefit from fully modeled and implemented links as traceability-inherent elements.

Finally, this thesis contributes to the literature by providing an implementation based on two novel traceability technologies. While blockchain has been discussed in the context of supply chain traceability, this research offers a new approach for virtualizing and tracing complex product configurations based on fungible and non-fungible tokens. The developed smart contract shows in detail how traceability logic can be virtualized to the blockchain. Moreover, this thesis applies a graph database for a manufacturing traceability implementation. This may act as an initiator to build analogous applications and help to transition from document-based and relational systems to a more connected conception of manufacturing data. Furthermore, the DApp demonstrated the technologies' compatibility and capability, which paves the way to forming more holistic traceability IT ecosystems.

8 Summary and outlook

Traceability systems can be understood as key enablers to smart manufacturing, as they provide transparency and structured documentation for product, resource, process, and order data throughout a product's value generation flow. In modern production networks, the regulatory and organizational push towards traceability is rising, which is driven by the growing recall rates and costs. Simultaneously, a technological pull induced by novel data-driven technologies opens the doors for future-oriented concepts and solutions. Especially in the automotive industry, the recall costs are on a one-time high with expected exponential growth rates for electrified and autonomous vehicles. Providing complex manufacturing networks like the automotive industry with appropriate traceability solutions, however, remains a great challenge as customized products create unique data histories that need to be efficiently maintained and linked. Moreover, they are produced in opaque and decentralized production networks, which not only necessitate new data models but also suitable storage technologies.

This thesis aims to develop a traceability model for complex manufacturing systems based on the representative use case of automotive wire harness manufacturing. In alignment with other research fields, a modeling methodology is proposed to systematically specify and implement the traceability system. Based on the traceability building blocks, a traceability reference model is built, which described the general objects, linkages, functions, and actors that need to be considered. In contrast to previous works, a graph-based modeling approach is chosen, which allows conceptualizing and implementing data relations as inherent characteristics of the traceability model. In this thesis, manufacturing-relevant data, supply-chain-relevant data, as well as static and dynamic data are differentiated which enables a detailed assessment of relevant object types and linkages and facilitates the translation into suitable technologies. In future works, this thesis can be used to develop a traceability ontology based on the ISA-95 standard and the here proposed traceability semantic. However, as ontologies have not been able to assert themselves for years, this thesis promotes the prevailing approach of applying user-specific semantics. Nevertheless, should ontologies regain momentum, they could facilitate the modeling and transfer of traceability solutions.

For the manufacturing model, a graph-based implementation based on Neo4j is chosen. The graph database allows maintaining direct relationships across all object types and combining static and dynamic data within one connected property graph. The supply chain model is implemented using an Ethereum-based permissioned blockchain that applies fungible and non-fungible tokens to virtualize products. The blockchain-based application realizes the macro perspective on traceability data through a trusted and immutable database that can be equally maintained by all companies contributing to the final product. For both solutions, dynamic state changes are incorporated which allow tracking trace object transformations. By interacting through a custom decentralized application and the Neo4j browser interface, product histories and states, as well as process, resource, and order information can be intuitively derived. During performance assessment, the technologies' capabilities to enable full and real-time traceability to a mass-customized production flow are verified. The technological choice is made based on use case requirements and technological potential as assessed in scientific literature as well as experiences from first prototypes. A technological benchmark, which would require a full repetition of the implementation, integration, and analysis phases, would allow a more detailed assessment and evaluation of suitable traceability technologies. This work could thus be enhanced by benchmarking different solutions within the technological scope, for example by using other graph databases or blockchain frameworks. Moreover, benchmarking with relational or document-based systems could further deepen the understanding of the chosen technologies for the traceability use case.

In future works, the solution can be further enhanced by deriving more knowledge from the built database. In the age of data-driven manufacturing, artificial intelligence and data analytics play an increasing role in data-mature industries. The traceability solution can be seen as the data groundwork to build further data-driven applications. In this context, the tracking and tracing functions could be enhanced through quality analytics functions, such as predicting the product quality or building failure clusters. From a research perspective, data analytics based on graph- or blockchain-based storage has not received much attention yet. However, as part of the traceability model is fed with simulated data, practical partners that provide real manufacturing and supply chain data would be needed to implement and verify quality analytics algorithms. In addition, a general field implementation within a wire harness supply chain would contribute to verifying and extending the system. Concluding, this thesis presents a holistic approach to traceability development from methodology specification via data modeling to full implementation and testing. The resulting traceability system provides a structured and connected data ground that can be leveraged for any data-driven application in manufacturing.

Bibliography

[1] BORDEL SÁNCHEZ, B., R. ALCARRIA, D. MARTÍN und T. ROBLES. TF4SM: A Framework for Developing Traceability Solutions in Small Manufacturing Companies [online]. *Sensors (Basel, Switzerland),* 2015, **15**(11), S. 29478-29510. Verfügbar unter: doi:10.3390/s151129478

[2] TAO, F., Q. QI, A. LIU und A. KUSIAK. Data-driven smart manufacturing [online]. *Journal of Manufacturing Systems,* 2018, **48**, S. 157-169. ISSN 02786125. Verfügbar unter: doi:10.1016/j.jmsy.2018.01.006

[3] TSIRONIS, L., N. BILALIS und V. MOUSTAKIS. Using machine learning to support quality management [online]. *The TQM Magazine,* 2005, **17**(3), S. 237-248. ISSN 0954-478X. Verfügbar unter: doi:10.1108/09544780510594207

[4] KUSIAK, A. Smart manufacturing [online]. *International Journal of Production Research,* 2018, **56**(1-2), S. 508-517. ISSN 0020-7543. Verfügbar unter: doi:10.1080/00207543.2017.1351644

[5] GHOBAKHLOO, M. Determinants of information and digital technology implementation for smart manufacturing [online]. *International Journal of Production Research,* 2020, **58**(8), S. 2384-2405. ISSN 0020-7543. Verfügbar unter: doi:10.1080/00207543.2019.1630775

[6] KRAFTFAHRT BUNDESAMT. *Rückrufe* [online], 2019 [Zugriff am: 24. September 2020]. Verfügbar unter: https://www.kba.de/DE/Marktueberwachung/Rueckrufe/rueckrufe_node.html;jsessinid=7CAF303075B07F1AA562A4EC4B644B52.live11291

[7] MURPHY, F., F. PÜTZ, M. MULLINS, T. ROHLFS, D. WRANA und M. BIERMANN. The impact of autonomous vehicle technologies on product recall risk [online]. *International Journal of Production Research,* 2019, **57**(20), S. 6264-6277. ISSN 0020-7543. Verfügbar unter: doi:10.1080/00207543.2019.1566651

[8] STEINKAMP, N. und R. LEVINE. *2017 Automotive Warranty & Recall Report* [online], 2017 [Zugriff am: 25. September 2019]

[9] MARKETSANDMARKETS. *Automotive Wiring Harness Market* [online], 2018 [Zugriff am: 11. November 2019]. Verfügbar unter:

https://www.marketsandmarkets.com/Market-Reports/automtive-wiring-harness-market-170344950.html

[10] NECKENICH, J. *3D-Master-Leitungssatz– Konzept zur Entwicklung von Leitungssätzen als 3D-Master in einem realistischen, vollständigen DMU-Modell,* 2017

[11] VASCONCELOS, F.C. und R. RAMIREZ. Complexity in business environments [online]. *Journal of Business Research,* 2011, **64**(3), S. 236-241. ISSN 01482963. Verfügbar unter: doi:10.1016/j.jbusres.2009.11.007

[12] HU, S.J., X. ZHU, H. WANG und Y. KOREN. Product variety and manufacturing complexity in assembly systems and supply chains [online]. CIRP Annals, 2008, **57**(1), S. 45-48. ISSN 00078506. Verfügbar unter: doi:10.1016/j.cirp.2008.03.138

[13] EIMARAGHY, W.H. und R.J. URBANIC. Assessment of Manufacturing Operational Complexity [online]. CIRP Annals, 2004, **53**(1), S. 401-406. ISSN 00078506. Verfügbar unter: doi:10.1016/S0007-8506(07)60726-4

[14] TAVČAR, J. und J. DUHOVNIK. Engineering change management in individual and mass production [online]. Robotics and Computer-Integrated Manufac-turing, 2005, **21**(3), S. 205-215. ISSN 07365845. Verfügbar unter: doi:10.1016/j.rcim.2004.07.017

[15] NOVAK, S. und S.D. EPPINGER. Sourcing By Design: Product Complexity and the Supply Chain [online]. Management Science, 2001, **47**(1), S. 189-204. ISSN 0025-1909. Verfügbar unter: doi:10.1287/mnsc.47.1.189.10662

[16] TROMMNAU, J., J. KÜHNLE, J. SIEGERT, R. INDERKA und T. BAUERNHANSL. Overview of the State of the Art in the Production Process of Automotive Wire Harnesses, Current Research and Future Trends [online]. *Procedia CIRP,* 2019, **81**, S. 387-392. ISSN 22128271. Verfügbar unter: doi:10.1016/j.procir.2019.03.067

[17] LAUKKONEN, J. What Is Drive-By-Wire Technology? [online]. A brief look at how electronic driving systems work, 2021 [Zugriff am: 21. November 2021]. Verfügbar unter: https://www.lifewire.com/what-is-drive-by-wire-534825

[18] ERICH, E. Standardisierung und Innovationen im Bordnetz [online]. *ATZelektronik,* 2007, **2**(4), S. 60-64. ISSN 1862-1791. Verfügbar unter: doi:10.1007/BF03223883

[19] CEVIKCAN, E. und M.B. DURMUSOGLU. An integrated job release and scheduling approach on parallel machines: An application in electric wire-harness industry [online]. *Computers & Industrial Engineering,* 2014, **76**, S. 318-332. ISSN 03608352. Verfügbar unter: doi:10.1016/j.cie.2014.08.012

[20] SCHLAICH, G. *Kabelbaummontage mit Industrierobotern.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. ISBN 978-3-540-19301-2

[21] EMMERICH, H.H. *Flexible Montage von Leitungssätzen mit Industrierobotern.* Berlin: Springer, 1992. IPA-IAO Forschung und Praxis. Bd. 160. ISBN 3-540-55227-8

[22] LAWRENCE, A. und J. VERWEY. The Automotive Semiconductor Market –Key Determinants of U.S. Firm Competitiveness [online]. *U.S. International Trade Commission (USITC),* **2019** [Zugriff am: 12. Juni 2021]. Verfügbar unter: https://www.usitc.gov/publications/332/executive_briefings/ebot_amanda_lawrence_john_verwey_the_automotive_semiconductor_market_pdf.pdf

[23] CANKUT, Y. Kabelbäume automatisiert fertigen [online]. *ATZproduktion,* 2011, **4**(1), S. 28-31. ISSN 1865-4908. Verfügbar unter: doi:10.1365/s35726-011-0008-1

[24] REINHART, G. *Flexible Automatisierung der Konstruktion und Fertigung elektrischer Leitungssätze.* Berlin: Springer, 1988. IWB-Forschungsberichte / Institut für Werkzeugmaschinen und Betriebswissenschaften, Technische Universität München. Bd. 12. ISBN 3-540-19003-1

[25] LINN, J., RICHARD und L. HUNSZU. An automatic assembly liaison extraction method and assembly liaison model [online]. *IIE Transactions,* 1999, **31**(4), S. 353-363. ISSN 0740-817X. Verfügbar unter: doi:10.1080/07408179908969839

[26] RACHURI, S., Y.-H. HAN, S. FOUFOU, S.C. FENG, U. ROY, F. WANG, R.D. SRIRAM und K.W. LYONS. A Model for Capturing Product Assembly Information [online]. *Journal of Computing and Information Science in Engineering,* 2006, **6**(1), S. 11-21. ISSN 0010-4485. Verfügbar unter: doi:10.1115/1.2164451

[27] MATÍAS, J.H., H.P. GARCÍA, J.P. GARCÍA und A.V. IDOIPE. Automatic generation of a bill of materials based on attribute patterns with variant specifications in a customer-oriented

environment [online]. *Journal of Materials Processing Technology,* 2008, **199**(1-3), S. 431-436. ISSN 09240136. Verfügbar unter: doi:10.1016/j.jmatprotec.2007.08.038

[28] BOGNER, R., P.J. HOYER, J. KYRIAZIS und G. URBAN. VOBES: Das Volkswagen-Bordnetze Entwicklungssys-tem. In: Elektronik im Kraftfahrzeug. Tagung Baden-Baden, 8. und 9. Oktober 1998 = Electronic systems for vehicles. Düsseldorf: VDI-Verl., 1998, S. 1085-1100. ISBN 3180914157

[29] HEISLER, P., P. STEINMETZ, I.S. YOO und J. FRANKE. Automatization of the Cable-Routing-Process within the Automated Production of Wiring Systems [online]. *Applied Mechanics and Materials,* 2017, **871**, S. 186-192. Verfügbar unter: doi:10.4028/www.scientific.net/AMM.871.186

[30] ABDULMALEK, F.A., J. RAJGOPAL und K.L. NEEDY. A Classification Scheme for the Process Industry to Guide the Implementation of Lean [online]. *Engineering Management Journal,* 2006, **18**(2), S. 15-25. ISSN 1042-9247. Verfügbar unter: doi:10.1080/10429247.2006.11431690

[31] HANDFIELD, R.B. und E.L. NICHOLS. *Introduction to supply chain management.* Upper Saddle River, N.J.: Prentice-Hall, 1999. ISBN 9780136216162

[32] BAQAR RAZA, M. und R. HARRISON. Design, Development & Implementation Of Ontological Knowledge Based System For Automotive Assembly Lines [online]. *International Journal of Data Mining & Knowledge Management Process,* 2011, **1**(5), S. 21-40. ISSN 2231007X. Verfügbar unter: doi:10.5121/ijdkp.2011.1503

[33] TILINDIS, J. und V. KLEIZA. The Effect of Learning Factors due to Low Volume Order Fluctuations in the Automotive Wiring Harness Production [online]. *Procedia CIRP,* 2014, **19**, S. 129-134. ISSN 22128271. Verfügbar unter: doi:10.1016/j.procir.2014.05.019

[34] YIN, R.K. *Case study research. Design and methods.* 4. ed. Los Angeles: Sage, 2009. Applied social research methods series. 5. ISBN 1412960991

[35] EISENHARDT, K.M. Building Theories from Case Study Research [online]. *Academy of Management Review,* 1989, **14**(4), S. 532-550. ISSN 0363-7425. Verfügbar unter: doi:10.5465/amr.1989.4308385

[36] HAMROL, A., D. KOWALIK und A. KUJAWIŃSK. Impact of selected work condition factors on quality of manual assembly process [online]. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 2011, **21**(2), S. 156-163. ISSN 10908471. Verfügbar unter: doi:10.1002/hfm.20233

[37] CHENG, M.J. und J.E.L. SIMMONS. Traceability in Manufacturing Systems. *International Journal of Operations & Production Management,* 1994, **14**(10), S. 4-16

[38] JANSEN-VULLERS, M.H., C.A. VAN DORP und A.J.M. BEULENS. Managing traceability information in manufacture [online]. *International Journal of Information Management,* 2003, **23**(5), S. 395-413. ISSN 02684012. Verfügbar unter: doi:10.1016/S0268-4012(03)00066-5

[39] WANK, A. *Methodik zur Wertstromintegration einer aktiven Bauteilrückverfolgung in die diskrete Variantenfertigung.* Schriftenreihe des PTW: "Innovative Fertigungstechnik". Aachen: Shaker Verlag, 2019. ISBN 978-3-8440-6554-1

[40] HU, J., X. ZHANG, L.M. MOGA und M. NECULITA. Modeling and implementation of the vegetable supply chain traceability system [online]. *Food Control,* 2013, **30**(1), S. 341-353. ISSN 09567135. Verfügbar unter: doi:10.1016/j.foodcont.2012.06.037

[41] AGRAWAL, R., A. CHEUNG, K. KAILING und S. SCHONAUER. Towards Traceability across Sovereign, Distributed RFID Databases. In: *2006 10th International Database Engineering and Applications Symposium (IDEAS'06):* IEEE, 2006, S. 174-184

[42] RINGSBERG, H. Perspectives on food traceability: a systematic literature review [online]. *Supply Chain Management: An International Journal,* 2014, **19**(5/6), S. 558-576. Verfügbar unter: doi:10.1108/SCM-01-2014-0026

[43] KARLSEN, K.M., B. DREYER, P. OLSEN und E.O. ELVEVOLL. Literature review: Does a common theoretical framework to implement food traceability exist? [online]. *Food Control,* 2013, **32**(2), S. 409-417. ISSN 09567135. Verfügbar unter: doi:10.1016/j.foodcont.2012.12.011

[44] DUPUY, C., V. BOTTA-GENOULAZ und A. GUINET. Batch dispersion model to optimise traceability in food industry [online]. *Journal of Food Engineering,* 2005, **70**(3), S. 333-339. ISSN 02608774. Verfügbar unter: doi:10.1016/j.jfoodeng.2004.05.074

[45] BECHINI, A., M.G.C.A. CIMINO, F. MARCELLONI und A. TOMASI. Patterns and technologies for enabling supply chain traceability through collaborative e-business [online]. *Information and Software Technology,* 2008, **50**(4), S. 342-359. ISSN 09505849. Verfügbar unter: doi:10.1016/j.infsof.2007.02.017

[46] IEEE, *Standard Glossary of Software Engineering Terminology.* Piscataway, NJ, USA: IEEE

[47] LINDVALL, M. und K. SANDHAL. Practical Implications of Traceability. *Software - Practice and Experience,* 1996, **26**(10), S. 1161-1180

[48] DRIVALOS, N., D.S. KOLOVOS, R.F. PAIGE und K.J. FERNANDES. Engineering a DSL for Software Traceability. In: D. GAŠEVIĆ, R. LÄMMEL, E. VAN WYK und D. GAÉSEVIC, Hg. *Software language engineering. First international conference, SLE 2008, Toulouse, France, September 29 - 30, 2008; revised selected papers.* Berlin: Springer, 2009, S. 151-167. ISBN 978-3-642-00433-9

[49] RAMESH, B. und M. JARKE. Toward reference models for requirements traceability [online]. *IEEE Transactions on Software Engineering,* 2001, **27**(1), S. 58-93. Verfügbar unter: doi:10.1109/32.895989

[50] GOTEL, O.C.Z. und C.W. FINKELSTEIN. An analysis of the requirements traceability problem. In: *Proceedings of IEEE International Conference on Requirements Engineering:* IEEE Comput. Soc. Press, 18. April 1994, S. 94-101. ISBN 0-8186-5480-5

[51] GOKNIL, A., I. KURTEV und K. VAN DEN BERG. Change Impact Analysis based on Formalization of Trace Relations for Requirements. *ECMDA Traceability Workshop (ECMDA-TW) 2008 Proceedings,* 2008

[52] KÖNIGS, S.F., G. BEIER, A. FIGGE und R. STARK. Traceability in Systems Engineering – Review of industrial practices, state-of-the-art technologies and new research solutions [online]. *Advanced Engineering Informatics,* 2012, **26**(4), S. 924-940. ISSN 14740346. Verfügbar unter: doi:10.1016/j.aei.2012.08.002

[53] GRUNDY, P. Traceability's Impact On Manufacture. *Circuits Assembly,* 2005, **16**(1), S. 48

[54] ADERSBERGER, J. *Modellbasierte Extraktion, Repräsentation und Analyse von Traceability-Informationen,* 2012

[55] ESPINOZA, A., P. ALARCON und J. GARBAJOSA. Analyzing and Systematizing Current Traceability Schemas. In: *2006 30th Annual IEEE/NASA Software Engineering Workshop:* IEEE, 24. April 2006 - 28. April 2006, S. 21-32. ISBN 0-7695-2624-1

[56] ANQUETIL, N., U. KULESZA, R. MITSCHKE, A. MOREIRA, J.-C. ROYER, A. RUMMLER und A. SOUSA. A model-driven traceability framework for software product lines [online]. *Software & Systems Modeling,* 2010, **9**(4), S. 427-451. ISSN 1619-1366. Verfügbar unter: doi:10.1007/s10270-009-0120-9

[57] BOUGDIRA, A., A. AHAITOUF und I. AKHARRAZ. Conceptual framework for general traceability solution: description and bases [online]. *Journal of Modelling in Management,* 2019, **15**(2), S. 509-530. Verfügbar unter: doi:10.1108/JM2-12-2018-0207

[58] FOLINAS, D., I. MANIKAS und B. MANOS. Traceability data management for food chains [online]. *British Food Journal,* 2006, **108**(8), S. 622-633. ISSN 0007-070X. Verfügbar unter: doi:10.1108/0007070 0610682319

[59] XU, D.F., Q. LI, H.-B. JUN, J. BROWNE, Y.L. CHEN und D. KIRITSIS. Modelling for product information tracking and feedback via wireless technology in closed-loop supply chains [online]. *International Journal of Computer Integrated Manufacturing,* 2009, **22**(7), S. 648-670. ISSN 0951-192X. Verfügbar unter: doi:10.1080/09511920 701675755

[60] AUNG, M.M. und Y.S. CHANG. Traceability in a food supply chain: Safety and quality perspectives [online]. *Food Control,* 2014, **39**, S. 172-184. ISSN 09567135. Verfügbar unter: doi:10.1016/j.foodcont. 2013.11.007

[61] BOSONA, T. und G. GEBRESENBET. Food traceability as an integral part of logistics management in food and agricultural supply chain [online]. *Food Control,* 2013, **33**(1), S. 32-48. ISSN 09567135. Verfügbar unter: doi:10.1016/j.foodcont.2013.02.004

[62] DABBENE, F., P. GAY und C. TORTIA. Traceability issues in food supply chain management: A review [online]. *Biosystems Engineering,* 2014, **120**, S. 65-80. ISSN 15375110. Verfügbar unter: doi:10.1016 /j.biosystemseng.2013.09.006

[63] KELEPOURIS, T., K. PRAMATARI und G. DOUKIDIS. RFID-enabled traceability in the food supply chain [online]. *Industrial Management & Data Systems,* 2007, **107**(2), S. 183-200. ISSN 0263-5577. Verfügbar unter: doi:10.1108/02635570710723804

[64] ZHONG, R.Y., G.Q. HUANG, S. LAN, Q.Y. DAI, X. CHEN und T. ZHANG. A big data approach for logistics trajectory discovery from RFID-enabled production data [online]. *International Journal of Production Economics,* 2015, **165**, S. 260-272. ISSN 09255273. Verfügbar unter: doi:10.1016/j.ijpe.2015.02.014

[65] ZAMFIR, I. Towards a mandatory EU system of due dili-gence for supply chains [online], 2020, Verfügbar unter: https://www.europarl.europa.eu/ReData/etudes/BRIE/2020/659299/EPRS_BRI(2020)659299_EN.pdf

[66] MCFARLANE, D., V. GIANNIKAS, A.C.Y. WONG und M. HARRISON. Product intelligence in industrial control: Theory and practice [online]. *Annual Reviews in Control,* 2013, **37**(1), S. 69-88. ISSN 13675788. Verfügbar unter: doi:10.1016/j.arcontrol.2013.03.003

[67] MCFARLANE, D., S. SARMA, J.L. CHIRN, C.Y. WONG und K. ASHTON. Auto ID systems and intelligent manufacturing control [online]. *Engineering Applications of Artificial Intelligence,* 2003, **16**(4), S. 365-376. ISSN 09521976. Verfügbar unter: doi:10.1016/S0952-1976(03)00077-0

[68] WU, Z., Z. MENG und J. GRAY. IoT-Based Techniques for Online M2M-Interactive Itemized Data Registration and Offline Information Traceability in a Digital Manufacturing System [online]. *IEEE Transactions on Industrial Informatics,* 2017, **13**(5), S. 2397-2405. ISSN 1551-3203. Verfügbar unter: doi:10.1109/TII.2017.2704613

[69] KHABBAZI, M.R., N. ISMAIL, M.Y. ISMAIL und S.A. MOUSAVI. Data Modeling of Traceability Information for Manufacturing Control System. In: *2009 International Conference on Information Management and Engineering:* IEEE, 3. April 2009 - 5. April 2009, S. 633-637. ISBN 978-0-7695-3595-1

[70] USMAN, Z., R.I.M. YOUNG, N. CHUNGOORA, C. PALMER, K. CASE und J.A. HARDING. Towards a formal manufacturing reference ontology [online]. *International Journal of Production Research,* 2013, **51**(22), S. 6553-6572. ISSN 0020-7543. Verfügbar unter: doi:10.1080/00207543.2013.801570

[71] LI, Y., L. WAN und T. XIONG. Product data model for PLM system [online]. *The International Journal of Advanced Manufacturing Technology,* 2011, **55**(9-12), S. 1149-1158. ISSN 0268-3768. Verfügbar unter: doi:10.1007/s00170-010-3130-4

[72] PANETTO, H., M. DASSISTI und A. TURSI. ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment [online]. *Advanced Engineering Informatics,* 2012, **26**(2), S. 334-348. ISSN 14740346. Verfügbar unter: doi:10.1016/j.aei.2011.12.002

[73] ZHUANG, C., J. GONG und J. LIU. Digital twin-based assembly data management and process traceability for complex products [online]. *Journal of Manufacturing Systems,* 2020. ISSN 02786125. Verfügbar unter: doi:10.1016/j.jmsy.2020.05.011

[74] SCHWARZ, H., J. EBERT, J. LEMCKE, T. RAHMANI und S. ZIVKOVIC. Using Expressive Traceability Relationships for Ensuring Consistent Process Model Refinement. In: R. CALINESCU, Hg. *15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), 2010. 22 - 26 March 2010, Oxford, United Kingdom ; [including] 5th International UML & AADL Workshop.* Piscataway, NJ: IEEE, 2010, x-x. ISBN 978-1-4244-6638-2

[75] SCHWARZ, H., J. EBERT und A. WINTER. Graph-based traceability: a comprehensive approach [online]. *Software & Systems Modeling,* 2010, **9**(4), S. 473-492. ISSN 1619-1366. Verfügbar unter: doi:10.1007/s10270-009-0141-4

[76] BURGDORF, M. *Über eine Million Mercedes-Benz Fahrzeuge müssen wegen spontan auslösender Airbags in die Werkstatt* [online], 2017. 19 Oktober 2017 [Zugriff am: 24. September 2020]. Verfügbar unter: https://www.rueckrufe.net/2017/10/19/ueber-eine-million-mercedes-benz-fahrzeuge-muessen-wegen-spontan-ausloesender-airbags-in-die-werkstatt/

[77] BURGDORF, M. *Beim BMW X1 von Juni bis Juli 2017 kann ein gelöster Stecker zum Verlust der Kraftübertragung führen* [online], 2017. 17 Oktober 2017 [Zugriff am: 24. September 2020]. Verfügbar unter: https://www.rueckrufe.net/2017/10/17/beim-bmw-x1-von-juni-bis-juli-2017-kann-ein-geloester-stecker-zum-verlust-der-kraftueber tragung-fuehren/

[78] CAO, H., P. FOLAN, J. MASCOLO und J. BROWNE. RFID in product lifecycle management [online]. A case in the automotive industry. *International Journal of Computer Integrated Manufacturing,* 2009, **22**(7), S. 616-637. ISSN 0951-192X. Verfügbar unter: doi:10.1080/09511920701522981

[79] HUANG, G.Q., T. QU, Y. ZHANG und H.D. YANG. RFID- enabled product-service system for automotive part and accessory manufacturing alliances [online]. *International Journal of Production Research,* 2012, **50**(14), S. 3821-3840. ISSN 0020-7543. Verfügbar unter: doi:10.1080/00207543.2011.592863

[80] MICHALOS, G., S. MAKRIS, N. PAPAKOSTAS, D. MOURTZIS und G. CHRYSSOLOURIS. Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach [online]. *CIRP Journal of Manufacturing Science and Technology,* 2010, **2**(2), S. 81-91. ISSN 17555817. Verfügbar unter: doi:10.1016/j.cirpj.2009.12.001

[81] KANG, Y.-S., H. KIM und Y.-H. LEE. Implementation of an RFID-Based Sequencing-Error-Proofing System for Automotive Manufacturing Logistics [online]. *Applied Sciences,* 2018, **8**(1), S. 109. Verfügbar unter: doi:10.3390/app8010109

[82] BUDAK, A., A. USTUNDAG, M.S. KILINC und E. CEVIKCAN. Digital Traceability Through Production Value Chain. In: A. USTUNDAG und E. CEVIKCAN, Hg. *Industry 4.0: Managing The Digital Transformation.* Cham: Springer International Publishing, 2018, S. 251-265. ISBN 978-3-319-57869-9

[83] HANKEL, M. *Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)* [online], 2015 [Zugriff am: 23. März 2021]. Verfügbar unter: https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/ZVEI-Faktenblatt-Industrie4_0-RAMI-4_0.pdf

[84] HUANG, Z., C. JOWERS, A. DEHGHAN-MANSHADI und M.S. DARGUSCH. Smart manufacturing and DVSM based on an Ontological approach [online]. Computers in Industry, 2020, 117, S. 103189. ISSN 01663615. Verfügbar unter: doi:10.1016/j.compind.2020.103189

[85] KÜHNE, T. Matters of (Meta-) Modeling [online]. *Software & Systems Modeling,* 2006, **5**(4), S. 369-385. ISSN 1619-1366. Verfügbar unter: doi:10.1007/s10270-006-0017-9

[86] CODD, E.F. Data models in database management [online]. *ACM SIGART Bulletin,* 1981, (74), S. 112-114. Verfügbar unter: doi:10.1145/960128.806891

[87] CHEN, P.P.-S. The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems,* 1976, **1**(1), S. 9-36

[88] KHABBAZI, M.R., M.D. YUSOF ISMAIL und S.A. MOUSAVI. Modeling of Traceability Information System for Material Flow Control Data. *Australian Journal of Basic and Applied Sciences,* 2010, **4**(2), S. 208-216

[89] MOUTRIE, C. *SAP ME How-To-Guide for Setting up Products:* SAP AG, 2012

[90] ZHOU, B.-h., S.-j. WANG und L.-f. XI. Data model design for manufacturing execution system [online]. *Journal of Manufacturing Technology Management,* 2005, **16**(8), S. 909-935. Verfügbar unter: doi:10.1108/17410380510627889

[91] TANG, X. und H. YUN. Data model for quality in product lifecycle [online]. *Computers in Industry,* 2008, **59**(2-3), S. 167-179. ISSN 01663615. Verfügbar unter: doi:10.1016/j.compind.2007.06.011

[92] BERGS, T., S. GIERLINGS, T. AUERBACH, A. KLINK, D. SCHRAKNEPPER und T. AUGSPURGER. The Concept of Digital Twin and Digital Shadow in Manufacturing [online]. Procedia CIRP, 2021, **101**(3), S. 81-84. ISSN 22128271. Verfügbar unter: doi:10.1016/j.procir.2021.02.010

[93] AHMAD, M., B.R. FERRER, B. AHMAD, D. VERA, J.L. MARTINEZ LASTRA und R. HARRISON. Knowledge-based PPR modelling for assembly automation [online]. *CIRP Journal of Manufacturing Science and Technology,* 2018, **21**, S. 33-46. ISSN 17555817. Verfügbar unter: doi:10.1016/j.cirpj.2018.01.001

[94] FORTINEAU, V., T. PAVIOT und S. LAMOURI. 5 root concepts for a meta-ontology to model product along its whole lifecycle* [online]. *IFAC Proceedings Volumes,* 2013, **46**(7), S. 47-52. ISSN 14746670. Verfügbar unter: doi:10.3182/20130522-3-BR-4036.00061

[95] WALLY, B., C. HUEMER und A. MAZAK. Aligning Business Services with Production Services: The Case of REA and ISA-95. In: *2017 IEEE 10th International Conference on Service-Oriented Computing and Applications. 22-25 November 2017, Kanazawa, Japan.* Piscataway, NJ: IEEE, 2017, S. 9-17. ISBN 978-1-5386-1326-9

[96] International Society of Automation: 2010, *ANSI/ISA-95 (IEC 62264)*

[97] RAMIS FERRER, B., B. AHMAD, D. VERA, A. LOBOV, R. HARRISON und J.L. MARTÍNEZ LASTRA. Product, process and resource model coupling for knowledge-driven assembly automation [online]. *at - Automatisierungstechnik,* 2016, **64**(3). ISSN 0178-2312. Verfügbar unter: doi:10.1515/auto-2015-0073

[98] ANGLES, R. und C. GUTIERREZ. Survey of graph database models [online]. *ACM Computing Surveys,* 2008, **40**(1), S. 1-39. ISSN 0360-0300. Verfügbar unter: doi:10.1145/1322432.1322433

[99] GHRAB, A., O. ROMERO, S. SKHIRI, A. VAISMAN und E. ZIMANYI. *GRAD: On Graph Database Modeling* [online], 2016 [Zugriff am: 26. Mai 2020]. Verfügbar unter: https://arxiv.org/abs/1602.00503

[100] WEISE, J., S. BENKHARDT und S. MOSTAGHIM. A Survey on Graph-based Systems in Manufacturing Processes. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI):* IEEE, 18. November 2018, S. 112-119. ISBN 978-1-5386-9276-9

[101] CODD, E.F. A relational model of data for large shared data banks. *Communications of the ACM,* 1970, **13**(6), S. 377-387

[102] CUI, Y., S. KARA und K.C. CHAN. Manufacturing big data ecosystem: A systematic literature review [online]. *Robotics and Computer-Integrated Manufacturing,* 2020, **62**, S. 101861. ISSN 07365845. Verfügbar unter: doi:10.1016/j.rcim.2019.101861

[103] GEERTS, G.L. und D.E. O'LEARY. A supply chain of things: The EAGLET ontology for highly visible supply chains [online]. *Decision Support Systems,* 2014, **63**, S. 3-22. ISSN 0167-9236. Verfügbar unter: doi:10.1016/j.dss.2013.09.007

[104] ROBINSON, I., J. WEBBER und E. EIFREM. *Graph databases. New opportunities for connected data.* 2nd ed. Sebastopol, CA: O'Reilly, 2015. ISBN 978-1-491-93200-1

[105] BONDY, J.A. und U.S.R. MURTY. *Graph theory with applications.* New York: Elsevier Science Publishing Co., Inc., 1976. ISBN 0-444-19451-7

[106] VIRGILIO, R. de, A. MACCIONI und R. TORLONE. Model-Driven Design of Graph Databases. In: E. YU, Hg. *Conceptual modeling. 33th international conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014; proceedings.* Cham: Springer, 2014, S. 172-185. ISBN 978-3-319-12205-2

[107] NEO4J INC. *The Neo4j Cypher Manual v4.2* [online], 2020 [Zugriff am: 16. Juni 2021]. Verfügbar unter: https://neo4j.com/docs/cypher-manual/current/

[108] KAMMLER, F., S. HAGEN, J. BRINKER und O. THOMAS. Leveraging the value of data-driven service system in manufacturing: a graph-based approach. *Proceedings of the 27th European Conference on Information Systems (ECIS),* 2019

[109] NAKAMOTO, S. *Bitcoin* [online]. *A Peer-to-Peer Electronic Cash System,* 2008. 2008 [Zugriff am: 13. Januar 2020]. Verfügbar unter: https://git.dhimmel.com/bitcoin-whitepaper/

[110] BUTERIN, V. *A next generation smart contract & decentralized application platform* [online]. *Ethereum white paper,* 2013 [Zugriff am: 9. Dezember 2019]. Verfügbar unter: https://github.com/ethereum /wiki/wiki/Whitepaper

[111] SAVEEN A. ABEYRATNE, RADMEHR P. MONFARED. Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger [online]. *International Journal of Research in Engineering and Technology,* 2016, **05**(09), S. 1-10. Verfügbar unter: https:// repository.lboro.ac.uk/articles/Blockchain_ready_manufacturing_supply_chain_using_distributed_ledger/9566069

[112] BAHGA, A. und V.K. MADISETTI. Blockchain Platform for Industrial Internet of Things [online]. *Journal of Software Engineering and Applications,* 2016, **09**(10), S. 533-546. Verfügbar unter: doi:10.4236/ jsea.2016.910036

[113] GEORGOUDAKIS, M., C. ALEXAKOS, A. KALOGERAS, J. GIALELIS und S. KOUBIAS. Decentralized Production control through ANSI / ISA-95 ~ based ontology and agents. In: *Proceedings / 2006 IEEE International Workshop on Factory Communication Systems. June 28th - 30th, 2006, Torino Incontra Conference Centre, Torino, Italy.*

Piscataway, NJ: IEEE Operations Center, 2006, S. 374-379. ISBN 1-4244-0379-0

[114] SAGE RESEARCH METHODS. Research Framework. In: A. MILLS, G. DUREPOS und E. WIEBE, Hg. *Encyclopedia of Case Study Research.* 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc, 2010. ISBN 9781412956703

[115] WINKLER, S. und J. von PILGRIM. A survey of traceability in requirements engineering and model-driven development [online]. *Software & Systems Modeling,* 2010, **9**(4), S. 529-565. ISSN 1619-1366. Verfügbar unter: doi:10.1007/s10270-009-0145-0

[116] SCHEER, A.-W. *Business Process Engineering.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. ISBN 978-3-642-79144-4

[117] NEEDHAM, M. und A. HODLER. *Graph Algorithms. Practical Examples in Apache Spark and Neo4j:* O'Reilly Media, Inc, 2019. ISBN 978-1-492-05781-9

[118] NIELSEN, J. *Usability engineering. Chapter 5 - Usability Heuristics.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. ISBN 978-0125184069

[119] TOYODA, K., P.T. MATHIOPOULOS, I. SASASE und T. OHTSUKI. A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain [online]. *IEEE Access,* 2017, **5**, S. 17465-17477. Verfügbar unter: doi:10.1109/ACCESS.2017.2720760

[120] MIEHLE, D., D. HENZE, A. SEITZ, A. LUCKOW und B. BRUEGGE. PartChain: A Decentralized Traceability Application for Multi-Tier Supply Chain Networks in the Automotive Industry. In: *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON),* 2019, S. 140-145. ISBN 978-1-7281-1264-0

[121] WESTERKAMP, M., F. VICTOR und A. KUPPER. Blockchain-Based Supply Chain Traceability: Token Recipes Model Manufacturing Processes. In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData),* 2018, S. 1595-1602. ISBN 978-1-5386-7975-3

[122] DIN: 1988, *DIN 44300: Information processing - Concepts - General terms*

[123] POKORNÝ, J. Graph Databases: Their Power and Limitations. In: K. SAEED und W. HOMENDA, Hg. *Computer Information Systems and Industrial Management.* Cham: Springer International Publishing, 2015, S. 58-69. ISBN 978-3-319-24368-9

[124] JOUILI, S. und V. VANSTEENBERGHE. An Empirical Comparison of Graph Databases. In: *2013 International Conference on Social Computing:* IEEE, 2013, S. 708-715. ISBN 978-0-7695-5137-1

[125] ETHEREUM FOUNDATION. *Geth Documentation* [online] [Zugriff am: 29. März 2021]. Verfügbar unter: https://geth.ethereum.org/docs/

[126] APACHE SOFTWARE FOUNDATION. *Kafka 2.8 Documentation* [online] [Zugriff am: 16. Juni 2021]. Verfügbar unter: https://kafka.apache.org/26/documentation/

[127] HELU, M., T. SPROCK, D. HARTENSTINE, R. VENKETESH und W. SOBEL. Scalable data pipeline architecture to support the industrial internet of things [online]. *CIRP Annals,* 2020, **69**(1), S. 385-388. ISSN 00078506. Verfügbar unter: doi:10.1016/j.cirp.2020.04.006

[128] YUAN, Z., B. PANG, Y. DU, X. LIU, J. YAO und C. KONG. Design and Implementation of Internet of Things Message Subscription System Based on Kafka. In: *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN):* IEEE, 12. Juni 2019 - 15. Juni 2019, S. 603-606. ISBN 978-1-7281-2184-0

[129] PEDAMKAR, P. *Introduction to Kafka Alternatives* [online] [Zugriff am: 16. Juni 2021]. Verfügbar unter: https://www.educba.com/kafka-alternatives/

[130] HASAN, H.R. und K. SALAH. Blockchain-Based Proof of Delivery of Physical Assets With Single and Multiple Transporters [online]. *IEEE Access,* 2018, **6**, S. 46781-46793. Verfügbar unter: doi:10.1109/ACCESS.2018.2866512

[131] YAFIMAVA, D. *What are consortium blockchains, and what purpose do they serve?* [online], 2019. 15 Januar 2019 [Zugriff am: 17. Mai 2021]. Verfügbar unter: https://openledger.info/insights/con sortium-blockchains/

[132] ZHENG, Z., S. XIE, H.N. DAI, X. CHEN und H. WANG. Blockchain challenges and opportunities: a survey [online]. *International Journal of Web and Grid Services,* 2018, **14**(4), S. 352. ISSN 1741-1106. Verfügbar unter: doi:10.1504/IJWGS.2018.095647

[133] SZILAGYI, P. *EIP 225: Clique proof-of-authority consensus protocol* [online], 2017 [Zugriff am: 9. Dezember 2019]. Verfügbar unter: https://eips.ethereum.org/EIPS/eip-225

[134] ANGELIS, S.d., L. ANIELLO, R. BALDONI, F. LOMBARDI, A. MARGHERI und V. SASSONE. PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain. In: *Italian Conference on Cyber Security,* 2018

[135] VUKOLIĆ, M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In: J. CAMENISCH und D. KESDOĞAN, Hg. *Open Problems in Network Security.* Cham: Springer International Publishing, 2016, S. 112-125. ISBN 978-3-319-39027-7

[136] ETHEREUM DEVELOPER DOCS. *ERC-20 Token Standard* [online] [Zugriff am: 3. Dezember 2020]. Verfügbar unter: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[137] ETHEREUM DEVELOPER DOCS. *ERC-721 Non-Fungible Token Standard* [online] [Zugriff am: 3. Dezember 2020]. Verfügbar unter: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/

[138] ENJIN. *ERC-1155: Smart Contract Sample Implementation* [online] [Zugriff am: 3. Dezember 2020]. Verfügbar unter: https://github.com/enjin/erc-1155

[139] PÉTER SZILÁGYI. *EIP 225: Clique proof-of-authority consensus protocol* [online], 2019 [Zugriff am: 9. Dezember 2019]. Verfügbar unter: https://eips.ethereum.org/EIPS/eip-225

[140] DOMINGUEZ-SAL, D., P. URBÓN-BAYES, A. GIMÉ-NEZ-VAÑÓ, S. GÓMEZ-VILLAMOR, N. MARTÍNEZ-BAZÁN und J.L. LARRIBA-PEY. Survey of Graph Data-base Performance on the HPC Scalable Graph Analysis Benchmark. In: Web-Age Information Management, S. 37-48

[141] PACKER, D. *The Future of the Intelligent Application: Scalability for Unlimited Growth* [online], 2020 [Zugriff am: 28. Mai 2021]. Verfügbar unter: https://neo4j.com/blog/future-intelligent-application-scalability-for-unlimited-growth/

Own publications referring to this work

[P1] KUHN, M., H.G. NGUYEN, H. OTTEN und J. FRANKE. Blockchain Enabled Traceability – Securing Process Quality in Manufacturing Chains in the Age of Autonomous Driving. In: *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD):* IEEE, 2018, S. 131-136. ISBN 978-1-5386-4315-0

[P2] KUHN, M. und H.G. NGUYEN. *The future of harness development and manufacturing* [online]. *Results from an expert case study,* 2019 [Zugriff am: 13. August 2019]. Verfügbar unter: https://www.faps.fau.de/seminare/the-future-of-harness-development-and-manufacturing/

[P3] KUHN, M., F. SCHAEFER und H. OTTEN. Process complexity as a future challenge – a quality management perspective [online]. *The TQM Journal,* 2018, **30**(6), S. 701-716. ISSN 1754-2731. Verfügbar unter: doi:10.1108/TQM-10-2017-0123

[P4] NGUYEN, H.G., M. KUHN und J. FRANKE. Manufacturing automation for automotive wiring harnesses [online]. *Procedia CIRP,* 2021, **97**(4), S. 379-384. ISSN 22128271. Verfügbar unter: doi:10.1016/j.procir.2020.05.254

[P5] KUHN, M. und J. FRANKE. Smart Manufacturing Traceability for Automotive E/E Systems enabled by Event-Driven Microservice Architecture. In: *2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT):* IEEE, 2020, S. 142-148. ISBN 978-1-7281-5332-2

[P6] KUHN, M. und J. FRANKE. Data continuity and traceability in complex manufacturing systems: a graph-based modeling approach [online]. *International Journal of Computer Integrated Manufacturing,* 2021, **34**(5), S. 549-566. ISSN 0951-192X. Verfügbar unter: doi:10.1080/0951192X.2021.1901320

[P7] KUHN, M., F. FUNK, G. ZHANG und J. FRANKE. Blockchain-based application for the traceability of complex assembly structures [online]. *Journal of Manufacturing Systems,* 2021, **59**, S. 617-630. ISSN 02786125. Verfügbar unter: doi:10.1016/j.jmsy.2021.04.013

[P8] MEYER, T., M. KUHN und E. HARTMANN. Blockchain technology enabling the Physical Internet: A synergetic application framework [online]. *Computers & Industrial Engineering,* 2019, **136**, S. 5-17. ISSN 03608352. Verfügbar unter: doi:10.1016/j.cie.2019.07.006

[P9] KUHN, M., F. FUNK und J. FRANKE. Blockchain architecture for automotive traceability [online]. *Procedia CIRP,* 2021, **97**, S. 390-395. ISSN 22128271. Verfügbar unter: doi:10.1016/j.procir.2020.05.256

[P10] KUHN, M., M. PAUL, F. SCHAEFER und H. OTTEN. Potentials of Blockchain Technology for the Quality Management of Interconnected Processes. *Proceedings of the 3rd ICQEM Conference, Barcelona,* 2018, S. 438-457

Students' theses referring to this work

[S1] HEIGL, R. *Analyse der zunehmenden Prozesskomplexität in der Bordnetzbranche auf Basis einer Case Study.* Masterarbeit. Erlangen, 13. Juli 2018

[S2] MILLER, S. *Ermittlung der Auswirkungen und Bedeutung der zunehmenden Prozesskomplexität für das Qualitätsmanagement auf Basis einer Case Study.* Masterarbeit. Erlangen, 28. Dezember 2017

[S3] ANFT, F. *Untersuchung und Benchmarking der Prozesskomplexität in der Bordnetzindustrie auf Basis einer Case Study.* Masterarbeit. Erlangen, 22. Oktober 2018

[S4] PAUL, M. *Blockchain technology for the quality management of complex and interconnected processes. Masterarbeit.* Erlangen, 31. März 2018

[S5] HAN, L. *Development of a Blockchain-based Tracking Network for Vehicle Components.* Masterarbeit. Erlangen, 30. September 2019

[S6] WÜTERICH, B. *einer Blockchain-basierten Anwendung zur Sicherstellung der Rückverfolgbarkeit.* Masterarbeit. Erlangen, 14. Mai 2019

[S7] FUNK, F. *Development and Implementation of a Blockchain-based Traceability Model for Automotive Electrical Systems.* Masterarbeit. Erlangen, 14. Februar 2020

[S8] ZHANG, G. *Functionality optimization of a blockchain-based traceability application and visualization of its data in a user-friendly front-end. Masterarbeit.* Erlangen, 1. November 2020

Reihenübersicht

Koordination der Reihe (Stand 2022):
Geschäftsstelle Maschinenbau, Dr.-Ing. Oliver Kreis, www.mb.fau.de/diss/

Im Rahmen der Reihe sind bisher die nachfolgenden Bände erschienen.

Band 1 – 52
Fertigungstechnik – Erlangen
ISSN 1431-6226
Carl Hanser Verlag, München

Band 53 – 307
Fertigungstechnik – Erlangen
ISSN 1431-6226
Meisenbach Verlag, Bamberg

ab Band 308
FAU Studien aus dem Maschinenbau
ISSN 2625-9974
FAU University Press, Erlangen

Die Zugehörigkeit zu den jeweiligen Lehrstühlen ist wie folgt gekennzeichnet:

Lehrstühle:

FAPS	Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik
FMT	Lehrstuhl für Fertigungsmesstechnik
KTmfk	Lehrstuhl für Konstruktionstechnik
LFT	Lehrstuhl für Fertigungstechnologie
LGT	Lehrstuhl für Gießereitechnik
LPT	Lehrstuhl für Photonische Technologien
REP	Lehrstuhl für Ressourcen- und Energieeffiziente Produktionsmaschinen

Band 1: Andreas Hemberger
Innovationspotentiale in der rechnerintegrierten Produktion durch wissensbasierte Systeme
FAPS, 208 Seiten, 107 Bilder. 1988.
ISBN 3-446-15234-2.

Band 2: Detlef Classe
Beitrag zur Steigerung der Flexibilität automatisierter Montagesysteme durch Sensorintegration und erweiterte Steuerungskonzepte
FAPS, 194 Seiten, 70 Bilder. 1988.
ISBN 3-446-15529-5.

Band 3: Friedrich-Wilhelm Nolting
Projektierung von Montagesystemen
FAPS, 201 Seiten, 107 Bilder, 1 Tab.
1989.ISBN 3-446-15541-4.

Band 4: Karsten Schlüter
Nutzungsgradsteigerung von Montagesystemen durch den Einsatz der Simulationstechnik
FAPS, 177 Seiten, 97 Bilder. 1989.
ISBN 3-446-15542-2.

Band 5: Shir-Kuan Lin
Aufbau von Modellen zur Lageregelung von Industrierobotern
FAPS, 168 Seiten, 46 Bilder. 1989.
ISBN 3-446-15546-5.

Band 6: Rudolf Nuss
Untersuchungen zur Bearbeitungsqualität im Fertigungssystem Laserstrahlschneiden
LFT, 206 Seiten, 115 Bilder, 6 Tab.
1989. ISBN 3-446-15783-2.

Band 7: Wolfgang Scholz
Modell zur datenbankgestützten Planung automatisierter Montageanlagen
FAPS, 194 Seiten, 89 Bilder. 1989.
ISBN 3-446-15825-1.

Band 8: Hans-Jürgen Wißmeier
Beitrag zur Beurteilung des Bruchverhaltens von Hartmetall-Fließpreßmatrizen
LFT, 179 Seiten, 99 Bilder, 9 Tab.
1989. ISBN 3-446-15921-5.

Band 9: Rainer Eisele
Konzeption und Wirtschaftlichkeit von Planungssystemen in der Produktion
FAPS, 183 Seiten, 86 Bilder. 1990.
ISBN 3-446-16107-4.

Band 10: Rolf Pfeiffer
Technologisch orientierte Montageplanung am Beispiel der Schraubtechnik
FAPS, 216 Seiten, 102 Bilder, 16 Tab. 1990. ISBN 3-446-16161-9.

Band 11: Herbert Fischer
Verteilte Planungssysteme zur Flexibilitätssteigerung der rechnerintegrierten Teilefertigung
FAPS, 201 Seiten, 82 Bilder. 1990.
ISBN 3-446-16105-8.

Band 12: Gerhard Kleineidam
CAD/CAP: Rechnergestützte Montagefeinplanung
FAPS, 203 Seiten, 107 Bilder. 1990.
ISBN 3-446-16112-0.

Band 13: Frank Vollertsen
Pulvermetallurgische Verarbeitung eines übereutektoiden verschleißfesten Stahls
LFT, XIII u. 217 Seiten, 67 Bilder, 34 Tab. 1990.
ISBN 3-446-16133-3.

Band 14: Stephan Biermann
Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO2-Hochleistungslasern
LFT, VIII u. 170 Seiten, 93 Bilder, 4 Tab. 1991. ISBN 3-446-16269-0.

Band 15: Uwe Geißler
Material- und Datenfluß in einer flexiblen Blechbearbeitungszelle
LFT, 124 Seiten, 41 Bilder, 7 Tab.
1991. ISBN 3-446-16358-1.

Band 16: Frank Oswald Hake
Entwicklung eines rechnergestützten Diagnosesystems für automatisierte Montagezellen
FAPS, XIV u. 166 Seiten, 77 Bilder.
1991. ISBN 3-446-16428-6.

Band 17: Herbert Reichel
Optimierung der Werkzeugbereitstellung durch rechnergestützte Arbeitsfolgenbestimmung
FAPS, 198 Seiten, 73 Bilder, 2 Tab.
1991. ISBN 3-446-16453-7.

Band 18: Josef Scheller
Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen
FAPS, 198 Seiten, 65 Bilder. 1991.
ISBN 3-446-16454-5.

Band 19: Arnold vom Ende
Untersuchungen zum Biegeumforme mit elastischer Matrize
LFT, 166 Seiten, 55 Bilder, 13 Tab.
1991. ISBN 3-446-16493-6.

Band 20: Joachim Schmid
Beitrag zum automatisierten Bearbeiten von Keramikguß mit Industrierobotern
FAPS, XIV u. 176 Seiten, 111 Bilder, 6 Tab. 1991.
ISBN 3-446-16560-6.

Band 21: Egon Sommer
Multiprozessorsteuerung für kooperierende Industrieroboter in Montagezellen
FAPS, 188 Seiten, 102 Bilder. 1991.
ISBN 3-446-17062-6.

Band 22: Georg Geyer
Entwicklung problemspezifischer Verfahrensketten in der Montage
FAPS, 192 Seiten, 112 Bilder. 1991.
ISBN 3-446-16552-5.

Band 23: Rainer Flohr
Beitrag zur optimalen Verbindungstechnik in der Oberflächenmontage (SMT)
FAPS, 186 Seiten, 79 Bilder. 1991.
ISBN 3-446-16568-1.

Band 24: Alfons Rief
Untersuchungen zur Verfahrensfolge Laserstrahlschneiden und -schweißen in der Rohkarosseriefertigung
LFT, VI u. 145 Seiten, 58 Bilder, 5 Tab. 1991. ISBN 3-446-16593-2.

Band 25: Christoph Thim
Rechnerunterstützte Optimierung von Materialflußstrukturen in der Elektronikmontage durch Simulation
FAPS, 188 Seiten, 74 Bilder. 1992.
ISBN 3-446-17118-5.

Band 26: Roland Müller
CO2 -Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen
LFT, 141 Seiten, 107 Bilder, 4 Tab.
1992. ISBN 3-446-17104-5.

Band 27: Günther Schäfer
Integrierte Informationsverarbeitung bei der Montageplanung
FAPS, 195 Seiten, 76 Bilder. 1992.
ISBN 3-446-17117-7.

Band 28: Martin Hoffmann
Entwicklung einerCAD/CAM-Prozeßkette für die Herstellung von Blechbiegeteilen
LFT, 149 Seiten, 89 Bilder. 1992.
ISBN 3-446-17154-1.

Band 29: Peter Hoffmann
Verfahrensfolge Laserstrahlschneiden und –schweißen: Prozeßführung und Systemtechnik in der 3D-Laserstrahlbearbeitung von Blechformteilen
LFT, 186 Seiten, 92 Bilder, 10 Tab. 1992. ISBN 3-446-17153-3.

Band 30: Olaf Schrödel
Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen
FAPS, 180 Seiten, 84 Bilder. 1992.
ISBN 3-446-17242-4.

Band 31: Hubert Reinisch
Planungs- und Steuerungswerkzeuge zur impliziten Geräteprogrammierung in Roboterzellen
FAPS, XI u. 212 Seiten, 112 Bilder. 1992. ISBN 3-446-17380-3.

Band 32: Brigitte Bärnreuther
Ein Beitrag zur Bewertung des Kommunikationsverhaltens von Automatisierungsgeräten in flexiblen Produktionszellen
FAPS, XI u. 179 Seiten, 71 Bilder. 1992. ISBN 3-446-17451-6.

Band 33: Joachim Hutfless
Laserstrahlregelung und Optikdiagnostik in der Strahlführung einer CO2-Hochleistungslaseranlage
LFT, 175 Seiten, 70 Bilder, 17 Tab. 1993. ISBN 3-446-17532-6.

Band 34: Uwe Günzel
Entwicklung und Einsatz eines Simula-tionsverfahrens für operative undstrategische Probleme der Produktionsplanung und -steuerung
FAPS, XIV u. 170 Seiten, 66 Bilder, 5 Tab. 1993. ISBN 3-446-17604-7.

Band 35: Bertram Ehmann
Operatives Fertigungscontrolling durch Optimierung auftragsbezogener Bearbeitungsabläufe in der Elektronikfertigung
FAPS, XV u. 167 Seiten, 114 Bilder. 1993. ISBN 3-446-17658-6.

Band 36: Harald Kolléra
Entwicklung eines benutzerorientierten Werkstattprogrammiersystems für das Laserstrahlschneiden
LFT, 129 Seiten, 66 Bilder, 1 Tab. 1993. ISBN 3-446-17719-1.

Band 37: Stephanie Abels
Modellierung und Optimierung von Montageanlagen in einem integrierten Simulationssystem
FAPS, 188 Seiten, 88 Bilder. 1993.
ISBN 3-446-17731-0.

Band 38: Robert Schmidt-Hebbel
Laserstrahlbohren durchflußbestimmender Durchgangslöcher
LFT, 145 Seiten, 63 Bilder, 11 Tab. 1993. ISBN 3-446-17778-7.

Band 39: Norbert Lutz
Oberflächenfeinbearbeitung keramischer Werkstoffe mit XeCl-Excimerlaserstrahlung
LFT, 187 Seiten, 98 Bilder, 29 Tab. 1994. ISBN 3-446-17970-4.

Band 40: Konrad Grampp
Rechnerunterstützung bei Test und Schulung an Steuerungssoftware von SMD-Bestücklinien
FAPS, 178 Seiten, 88 Bilder. 1995.
ISBN 3-446-18173-3.

Band 41: Martin Koch
Wissensbasierte Unterstützung derAngebotsbearbeitung in der Investitionsgüterindustrie
FAPS, 169 Seiten, 68 Bilder. 1995.
ISBN 3-446-18174-1.

Band 42: Armin Gropp
Anlagen- und Prozeßdiagnostik beim Schneiden mit einem gepulsten Nd:YAG-Laser
LFT, 160 Seiten, 88 Bilder, 7 Tab. 1995. ISBN 3-446-18241-1.

Band 43: Werner Heckel
Optische 3D-Konturerfassung und on-line Biegewinkelmessung mit dem Lichtschnittverfahren
LFT, 149 Seiten, 43 Bilder, 11 Tab. 1995. ISBN 3-446-18243-8.

Band 44: Armin Rothhaupt
Modulares Planungssystem zur Optimierung der Elektronikfertigung
FAPS, 180 Seiten, 101 Bilder. 1995.
ISBN 3-446-18307-8.

Band 45: Bernd Zöllner
Adaptive Diagnose in der Elektronikproduktion
FAPS, 195 Seiten, 74 Bilder, 3 Tab. 1995. ISBN 3-446-18308-6.

Band 46: Bodo Vormann
Beitrag zur automatisierten Handhabungsplanung komplexer Blechbiegeteile
LFT, 126 Seiten, 89 Bilder, 3 Tab. 1995. ISBN 3-446-18345-0.

Band 47: Peter Schnepf
Zielkostenorientierte Montageplanung
FAPS, 144 Seiten, 75 Bilder. 1995.
ISBN 3-446-18397-3.

Band 48: Rainer Klotzbücher
Konzept zur rechnerintegrierten Materialversorgung in flexiblen Fertigungssystemen
FAPS, 156 Seiten, 62 Bilder. 1995.
ISBN 3-446-18412-0.

Band 49: Wolfgang Greska
Wissensbasierte Analyse undKlassifizierung von Blechteilen
LFT, 144 Seiten, 96 Bilder. 1995.
ISBN 3-446-18462-7.

Band 50: Jörg Franke
Integrierte Entwicklung neuer Produkt- und Produktionstechnologien für räumliche spritzgegossene Schaltungsträger (3-D MID)
FAPS, 196 Seiten, 86 Bilder, 4 Tab. 1995. ISBN 3-446-18448-1.

Band 51: Franz-Josef Zeller
Sensorplanung und schnelle Sensorregelung für Industrieroboter
FAPS, 190 Seiten, 102 Bilder, 9 Tab. 1995. ISBN 3-446-18601-8.

Band 52: Michael Solvie
Zeitbehandlung und Multimedia-Unterstützung in Feldkommunikationssystemen
FAPS, 200 Seiten, 87 Bilder, 35 Tab. 1996. ISBN 3-446-18607-7.

Band 53: Robert Hopperdietzel
Reengineering in der Elektro- und Elektronikindustrie
FAPS, 180 Seiten, 109 Bilder, 1 Tab. 1996. ISBN 3-87525-070-2.

Band 54: Thomas Rebhahn
Beitrag zur Mikromaterialbearbeitung mit Excimerlasern - Systemkomponenten und Verfahrensoptimierungen
LFT, 148 Seiten, 61 Bilder, 10 Tab. 1996. ISBN 3-87525-075-3.

Band 55: Henning Hanebuth
Laserstrahlhartlöten mit Zweistrahltechnik
LFT, 157 Seiten, 58 Bilder, 11 Tab. 1996. ISBN 3-87525-074-5.

Band 56: Uwe Schönherr
Steuerung und Sensordatenintegration für flexible Fertigungszellen mitkooperierenden Robotern
FAPS, 188 Seiten, 116 Bilder, 3 Tab. 1996. ISBN 3-87525-076-1.

Band 57: Stefan Holzer
Berührungslose Formgebung mitLaserstrahlung
LFT, 162 Seiten, 69 Bilder, 11 Tab. 1996. ISBN 3-87525-079-6.

Band 58: Markus Schultz
Fertigungsqualität beim 3D-Laserstrahlschweißen vonBlechformteilen
LFT, 165 Seiten, 88 Bilder, 9 Tab. 1997. ISBN 3-87525-080-X.

Band 59: Thomas Krebs
Integration elektromechanischer CA-Anwendungen über einem STEP-Produktmodell
FAPS, 198 Seiten, 58 Bilder, 8 Tab. 1997. ISBN 3-87525-081-8.

Band 60: Jürgen Sturm
Prozeßintegrierte Qualitätssicherung in der Elektronikproduktion
FAPS, 167 Seiten, 112 Bilder, 5 Tab. 1997. ISBN 3-87525-082-6.

Band 61: Andreas Brand
Prozesse und Systeme zur Bestückung räumlicher elektronischer Baugruppen (3D-MID)
FAPS, 182 Seiten, 100 Bilder. 1997. ISBN 3-87525-087-7.

Band 62: Michael Kauf
Regelung der Laserstrahlleistung und der Fokusparameter einer CO_2-Hochleistungslaseranlage
LFT, 140 Seiten, 70 Bilder, 5 Tab. 1997. ISBN 3-87525-083-4.

Band 63: Peter Steinwasser
Modulares Informationsmanagement in der integrierten Produkt- und Prozeßplanung
FAPS, 190 Seiten, 87 Bilder. 1997. ISBN 3-87525-084-2.

Band 64: Georg Liedl
Integriertes Automatisierungskonzept für den flexiblen Materialfluß in der Elektronikproduktion
FAPS, 196 Seiten, 96 Bilder, 3 Tab. 1997. ISBN 3-87525-086-9.

Band 65: Andreas Otto
Transiente Prozesse beim Laserstrahlschweißen
LFT, 132 Seiten, 62 Bilder, 1 Tab. 1997. ISBN 3-87525-089-3.

Band 66: Wolfgang Blöchl
Erweiterte Informationsbereitstellung an offenen CNC-Steuerungen zur Prozeß- und Programmoptimierung
FAPS, 168 Seiten, 96 Bilder. 1997. ISBN 3-87525-091-5.

Band 67: Klaus-Uwe Wolf
Verbesserte Prozeßführung und Prozeßplanung zur Leistungs- und Qualitätssteigerung beim Spulenwickeln
FAPS, 186 Seiten, 125 Bilder. 1997. ISBN 3-87525-092-3.

Band 68: Frank Backes
Technologieorientierte Bahnplanung für die 3D-Laserstrahlbearbeitung
LFT, 138 Seiten, 71 Bilder, 2 Tab. 1997. ISBN 3-87525-093-1.

Band 69: Jürgen Kraus
Laserstrahlumformen von Profilen
LFT, 137 Seiten, 72 Bilder, 8 Tab. 1997. ISBN 3-87525-094-X.

Band 70: Norbert Neubauer
Adaptive Strahlführungen für CO_2-Laseranlagen
LFT, 120 Seiten, 50 Bilder, 3 Tab. 1997. ISBN 3-87525-095-8.

Band 71: Michael Steber
Prozeßoptimierter Betrieb flexibler Schraubstationen in der automatisierten Montage
FAPS, 168 Seiten, 78 Bilder, 3 Tab. 1997. ISBN 3-87525-096-6.

Band 72: Markus Pfestorf
Funktionale 3D-Oberflächenkenngrößen in der Umformtechnik
LFT, 162 Seiten, 84 Bilder, 15 Tab. 1997. ISBN 3-87525-097-4.

Band 73: Volker Franke
Integrierte Planung und Konstruktion von Werkzeugen für die Biegebearbeitung
LFT, 143 Seiten, 81 Bilder. 1998. ISBN 3-87525-098-2.

Band 74: Herbert Scheller
Automatisierte Demontagesysteme und recyclinggerechte Produktgestaltung elektronischer Baugruppen
FAPS, 184 Seiten, 104 Bilder, 17 Tab. 1998. ISBN 3-87525-099-0.

Band 75: Arthur Meßner
Kaltmassivumformung metallischer Kleinstteile – Werkstoffverhalten, Wirkflächenreibung, Prozeßauslegung
LFT, 164 Seiten, 92 Bilder, 14 Tab. 1998. ISBN 3-87525-100-8.

Band 76: Mathias Glasmacher
Prozeß- und Systemtechnik zum Laserstrahl-Mikroschweißen
LFT, 184 Seiten, 104 Bilder, 12 Tab. 1998. ISBN 3-87525-101-6.

Band 77: Michael Schwind
Zerstörungsfreie Ermittlung mechanischer Eigenschaften von Feinblechen mit dem Wirbelstromverfahren
LFT, 124 Seiten, 68 Bilder, 8 Tab. 1998. ISBN 3-87525-102-4.

Band 78: Manfred Gerhard
Qualitätssteigerung in der Elektronikproduktion durch Optimierung der Prozeßführung beim Löten komplexer Baugruppen
FAPS, 179 Seiten, 113 Bilder, 7 Tab. 1998. ISBN 3-87525-103-2.

Band 79: Elke Rauh
Methodische Einbindung der Simulation in die betrieblichen Planungs- und Entscheidungsabläufe
FAPS, 192 Seiten, 114 Bilder, 4 Tab. 1998. ISBN 3-87525-104-0.

Band 80: Sorin Niederkorn
Meßeinrichtung zur Untersuchung der Wirkflächenreibung bei umformtechnischen Prozessen
LFT, 99 Seiten, 46 Bilder, 6 Tab. 1998. ISBN 3-87525-105-9.

Band 81: Stefan Schuberth
Regelung der Fokuslage beim Schweißen mit CO2-Hochleistungslasern unter Einsatz von adaptiven Optiken
LFT, 140 Seiten, 64 Bilder, 3 Tab. 1998. ISBN 3-87525-106-7.

Band 82: Armando Walter Colombo
Development and Implementation of Hierarchical Control Structures of Flexible Production Systems Using High Level Petri Nets
FAPS, 216 Seiten, 86 Bilder. 1998. ISBN 3-87525-109-1.

Band 83: Otto Meedt
Effizienzsteigerung bei Demontage und Recycling durch flexible Demontagetechnologien und optimierte Produktgestaltung
FAPS, 186 Seiten, 103 Bilder. 1998. ISBN 3-87525-108-3.

Band 84: Knuth Götz
Modelle und effiziente Modellbildung zur Qualitätssicherung in der Elektronikproduktion
FAPS, 212 Seiten, 129 Bilder, 24 Tab. 1998. ISBN 3-87525-112-1.

Band 85: Ralf Luchs
Einsatzmöglichkeiten leitender Klebstoffe zur zuverlässigen Kontaktierung elektronischer Bauelemente in der SMT
FAPS, 176 Seiten, 126 Bilder, 30 Tab. 1998. ISBN 3-87525-113-7.

Band 86: Frank Pöhlau
Entscheidungsgrundlagen zur Einführung räumlicher spritzgegossener Schaltungsträger (3-D MID)
FAPS, 144 Seiten, 99 Bilder. 1999. ISBN 3-87525-114-8.

Band 87: Roland T. A. Kals
Fundamentals on the miniaturization of sheet metal working processes
LFT, 128 Seiten, 58 Bilder, 11 Tab. 1999. ISBN 3-87525-115-6.

Band 88: Gerhard Luhn
Implizites Wissen und technisches Handeln am Beispiel der Elektronikproduktion
FAPS, 252 Seiten, 61 Bilder, 1 Tab. 1999. ISBN 3-87525-116-4.

Band 89: Axel Sprenger
Adaptives Streckbiegen von Aluminium-Strangpreßprofilen
LFT, 114 Seiten, 63 Bilder, 4 Tab. 1999. ISBN 3-87525-117-2.

Band 90: Hans-Jörg Pucher
Untersuchungen zur Prozeßfolge Umformen, Bestücken und Laserstrahllöten von Mikrokontakten
LFT, 158 Seiten, 69 Bilder, 9 Tab. 1999. ISBN 3-87525-119-9.

Band 91: Horst Arnet
Profilbiegen mit kinematischer Gestalterzeugung
LFT, 128 Seiten, 67 Bilder, 7 Tab. 1999. ISBN 3-87525-120-2.

Band 92: Doris Schubart
Prozeßmodellierung und Technologieentwicklung beim Abtragen mit CO2-Laserstrahlung
LFT, 133 Seiten, 57 Bilder, 13 Tab. 1999. ISBN 3-87525-122-9.

Band 93: Adrianus L. P. Coremans
Laserstrahlsintern von Metallpulver - Prozeßmodellierung, Systemtechnik, Eigenschaften laserstrahlgesinterter Metallkörper
LFT, 184 Seiten, 108 Bilder, 12 Tab. 1999. ISBN 3-87525-124-5.

Band 94: Hans-Martin Biehler
Optimierungskonzepte für Qualitätsdatenverarbeitung und Informationsbereitstellung in der Elektronikfertigung
FAPS, 194 Seiten, 105 Bilder. 1999. ISBN 3-87525-126-1.

Band 95: Wolfgang Becker
Oberflächenausbildung und tribologische Eigenschaften excimerlaserstrahlbearbeiteter Hochleistungskeramiken
LFT, 175 Seiten, 71 Bilder, 3 Tab. 1999. ISBN 3-87525-127-X.

Band 96: Philipp Hein
Innenhochdruck-Umformen von Blechpaaren: Modellierung, Prozeßauslegung und Prozeßführung
LFT, 129 Seiten, 57 Bilder, 7 Tab. 1999. ISBN 3-87525-128-8.

Band 97: Gunter Beitinger
Herstellungs- und Prüfverfahren für thermoplastische Schaltungsträger
FAPS, 169 Seiten, 92 Bilder, 20 Tab. 1999. ISBN 3-87525-129-6.

Band 98: Jürgen Knoblach
Beitrag zur rechnerunterstützten verursachungsgerechten Angebotskalkulation von Blechteilen mit Hilfe wissensbasierter Methoden
LFT, 155 Seiten, 53 Bilder, 26 Tab. 1999. ISBN 3-87525-130-X.

Band 99: Frank Breitenbach
Bildverarbeitungssystem zur Erfassung der Anschlußgeometrie elektronischer SMT-Bauelemente
LFT, 147 Seiten, 92 Bilder, 12 Tab. 2000. ISBN 3-87525-131-8.

Band 100: Bernd Falk
Simulationsbasierte Lebensdauervorhersage für Werkzeuge der Kaltmassivumformung
LFT, 134 Seiten, 44 Bilder, 15 Tab. 2000. ISBN 3-87525-136-9.

Band 101: Wolfgang Schlögl
Integriertes Simulationsdaten-Management für Maschinenentwicklung und Anlagenplanung
FAPS, 169 Seiten, 101 Bilder, 20 Tab. 2000. ISBN 3-87525-137-7.

Band 102: Christian Hinsel
Ermüdungsbruchversagen hartstoffbeschichteter Werkzeugstähle in der Kaltmassivumformung
LFT, 130 Seiten, 80 Bilder, 14 Tab. 2000. ISBN 3-87525-138-5.

Band 103: Stefan Bobbert
Simulationsgestützte Prozessauslegung für das Innenhochdruck-Umformen von Blechpaaren
LFT, 123 Seiten, 77 Bilder. 2000. ISBN 3-87525-145-8.

Band 104: Harald Rottbauer
Modulares Planungswerkzeug zum Produktionsmanagement in der Elektronikproduktion
FAPS, 166 Seiten, 106 Bilder. 2001.
ISBN 3-87525-139-3.

Band 105: Thomas Hennige
Flexible Formgebung von Blechen durch Laserstrahlumformen
LFT, 119 Seiten, 50 Bilder. 2001.
ISBN 3-87525-140-7.

Band 106: Thomas Menzel
Wissensbasierte Methoden für die rechnergestützte Charakterisierung und Bewertung innovativer Fertigungsprozesse
LFT, 152 Seiten, 71 Bilder. 2001.
ISBN 3-87525-142-3.

Band 107: Thomas Stöckel
Kommunikationstechnische Integration der Prozeßebene in Produktionssysteme durch Middleware-Frameworks
FAPS, 147 Seiten, 65 Bilder, 5 Tab.
2001. ISBN 3-87525-143-1.

Band 108: Frank Pitter
Verfügbarkeitssteigerung von Werkzeugmaschinen durch Einsatz mechatronischer Sensorlösungen
FAPS, 158 Seiten, 131 Bilder, 8 Tab.
2001. ISBN 3-87525-144-X.

Band 109: Markus Korneli
Integration lokaler CAP-Systeme in einen globalen Fertigungsdatenverbund
FAPS, 121 Seiten, 53 Bilder, 11 Tab.
2001. ISBN 3-87525-146-6.

Band 110: Burkhard Müller
Laserstrahljustieren mit Excimer-Lasern - Prozeßparameter und Modelle zur Aktorkonstruktion
LFT, 128 Seiten, 36 Bilder, 9 Tab.
2001. ISBN 3-87525-159-8.

Band 111: Jürgen Göhringer
Integrierte Telediagnose via Internet zum effizienten Service von Produktionssystemen
FAPS, 178 Seiten, 98 Bilder, 5 Tab.
2001. ISBN 3-87525-147-4.

Band 112: Robert Feuerstein
Qualitäts- und kosteneffiziente Integration neuer Bauelementetechnologien in die Flachbaugruppenfertigung
FAPS, 161 Seiten, 99 Bilder, 10 Tab.
2001. ISBN 3-87525-151-2.

Band 113: Marcus Reichenberger
Eigenschaften und Einsatzmöglichkeiten alternativer Elektroniklote in der Oberflächenmontage (SMT)
FAPS, 165 Seiten, 97 Bilder, 18 Tab.
2001. ISBN 3-87525-152-0.

Band 114: Alexander Huber
Justieren vormontierter Systeme mit dem Nd:YAG-Laser unter Einsatz von Aktoren
LFT, 122 Seiten, 58 Bilder, 5 Tab.
2001. ISBN 3-87525-153-9.

Band 115: Sami Krimi
Analyse und Optimierung von Montagesystemen in der Elektronikproduktion
FAPS, 155 Seiten, 88 Bilder, 3 Tab.
2001. ISBN 3-87525-157-1.

Band 116: Marion Merklein
Laserstrahlumformen von Aluminiumwerkstoffen - Beeinflussung der Mikrostruktur und der mechanischen Eigenschaften
LFT, 122 Seiten, 65 Bilder, 15 Tab.
2001. ISBN 3-87525-156-3.

Band 117: Thomas Collisi
Ein informationslogistisches Architekturkonzept zur Akquisition simulationsrelevanter Daten
FAPS, 181 Seiten, 105 Bilder, 7 Tab.
2002. ISBN 3-87525-164-4.

Band 118: Markus Koch
Rationalisierung und ergonomische Optimierung im Innenausbau durch den Einsatz moderner Automatisierungstechnik
FAPS, 176 Seiten, 98 Bilder, 9 Tab.
2002. ISBN 3-87525-165-2.

Band 119: Michael Schmidt
Prozeßregelung für das Laserstrahl-Punktschweißen in der Elektronikproduktion
LFT, 152 Seiten, 71 Bilder, 3 Tab.
2002. ISBN 3-87525-166-0.

Band 120: Nicolas Tiesler
Grundlegende Untersuchungen zum Fließpressen metallischer Kleinstteile
LFT, 126 Seiten, 78 Bilder, 12 Tab.
2002. ISBN 3-87525-175-X.

Band 121: Lars Pursche
Methoden zur technologieorientierten Programmierung für die 3D-Lasermikrobearbeitung
LFT, 111 Seiten, 39 Bilder, 0 Tab.
2002. ISBN 3-87525-183-0.

Band 122: Jan-Oliver Brassel
Prozeßkontrolle beim Laserstrahl-Mikroschweißen
LFT, 148 Seiten, 72 Bilder, 12 Tab.
2002. ISBN 3-87525-181-4.

Band 123: Mark Geisel
Prozeßkontrolle und -steuerung beim Laserstrahlschweißen mit den Methoden der nichtlinearen Dynamik
LFT, 135 Seiten, 46 Bilder, 2 Tab.
2002. ISBN 3-87525-180-6.

Band 124: Gerd Eßer
Laserstrahlunterstützte Erzeugung metallischer Leiterstrukturen auf Thermoplastsubstraten für die MID-Technik
LFT, 148 Seiten, 60 Bilder, 6 Tab.
2002. ISBN 3-87525-171-7.

Band 125: Marc Fleckenstein
Qualität laserstrahl-gefügter Mikroverbindungen elektronischer Kontakte
LFT, 159 Seiten, 77 Bilder, 7 Tab.
2002. ISBN 3-87525-170-9.

Band 126: Stefan Kaufmann
Grundlegende Untersuchungen zum Nd:YAG- Laserstrahlfügen von Silizium für Komponenten der Optoelektronik
LFT, 159 Seiten, 100 Bilder, 6 Tab.
2002. ISBN 3-87525-172-5.

Band 127: Thomas Fröhlich
Simultanes Löten von Anschlußkontakten elektronischer Bauelemente mit Diodenlaserstrahlung
LFT, 143 Seiten, 75 Bilder, 6 Tab.
2002. ISBN 3-87525-186-5.

Band 128: Achim Hofmann
Erweiterung der Formgebungsgrenzen beim Umformen von Aluminiumwerkstoffen durch den Einsatz prozessangepasster Platinen
LFT, 113 Seiten, 58 Bilder, 4 Tab. 2002. ISBN 3-87525-182-2.

Band 129: Ingo Kriebitzsch
3 - D MID Technologie in der Automobilelektronik
FAPS, 129 Seiten, 102 Bilder, 10 Tab. 2002. ISBN 3-87525-169-5.

Band 130: Thomas Pohl
Fertigungsqualität und Umformbarkeit laserstrahlgeschweißter Formplatinen aus Aluminiumlegierungen
LFT, 133 Seiten, 93 Bilder, 12 Tab. 2002. ISBN 3-87525-173-3.

Band 131: Matthias Wenk
Entwicklung eines konfigurierbaren Steuerungssystems für die flexible Sensorführung von Industrierobotern
FAPS, 167 Seiten, 85 Bilder, 1 Tab. 2002. ISBN 3-87525-174-1.

Band 132: Matthias Negendanck
Neue Sensorik und Aktorik für Bearbeitungsköpfe zum Laserstrahlschweißen
LFT, 116 Seiten, 60 Bilder, 14 Tab. 2002. ISBN 3-87525-184-9.

Band 133: Oliver Kreis
Integrierte Fertigung - Verfahrensintegration durch Innenhochdruck-Umformen, Trennen und Laserstrahlschweißen in einem Werkzeug sowie ihre tele- und multimediale Präsentation
LFT, 167 Seiten, 90 Bilder, 43 Tab. 2002. ISBN 3-87525-176-8.

Band 134: Stefan Trautner
Technische Umsetzung produktbezogener Instrumente der Umweltpolitik bei Elektro- und Elektronikgeräten
FAPS, 179 Seiten, 92 Bilder, 11 Tab. 2002. ISBN 3-87525-177-6.

Band 135: Roland Meier
Strategien für einen produktorientierten Einsatz räumlicher spritzgegossener Schaltungsträger (3-D MID)
FAPS, 155 Seiten, 88 Bilder, 14 Tab. 2002. ISBN 3-87525-178-4.

Band 136: Jürgen Wunderlich
Kostensimulation - Simulationsbasierte Wirtschaftlichkeitsregelung komplexer Produktionssysteme
FAPS, 202 Seiten, 119 Bilder, 17 Tab. 2002. ISBN 3-87525-179-2.

Band 137: Stefan Novotny
Innenhochdruck-Umformen von Blechen aus Aluminium- und Magnesiumlegierungen bei erhöhter Temperatur
LFT, 132 Seiten, 82 Bilder, 6 Tab. 2002. ISBN 3-87525-185-7.

Band 138: Andreas Licha
Flexible Montageautomatisierung zur Komplettmontage flächenhafter Produktstrukturen durch kooperierende Industrieroboter
FAPS, 158 Seiten, 87 Bilder, 8 Tab. 2003. ISBN 3-87525-189-X.

Band 139: Michael Eisenbarth
Beitrag zur Optimierung der Aufbau- und Verbindungstechnik für mechatronische Baugruppen
FAPS, 207 Seiten, 141 Bilder, 9 Tab. 2003. ISBN 3-87525-190-3.

Band 140: Frank Christoph
Durchgängige simulationsgestützte Planung von Fertigungseinrichtungen der Elektronikproduktion
FAPS, 187 Seiten, 107 Bilder, 9 Tab. 2003. ISBN 3-87525-191-1.

Band 141: Hinnerk Hagenah
Simulationsbasierte Bestimmung der zu erwartenden Maßhaltigkeit für das Blechbiegen
LFT, 131 Seiten, 36 Bilder, 26 Tab. 2003. ISBN 3-87525-192-X.

Band 142: Ralf Eckstein
Scherschneiden und Biegen metallischer Kleinstteile - Materialeinfluss und Materialverhalten
LFT, 148 Seiten, 71 Bilder, 19 Tab. 2003. ISBN 3-87525-193-8.

Band 143: Frank H. Meyer-Pittroff
Excimerlaserstrahlbiegen dünner metallischer Folien mit homogener Lichtlinie
LFT, 138 Seiten, 60 Bilder, 16 Tab. 2003. ISBN 3-87525-196-2.

Band 144: Andreas Kach
Rechnergestützte Anpassung von Laserstrahlschneidbahnen an Bauteilabweichungen
LFT, 139 Seiten, 69 Bilder, 11 Tab. 2004. ISBN 3-87525-197-0.

Band 145: Stefan Hierl
System- und Prozeßtechnik für das simultane Löten mit Diodenlaserstrahlung von elektronischen Bauelementen
LFT, 124 Seiten, 66 Bilder, 4 Tab. 2004. ISBN 3-87525-198-9.

Band 146: Thomas Neudecker
Tribologische Eigenschaften keramischer Blechumformwerkzeuge-Einfluss einer Oberflächenendbearbeitung mittels Excimerlaserstrahlung
LFT, 166 Seiten, 75 Bilder, 26 Tab. 2004. ISBN 3-87525-200-4.

Band 147: Ulrich Wenger
Prozessoptimierung in der Wickeltechnik durch innovative maschinenbauliche und regelungstechnische Ansätze
FAPS, 132 Seiten, 88 Bilder, 0 Tab. 2004. ISBN 3-87525-203-9.

Band 148: Stefan Slama
Effizienzsteigerung in der Montage durch marktorientierte Montagestrukturen und erweiterte Mitarbeiterkompetenz
FAPS, 188 Seiten, 125 Bilder, 0 Tab. 2004. ISBN 3-87525-204-7.

Band 149: Thomas Wurm
Laserstrahljustieren mittels Aktoren-Entwicklung von Konzepten und Methoden für die rechnerunterstützte Modellierung und Optimierung von komplexen Aktorsystemen in der Mikrotechnik
LFT, 122 Seiten, 51 Bilder, 9 Tab. 2004. ISBN 3-87525-206-3.

Band 150: Martino Celeghini
Wirkmedienbasierte Blechumformung: Grundlagenuntersuchungen zum Einfluss von Werkstoff und Bauteilgeometrie
LFT, 146 Seiten, 77 Bilder, 6 Tab.
2004. ISBN 3-87525-207-1.

Band 151: Ralph Hohenstein
Entwurf hochdynamischer Sensor- und Regelsysteme für die adaptiveLaserbearbeitung
LFT, 282 Seiten, 63 Bilder, 16 Tab.
2004. ISBN 3-87525-210-1.

Band 152: Angelika Hutterer
Entwicklung prozessüberwachender Regelkreise für flexible Formgebungsprozesse
LFT, 149 Seiten, 57 Bilder, 2 Tab.
2005. ISBN 3-87525-212-8.

Band 153: Emil Egerer
Massivumformen metallischer Kleinstteile bei erhöhter Prozesstemperatur
LFT, 158 Seiten, 87 Bilder, 10 Tab.
2005. ISBN 3-87525-213-6.

Band 154: Rüdiger Holzmann
Strategien zur nachhaltigen Optimierung von Qualität und Zuverlässigkeit in der Fertigung hochintegrierter Flachbaugruppen
FAPS, 186 Seiten, 99 Bilder, 19 Tab.
2005. ISBN 3-87525-217-9.

Band 155: Marco Nock
Biegeumformen mit Elastomerwerkzeugen Modellierung, Prozessauslegung und Abgrenzung des Verfahrens am Beispiel des Rohrbiegens
LFT, 164 Seiten, 85 Bilder, 13 Tab.
2005. ISBN 3-87525-218-7.

Band 156: Frank Niebling
Qualifizierung einer Prozesskette zum Laserstrahlsintern metallischer Bauteile
LFT, 148 Seiten, 89 Bilder, 3 Tab.
2005. ISBN 3-87525-219-5.

Band 157: Markus Meiler
Großserientauglichkeit trockenschmierstoffbeschichteter Aluminiumbleche im Presswerk Grundlegende Untersuchungen zur Tribologie, zum Umformverhalten und Bauteilversuche
LFT, 104 Seiten, 57 Bilder, 21 Tab.
2005. ISBN 3-87525-221-7.

Band 158: Agus Sutanto
Solution Approaches for Planning of Assembly Systems in Three-Dimensional Virtual Environments
FAPS, 169 Seiten, 98 Bilder, 3 Tab.
2005. ISBN 3-87525-220-9.

Band 159: Matthias Boiger
Hochleistungssysteme für die Fertigung elektronischer Baugruppen auf der Basis flexibler Schaltungsträger
FAPS, 175 Seiten, 111 Bilder, 8 Tab.
2005. ISBN 3-87525-222-5.

Band 160: Matthias Pitz
Laserunterstütztes Biegen höchstfester Mehrphasenstähle
LFT, 120 Seiten, 73 Bilder, 11 Tab.
2005. ISBN 3-87525-223-3.

Band 161: Meik Vahl
Beitrag zur gezielten Beeinflussung des Werkstoffflusses beim Innenhochdruck-Umformen von Blechen
LFT, 165 Seiten, 94 Bilder, 15 Tab.
2005. ISBN 3-87525-224-1.

Band 162: Peter K. Kraus
Plattformstrategien - Realisierung einer varianz- und kostenoptimierten Wertschöpfung
FAPS, 181 Seiten, 95 Bilder, 0 Tab.
2005. ISBN 3-87525-226-8.

Band 163: Adrienn Cser
Laserstrahlschmelzabtrag - Prozessanalyse und -modellierung
LFT, 146 Seiten, 79 Bilder, 3 Tab.
2005. ISBN 3-87525-227-6.

Band 164: Markus C. Hahn
Grundlegende Untersuchungen zur Herstellung von Leichtbauverbundstrukturen mit Aluminiumschaumkern
LFT, 143 Seiten, 60 Bilder, 16 Tab.
2005. ISBN 3-87525-228-4.

Band 165: Gordana Michos
Mechatronische Ansätze zur Optimierung von Vorschubachsen
FAPS, 146 Seiten, 87 Bilder, 17 Tab.
2005. ISBN 3-87525-230-6.

Band 166: Markus Stark
Auslegung und Fertigung hochpräziser Faser-Kollimator-Arrays
LFT, 158 Seiten, 115 Bilder, 11 Tab.
2005. ISBN 3-87525-231-4.

Band 167: Yurong Zhou
Kollaboratives Engineering Management in der integrierten virtuellen Entwicklung der Anlagen für die Elektronikproduktion
FAPS, 156 Seiten, 84 Bilder, 6 Tab.
2005. ISBN 3-87525-232-2.

Band 168: Werner Enser
Neue Formen permanenter und lösbarer elektrischer Kontaktierungen für mechatronische Baugruppen
FAPS, 190 Seiten, 112 Bilder, 5 Tab.
2005. ISBN 3-87525-233-0.

Band 169: Katrin Melzer
Integrierte Produktpolitik bei elektrischen und elektronischen Geräten zur Optimierung des Product-Life-Cycle
FAPS, 155 Seiten, 91 Bilder, 17 Tab.
2005. ISBN 3-87525-234-9.

Band 170: Alexander Putz
Grundlegende Untersuchungen zur Erfassung der realen Vorspannung von armierten Kaltfließpresswerkzeugen mittels Ultraschall
LFT, 137 Seiten, 71 Bilder, 15 Tab.
2006. ISBN 3-87525-237-3.

Band 171: Martin Prechtl
Automatisiertes Schichtverfahren für metallische Folien - System- und Prozesstechnik
LFT, 154 Seiten, 45 Bilder, 7 Tab.
2006. ISBN 3-87525-238-1.

Band 172: Markus Meidert
Beitrag zur deterministischen Lebensdauerabschätzung von Werkzeugen der Kaltmassivumformung
LFT, 131 Seiten, 78 Bilder, 9 Tab.
2006. ISBN 3-87525-239-X.

Band 173: Bernd Müller
Robuste, automatisierte Montagesysteme durch adaptive Prozessführung und montageübergreifende Fehlerprävention am Beispiel flächiger Leichtbauteile
FAPS, 147 Seiten, 77 Bilder, 0 Tab.
2006. ISBN 3-87525-240-3.

Band 174: Alexander Hofmann
Hybrides Laserdurchstrahlschweißen von Kunststoffen
LFT, 136 Seiten, 72 Bilder, 4 Tab.
2006. ISBN 978-3-87525-243-9.

Band 175: Peter Wölflick
Innovative Substrate und Prozesse mit feinsten Strukturen für bleifreie Mechatronik-Anwendungen
FAPS, 177 Seiten, 148 Bilder, 24 Tab. 2006.
ISBN 978-3-87525-246-0.

Band 176: Attila Komlodi
Detection and Prevention of Hot Cracks during Laser Welding of Aluminium Alloys Using Advanced Simulation Methods
LFT, 155 Seiten, 89 Bilder, 14 Tab. 2006. ISBN 978-3-87525-248-4.

Band 177: Uwe Popp
Grundlegende Untersuchungen zum Laserstrahlstrukturieren von Kaltmassivumformwerkzeugen
LFT, 140 Seiten, 67 Bilder, 16 Tab. 2006. ISBN 978-3-87525-249-1.

Band 178: Veit Rückel
Rechnergestützte Ablaufplanung und Bahngenerierung Für kooperierende Industrieroboter
FAPS, 148 Seiten, 75 Bilder, 7 Tab. 2006. ISBN 978-3-87525-250-7.

Band 179: Manfred Dirscherl
Nicht-thermische Mikrojustiertechnik mittels ultrakurzer Laserpulse
LFT, 154 Seiten, 69 Bilder, 10 Tab. 2007. ISBN 978-3-87525-251-4.

Band 180: Yong Zhuo
Entwurf eines rechnergestützten integrierten Systems für Konstruktion und Fertigungsplanung räumlicher spritzgegossener Schaltungsträger (3D-MID)
FAPS, 181 Seiten, 95 Bilder, 5 Tab. 2007. ISBN 978-3-87525-253-8.

Band 181: Stefan Lang
Durchgängige Mitarbeiterinformation zur Steigerung von Effizienz und Prozesssicherheit in der Produktion
FAPS, 172 Seiten, 93 Bilder. 2007.
ISBN 978-3-87525-257-6.

Band 182: Hans-Joachim Krauß
Laserstrahlinduzierte Pyrolyse präkeramischer Polymere
LFT, 171 Seiten, 100 Bilder. 2007.
ISBN 978-3-87525-258-3.

Band 183: Stefan Junker
Technologien und Systemlösungen für die flexibel automatisierte Bestückung permanent erregter Läufer mit oberflächenmontierten Dauermagneten
FAPS, 173 Seiten, 75 Bilder. 2007.
ISBN 978-3-87525-259-0.

Band 184: Rainer Kohlbauer
Wissensbasierte Methoden für die simulationsgestützte Auslegung wirkmedienbasierter Blechumformprozesse
LFT, 135 Seiten, 50 Bilder. 2007.
ISBN 978-3-87525-260-6.

Band 185: Klaus Lamprecht
Wirkmedienbasierte Umformung tiefgezogener Vorformen unter besonderer Berücksichtigung maßgeschneiderter Halbzeuge
LFT, 137 Seiten, 81 Bilder. 2007.
ISBN 978-3-87525-265-1.

Band 186: Bernd Zolleiß
Optimierte Prozesse und Systeme für die Bestückung mechatronischerBaugruppen
FAPS, 180 Seiten, 117 Bilder. 2007.
ISBN 978-3-87525-266-8.

Band 187: Michael Kerausch
Simulationsgestützte Prozessauslegung für das Umformen lokal wärmebehandelter Aluminiumplatinen
LFT, 146 Seiten, 76 Bilder, 7 Tab. 2007. ISBN 978-3-87525-267-5.

Band 188: Matthias Weber
Unterstützung der Wandlungsfähigkeit von Produktionsanlagen durch innovative Softwaresysteme
FAPS, 183 Seiten, 122 Bilder, 3 Tab. 2007. ISBN 978-3-87525-269-9.

Band 189: Thomas Frick
Untersuchung der prozessbestimmenden Strahl-Stoff-Wechselwirkungen beim Laserstrahlschweißen von Kunststoffen
LFT, 104 Seiten, 62 Bilder, 8 Tab. 2007. ISBN 978-3-87525-268-2.

Band 190: Joachim Hecht
Werkstoffcharakterisierung und Prozessauslegung für die wirkmedienbasierte Doppelblech-Umformung von Magnesiumlegierungen
LFT, 107 Seiten, 91 Bilder, 2 Tab. 2007. ISBN 978-3-87525-270-5.

Band 191: Ralf Völkl
Stochastische Simulation zur Werkzeuglebensdaueroptimierung und Präzisionsfertigung in der Kaltmassivumformung
LFT, 178 Seiten, 75 Bilder, 12 Tab. 2008. ISBN 978-3-87525-272-9.

Band 192: Massimo Tolazzi
Innenhochdruck-Umformen verstärkter Blech-Rahmenstrukturen
LFT, 164 Seiten, 85 Bilder, 7 Tab. 2008. ISBN 978-3-87525-273-6.

Band 193: Cornelia Hoff
Untersuchung der Prozesseinflussgrößen beim Presshärten des höchstfesten Vergütungsstahls 22MnB5
LFT, 133 Seiten, 92 Bilder, 5 Tab. 2008. ISBN 978-3-87525-275-0.

Band 194: Christian Alvarez
Simulationsgestützte Methoden zur effizienten Gestaltung von Lötprozessen in der Elektronikproduktion
FAPS, 149 Seiten, 86 Bilder, 8 Tab. 2008. ISBN 978-3-87525-277-4.

Band 195: Andreas Kunze
Automatisierte Montage von makromechatronischen Modulen zur flexiblen Integration in hybride Pkw-Bordnetzsysteme
FAPS, 160 Seiten, 90 Bilder, 14 Tab. 2008.
ISBN 978-3-87525-278-1.

Band 196: Wolfgang Hußnätter
Grundlegende Untersuchungen zur experimentellen Ermittlung und zur Modellierung von Fließortkurven bei erhöhten Temperaturen
LFT, 152 Seiten, 73 Bilder, 21 Tab. 2008. ISBN 978-3-87525-279-8.

Band 197: Thomas Bigl
Entwicklung, angepasste Herstellungsverfahren und erweiterte Qualitätssicherung von einsatzgerechten elektronischen Baugruppen
FAPS, 175 Seiten, 107 Bilder, 14 Tab. 2008.
ISBN 978-3-87525-280-4.

Band 198: Stephan Roth
Grundlegende Untersuchungen zum Excimerlaserstrahl-Abtragen unter Flüssigkeitsfilmen
LFT, 113 Seiten, 47 Bilder, 14 Tab. 2008. ISBN 978-3-87525-281-1.

Band 199: Artur Giera
Prozesstechnische Untersuchungen zum Rührreibschweißen metallischer Werkstoffe
LFT, 179 Seiten, 104 Bilder, 36 Tab. 2008. ISBN 978-3-87525-282-8.

Band 200: Jürgen Lechler
Beschreibung und Modellierung des Werkstoffverhaltens von presshärtbaren Bor-Manganstählen
LFT, 154 Seiten, 75 Bilder, 12 Tab. 2009. ISBN 978-3-87525-286-6.

Band 201: Andreas Blankl
Untersuchungen zur Erhöhung der Prozessrobustheit bei der Innenhochdruck-Umformung von flächigen Halbzeugen mit vor- bzw. nachgeschalteten Laserstrahlfügeoperationen
LFT, 120 Seiten, 68 Bilder, 9 Tab. 2009. ISBN 978-3-87525-287-3.

Band 202: Andreas Schaller
Modellierung eines nachfrageorientierten Produktionskonzeptes für mobile Telekommunikationsgeräte
FAPS, 120 Seiten, 79 Bilder, 0 Tab. 2009. ISBN 978-3-87525-289-7.

Band 203: Claudius Schimpf
Optimierung von Zuverlässigkeitsuntersuchungen, Prüfabläufen und Nacharbeitsprozessen in der Elektronikproduktion
FAPS, 162 Seiten, 90 Bilder, 14 Tab. 2009.
ISBN 978-3-87525-290-3.

Band 204: Simon Dietrich
Sensoriken zur Schwerpunktslagebestimmung der optischen Prozessemissionen beim Laserstrahltiefschweißen
LFT, 138 Seiten, 70 Bilder, 5 Tab. 2009. ISBN 978-3-87525-292-7.

Band 205: Wolfgang Wolf
Entwicklung eines agentenbasierten Steuerungssystems zur Materialflussorganisation im wandelbaren Produktionsumfeld
FAPS, 167 Seiten, 98 Bilder. 2009. ISBN 978-3-87525-293-4.

Band 206: Steffen Polster
Laserdurchstrahlschweißen transparenter Polymerbauteile
LFT, 160 Seiten, 92 Bilder, 13 Tab. 2009. ISBN 978-3-87525-294-1.

Band 207: Stephan Manuel Dörfler
Rührreibschweißen von walzplattiertem Halbzeug und Aluminiumblech zur Herstellung flächiger Aluminiumschaum-Sandwich-Verbundstrukturen
LFT, 190 Seiten, 98 Bilder, 5 Tab. 2009. ISBN 978-3-87525-295-8.

Band 208: Uwe Vogt
Seriennahe Auslegung von Aluminium Tailored Heat Treated Blanks
LFT, 151 Seiten, 68 Bilder, 26 Tab. 2009. ISBN 978-3-87525-296-5.

Band 209: Till Laumann
Qualitative und quantitative Bewertung der Crashtauglichkeit von höchstfesten Stählen
LFT, 117 Seiten, 69 Bilder, 7 Tab. 2009. ISBN 978-3-87525-299-6.

Band 210: Alexander Diehl
Größeneffekte bei Biegeprozessen- Entwicklung einer Methodik zur Identifikation und Quantifizierung
LFT, 180 Seiten, 92 Bilder, 12 Tab. 2010. ISBN 978-3-87525-302-3.

Band 211: Detlev Staud
Effiziente Prozesskettenauslegung für das Umformen lokal wärmebehandelter und geschweißter Aluminiumbleche
LFT, 164 Seiten, 72 Bilder, 12 Tab. 2010. ISBN 978-3-87525-303-0.

Band 212: Jens Ackermann
Prozesssicherung beim Laserdurchstrahlschweißen thermoplastischer Kunststoffe
LPT, 129 Seiten, 74 Bilder, 13 Tab. 2010. ISBN 978-3-87525-305-4.

Band 213: Stephan Weidel
Grundlegende Untersuchungen zum Kontaktzustand zwischen Werkstück und Werkzeug bei umformtechnischen Prozessen unter tribologischen Gesichtspunkten
LFT, 144 Seiten, 67 Bilder, 11 Tab. 2010. ISBN 978-3-87525-307-8.

Band 214: Stefan Geißdörfer
Entwicklung eines mesoskopischen Modells zur Abbildung von Größeneffekten in der Kaltmassivumformung mit Methoden der FE-Simulation
LFT, 133 Seiten, 83 Bilder, 11 Tab. 2010. ISBN 978-3-87525-308-5.

Band 215: Christian Matzner
Konzeption produktspezifischer Lösungen zur Robustheitssteigerung elektronischer Systeme gegen die Einwirkung von Betauung im Automobil
FAPS, 165 Seiten, 93 Bilder, 14 Tab. 2010. ISBN 978-3-87525-309-2.

Band 216: Florian Schüßler
Verbindungs- und Systemtechnik für thermisch hochbeanspruchte und miniaturisierte elektronische Baugruppen
FAPS, 184 Seiten, 93 Bilder, 18 Tab. 2010.
ISBN 978-3-87525-310-8.

Band 217: Massimo Cojutti
Strategien zur Erweiterung der Prozessgrenzen bei der Innhochdruck-Umformung von Rohren und Blechpaaren
LFT, 125 Seiten, 56 Bilder, 9 Tab. 2010. ISBN 978-3-87525-312-2.

Band 218: Raoul Plettke
Mehrkriterielle Optimierung komplexer Aktorsysteme für das Laserstrahljustieren
LFT, 152 Seiten, 25 Bilder, 3 Tab. 2010. ISBN 978-3-87525-315-3.

Band 219: Andreas Dobroschke
Flexible Automatisierungslösungen für die Fertigung wickeltechnischer Produkte
FAPS, 184 Seiten, 109 Bilder, 18 Tab. 2011.
ISBN 978-3-87525-317-7.

Band 220: Azhar Zam
Optical Tissue Differentiation for Sensor-Controlled Tissue-Specific Laser Surgery
LPT, 99 Seiten, 45 Bilder, 8 Tab. 2011. ISBN 978-3-87525-318-4.

Band 221: Michael Rösch
Potenziale und Strategien zur Optimierung des Schablonendruckprozesses in der Elektronikproduktion
FAPS, 192 Seiten, 127 Bilder, 19 Tab. 2011.
ISBN 978-3-87525-319-1.

Band 222: Thomas Rechtenwald
Quasi-isothermes Laserstrahlsintern von Hochtemperatur-Thermoplasten - Eine Betrachtung werkstoff-prozessspezifischer Aspekte am Beispiel PEEK
LPT, 150 Seiten, 62 Bilder, 8 Tab. 2011. ISBN 978-3-87525-320-7.

Band 223: Daniel Craiovan
Prozesse und Systemlösungen für die SMT-Montage optischer Bauelemente auf Substrate mit integrierten Lichtwellenleitern
FAPS, 165 Seiten, 85 Bilder, 8 Tab. 2011. ISBN 978-3-87525-324-5.

Band 224: Kay Wagner
Beanspruchungsangepasste Kaltmassivumformwerkzeuge durch lokal optimierte Werkzeugoberflächen
LFT, 147 Seiten, 103 Bilder, 17 Tab. 2011. ISBN 978-3-87525-325-2.

Band 225: Martin Brandhuber
Verbesserung der Prognosegüte des Versagens von Punktschweißverbindungen bei höchstfesten Stahlgüten
LFT, 155 Seiten, 91 Bilder, 19 Tab. 2011. ISBN 978-3-87525-327-6.

Band 226: Peter Sebastian Feuser
Ein Ansatz zur Herstellung von pressgehärteten Karosseriekomponenten mit maßgeschneiderten mechanischen Eigenschaften: Temperierte Umformwerkzeuge. Prozessfenster, Prozesssimuation und funktionale Untersuchung
LFT, 195 Seiten, 97 Bilder, 60 Tab. 2012. ISBN 978-3-87525-328-3.

Band 227: Murat Arbak
Material Adapted Design of Cold Forging Tools Exemplified by Powder Metallurgical Tool Steels and Ceramics
LFT, 109 Seiten, 56 Bilder, 8 Tab. 2012. ISBN 978-3-87525-330-6.

Band 228: Indra Pitz
Beschleunigte Simulation des Laserstrahlumformens von Aluminiumblechen
LPT, 137 Seiten, 45 Bilder, 27 Tab. 2012. ISBN 978-3-87525-333-7.

Band 229: Alexander Grimm
Prozessanalyse und -überwachung des Laserstrahlhartlötens mittels optischer Sensorik
LPT, 125 Seiten, 61 Bilder, 5 Tab. 2012. ISBN 978-3-87525-334-4.

Band 230: Markus Kaupper
Biegen von höhenfesten Stahlblechwerkstoffen - Umformverhalten und Grenzen der Biegbarkeit
LFT, 160 Seiten, 57 Bilder, 10 Tab. 2012. ISBN 978-3-87525-339-9.

Band 231: Thomas Kroiß
Modellbasierte Prozessauslegung für die Kaltmassivumformung unter Brücksichtigung der Werkzeug- und Pressenauffederung
LFT, 169 Seiten, 50 Bilder, 19 Tab. 2012. ISBN 978-3-87525-341-2.

Band 232: Christian Goth
Analyse und Optimierung der Entwicklung und Zuverlässigkeit räumlicher Schaltungsträger (3D-MID)
FAPS, 176 Seiten, 102 Bilder, 22 Tab. 2012.
ISBN 978-3-87525-340-5.

Band 233: Christian Ziegler
Ganzheitliche Automatisierung mechatronischer Systeme in der Medizin am Beispiel Strahlentherapie
FAPS, 170 Seiten, 71 Bilder, 19 Tab. 2012. ISBN 978-3-87525-342-9.

Band 234: Florian Albert
Automatisiertes Laserstrahllöten und -reparaturlöten elektronischer Baugruppen
LPT, 127 Seiten, 78 Bilder, 11 Tab. 2012. ISBN 978-3-87525-344-3.

Band 235: Thomas Stöhr
Analyse und Beschreibung des mechanischen Werkstoffverhaltens von presshärtbaren Bor-Manganstählen
LFT, 118 Seiten, 74 Bilder, 18 Tab. 2013. ISBN 978-3-87525-346-7.

Band 236: Christian Kägeler
Prozessdynamik beim Laserstrahlschweißen verzinkter Stahlbleche im Überlappstoß
LPT, 145 Seiten, 80 Bilder, 3 Tab. 2013. ISBN 978-3-87525-347-4.

Band 237: Andreas Sulzberger
Seriennahe Auslegung der Prozesskette zur wärmeunterstützten Umformung von Aluminiumblechwerkstoffen
LFT, 153 Seiten, 87 Bilder, 17 Tab. 2013. ISBN 978-3-87525-349-8.

Band 238: Simon Opel
Herstellung prozessangepasster Halbzeuge mit variabler Blechdicke durch die Anwendung von Verfahren der Blechmassivumformung
LFT, 165 Seiten, 108 Bilder, 27 Tab. 2013. ISBN 978-3-87525-350-4.

Band 239: Rajesh Kanawade
In-vivo Monitoring of Epithelium Vessel and Capillary Density for the Application of Detection of Clinical Shock and Early Signs of Cancer Development
LPT, 124 Seiten, 58 Bilder, 15 Tab. 2013. ISBN 978-3-87525-351-1.

Band 240: Stephan Busse
Entwicklung und Qualifizierung eines Schneidclinchverfahrens
LFT, 119 Seiten, 86 Bilder, 20 Tab. 2013. ISBN 978-3-87525-352-8.

Band 241: Karl-Heinz Leitz
Mikro- und Nanostrukturierung mit kurz und ultrakurz gepulster Laserstrahlung
LPT, 154 Seiten, 71 Bilder, 9 Tab. 2013. ISBN 978-3-87525-355-9.

Band 242: Markus Michl
Webbasierte Ansätze zur ganzheitlichen technischen Diagnose
FAPS, 182 Seiten, 62 Bilder, 20 Tab. 2013.
ISBN 978-3-87525-356-6.

Band 243: Vera Sturm
Einfluss von Chargenschwankungen auf die Verarbeitungsgrenzen von Stahlwerkstoffen
LFT, 113 Seiten, 58 Bilder, 9 Tab. 2013. ISBN 978-3-87525-357-3.

Band 244: Christian Neudel
Mikrostrukturelle und mechanisch-technologische Eigenschaften widerstandspunktgeschweißter Aluminium-Stahl-Verbindungen für den Fahrzeugbau
LFT, 178 Seiten, 171 Bilder, 31 Tab. 2014. ISBN 978-3-87525-358-0.

Band 245: Anja Neumann
Konzept zur Beherrschung der Prozessschwankungen im Presswerk
LFT, 162 Seiten, 68 Bilder, 15 Tab. 2014. ISBN 978-3-87525-360-3.

Band 246: Ulf-Hermann Quentin
Laserbasierte Nanostrukturierung mit optisch positionierten Mikrolinsen
LPT, 137 Seiten, 89 Bilder, 6 Tab. 2014. ISBN 978-3-87525-361-0.

Band 247: Erik Lamprecht
Der Einfluss der Fertigungsverfahren auf die Wirbelstromverluste von Stator-Einzelzahnblechpaketen für den Einsatz in Hybrid- und Elektrofahrzeugen
FAPS, 148 Seiten, 138 Bilder, 4 Tab. 2014. ISBN 978-3-87525-362-7.

Band 248: Sebastian Rösel
Wirkmedienbasierte Umformung von Blechhalbzeugen unter Anwendung magnetorheologischer Flüssigkeiten als kombiniertes Wirk- und Dichtmedium
LFT, 148 Seiten, 61 Bilder, 12 Tab. 2014. ISBN 978-3-87525-363-4.

Band 249: Paul Hippchen
Simulative Prognose der Geometrie indirekt pressgehärteter Karosseriebauteile für die industrielle Anwendung
LFT, 163 Seiten, 89 Bilder, 12 Tab. 2014. ISBN 978-3-87525-364-1.

Band 250: Martin Zubeil
Versagensprognose bei der Prozesssimulation von Biegeumform- und Falzverfahren
LFT, 171 Seiten, 90 Bilder, 5 Tab. 2014. ISBN 978-3-87525-365-8.

Band 251: Alexander Kühl
Flexible Automatisierung der Statorenmontage mit Hilfe einer universellen ambidexteren Kinematik
FAPS, 142 Seiten, 60 Bilder, 26 Tab. 2014.
ISBN 978-3-87525-367-2.

Band 252: Thomas Albrecht
Optimierte Fertigungstechnologien für Rotoren getriebeintegrierter PM-Synchronmotoren von Hybridfahrzeugen
FAPS, 198 Seiten, 130 Bilder, 38 Tab. 2014.
ISBN 978-3-87525-368-9.

Band 253: Florian Risch
Planning and Production Concepts for Contactless Power Transfer Systems for Electric Vehicles
FAPS, 185 Seiten, 125 Bilder, 13 Tab. 2014.
ISBN 978-3-87525-369-6.

Band 254: Markus Weigl
Laserstrahlschweißen von Mischverbindungen aus austenitischen und ferritischen korrosionsbeständigen Stahlwerkstoffen
LPT, 184 Seiten, 110 Bilder, 6 Tab. 2014. ISBN 978-3-87525-370-2.

Band 255: Johannes Noneder
Beanspruchungserfassung für die Validierung von FE-Modellen zur Auslegung von Massivumformwerkzeugen
LFT, 161 Seiten, 65 Bilder, 14 Tab. 2014. ISBN 978-3-87525-371-9.

Band 256: Andreas Reinhardt
Ressourceneffiziente Prozess- und Produktionstechnologie für flexible Schaltungsträger
FAPS, 123 Seiten, 69 Bilder, 19 Tab. 2014. ISBN 978-3-87525-373-3.

Band 257: Tobias Schmuck
Ein Beitrag zur effizienten Gestaltung globaler Produktions- und Logistiknetzwerke mittels Simulation
FAPS, 151 Seiten, 74 Bilder. 2014.
ISBN 978-3-87525-374-0.

Band 258: Bernd Eichenhüller
Untersuchungen der Effekte und Wechselwirkungen charakteristischer Einflussgrößen auf das Umformverhalten bei Mikroumformprozessen
LFT, 127 Seiten, 29 Bilder, 9 Tab. 2014. ISBN 978-3-87525-375-7.

Band 259: Felix Lütteke
Vielseitiges autonomes Transportsystem basierend auf Weltmodellerstellung mittels Datenfusion von Deckenkameras und Fahrzeugsensoren
FAPS, 152 Seiten, 54 Bilder, 20 Tab. 2014.
ISBN 978-3-87525-376-4.

Band 260: Martin Grüner
Hochdruck-Blechumformung mit formlos festen Stoffen als Wirkmedium
LFT, 144 Seiten, 66 Bilder, 29 Tab. 2014. ISBN 978-3-87525-379-5.

Band 261: Christian Brock
Analyse und Regelung des Laserstrahltiefschweißprozesses durch Detektion der Metalldampffackelposition
LPT, 126 Seiten, 65 Bilder, 3 Tab. 2015. ISBN 978-3-87525-380-1.

Band 262: Peter Vatter
Sensitivitätsanalyse des 3-Rollen-Schubbiegens auf Basis der Finite Elemente Methode
LFT, 145 Seiten, 57 Bilder, 26 Tab. 2015. ISBN 978-3-87525-381-8.

Band 263: Florian Klämpfl
Planung von Laserbestrahlungen durch simulationsbasierte Optimierung
LPT, 169 Seiten, 78 Bilder, 32 Tab. 2015. ISBN 978-3-87525-384-9.

Band 264: Matthias Domke
Transiente physikalische Mechanismen bei der Laserablation von dünnen Metallschichten
LPT, 133 Seiten, 43 Bilder, 3 Tab.
2015. ISBN 978-3-87525-385-6.

Band 265: Johannes Götz
Community-basierte Optimierung des Anlagenengineerings
FAPS, 177 Seiten, 80 Bilder, 30 Tab.
2015.
ISBN 978-3-87525-386-3.

Band 266: Hung Nguyen
Qualifizierung des Potentials von Verfestigungseffekten zur Erweiterung des Umformvermögens aushärtbarer Aluminiumlegierungen
LFT, 137 Seiten, 57 Bilder, 16 Tab.
2015. ISBN 978-3-87525-387-0.

Band 267: Andreas Kuppert
Erweiterung und Verbesserung von Versuchs- und Auswertetechniken für die Bestimmung von Grenzformänderungskurven
LFT, 138 Seiten, 82 Bilder, 2 Tab.
2015. ISBN 978-3-87525-388-7.

Band 268: Kathleen Klaus
Erstellung eines Werkstofforientierten Fertigungsprozessfensters zur Steigerung des Formgebungsvermögens von Alumi-niumlegierungen unter Anwendung einer zwischengeschalteten Wärmebehandlung
LFT, 154 Seiten, 70 Bilder, 8 Tab.
2015. ISBN 978-3-87525-391-7.

Band 269: Thomas Svec
Untersuchungen zur Herstellung von funktionsoptimierten Bauteilen im partiellen Presshärtprozess mittels lokal unterschiedlich temperierter Werkzeuge
LFT, 166 Seiten, 87 Bilder, 15 Tab.
2015. ISBN 978-3-87525-392-4.

Band 270: Tobias Schrader
Grundlegende Untersuchungen zur Verschleißcharakterisierung beschichteter Kaltmassivumformwerkzeuge
LFT, 164 Seiten, 55 Bilder, 11 Tab.
2015. ISBN 978-3-87525-393-1.

Band 271: Matthäus Brela
Untersuchung von Magnetfeld-Messmethoden zur ganzheitlichen Wertschöpfungsoptimierung und Fehlerdetektion an magnetischen Aktoren
FAPS, 170 Seiten, 97 Bilder, 4 Tab.
2015. ISBN 978-3-87525-394-8.

Band 272: Michael Wieland
Entwicklung einer Methode zur Prognose adhäsiven Verschleißes an Werkzeugen für das direkte Presshärten
LFT, 156 Seiten, 84 Bilder, 9 Tab.
2015. ISBN 978-3-87525-395-5.

Band 273: René Schramm
Strukturierte additive Metallisierung durch kaltaktives Atmosphärendruckplasma
FAPS, 136 Seiten, 62 Bilder, 15 Tab.
2015. ISBN 978-3-87525-396-2.

Band 274: Michael Lechner
Herstellung beanspruchungsangepasster Aluminiumblechhalbzeuge durch eine maßgeschneiderte Variation der Abkühlgeschwindigkeit nach Lösungsglühen
LFT, 136 Seiten, 62 Bilder, 15 Tab.
2015. ISBN 978-3-87525-397-9.

Band 275: Kolja Andreas
Einfluss der Oberflächenbeschaffenheit auf das Werkzeugeinsatzverhalten beim Kaltfließpressen
LFT, 169 Seiten, 76 Bilder, 4 Tab.
2015. ISBN 978-3-87525-398-6.

Band 276: Marcus Baum
Laser Consolidation of ITO Nanoparticles for the Generation of Thin Conductive Layers on Transparent Substrates
LPT, 158 Seiten, 75 Bilder, 3 Tab.
2015. ISBN 978-3-87525-399-3.

Band 277: Thomas Schneider
Umformtechnische Herstellung dünnwandiger Funktionsbauteile aus Feinblech durch Verfahren der Blechmassivumformung
LFT, 188 Seiten, 95 Bilder, 7 Tab.
2015. ISBN 978-3-87525-401-3.

Band 278: Jochen Merhof
Sematische Modellierung automatisierter Produktionssysteme zur Verbesserung der IT-Integration zwischen Anlagen-Engineering und Steuerungsebene
FAPS, 157 Seiten, 88 Bilder, 8 Tab.
2015. ISBN 978-3-87525-402-0.

Band 279: Fabian Zöller
Erarbeitung von Grundlagen zur Abbildung des tribologischen Systems in der Umformsimulation
LFT, 126 Seiten, 51 Bilder, 3 Tab.
2016. ISBN 978-3-87525-403-7.

Band 280: Christian Hezler
Einsatz technologischer Versuche zur Erweiterung der Versagensvorhersage bei Karosseriebauteilen aus höchstfesten Stählen
LFT, 147 Seiten, 63 Bilder, 44 Tab.
2016. ISBN 978-3-87525-404-4.

Band 281: Jochen Bönig
Integration des Systemverhaltens von Automobil-Hochvoltleitungen in die virtuelle Absicherung durch strukturmechanische Simulation
FAPS, 177 Seiten, 107 Bilder, 17 Tab.
2016.
ISBN 978-3-87525-405-1.

Band 282: Johannes Kohl
Automatisierte Datenerfassung für diskret ereignisorientierte Simulationen in der energieflexibelen Fabrik
FAPS, 160 Seiten, 80 Bilder, 27 Tab.
2016.
ISBN 978-3-87525-406-8.

Band 283: Peter Bechtold
Mikroschockwellenumformung mittels ultrakurzer Laserpulse
LPT, 155 Seiten, 59 Bilder, 10 Tab.
2016. ISBN 978-3-87525-407-5.

Band 284: Stefan Berger
Laserstrahlschweißen thermoplastischer Kohlenstofffaserverbundwerkstoffe mit spezifischem Zusatzdraht
LPT, 118 Seiten, 68 Bilder, 9 Tab.
2016. ISBN 978-3-87525-408-2.

Band 285: Martin Bornschlegl
Methods-Energy Measurement - Eine Methode zur Energieplanung für Fügeverfahren im Karosseriebau
FAPS, 136 Seiten, 72 Bilder, 46 Tab. 2016.
ISBN 978-3-87525-409-9.

Band 286: Tobias Rackow
Erweiterung des Unternehmenscontrollings um die Dimension Energie
FAPS, 164 Seiten, 82 Bilder, 29 Tab. 2016.
ISBN 978-3-87525-410-5.

Band 287: Johannes Koch
Grundlegende Untersuchungen zur Herstellung zyklisch-symmetrischer Bauteile mit Nebenformelementen durch Blechmassivumformung
LFT, 125 Seiten, 49 Bilder, 17 Tab. 2016. ISBN 978-3-87525-411-2.

Band 288: Hans Ulrich Vierzigmann
Beitrag zur Untersuchung der tribologischen Bedingungen in der Blechmassivumformung - Bereitstellung von tribologischen Modellversuchen und Realisierung von Tailored Surfaces
LFT, 174 Seiten, 102 Bilder, 34 Tab. 2016. ISBN 978-3-87525-412-9.

Band 289: Thomas Senner
Methodik zur virtuellen Absicherung der formgebenden Operation des Nasspressprozesses von Gelege-Mehrschichtverbunden
LFT, 156 Seiten, 96 Bilder, 21 Tab. 2016. ISBN 978-3-87525-414-3.

Band 290: Sven Kreitlein
Der grundoperationsspezifische Mindestenergiebedarf als Referenzwert zur Bewertung der Energieeffizienz in der Produktion
FAPS, 185 Seiten, 64 Bilder, 30 Tab. 2016.
ISBN 978-3-87525-415-0.

Band 291: Christian Roos
Remote-Laserstrahlschweißen verzinkter Stahlbleche in Kehlnahtgeometrie
LPT, 123 Seiten, 52 Bilder, 0 Tab. 2016. ISBN 978-3-87525-416-7.

Band 292: Alexander Kahrimanidis
Thermisch unterstützte Umformung von Aluminiumblechen
LFT, 165 Seiten, 103 Bilder, 18 Tab. 2016. ISBN 978-3-87525-417-4.

Band 293: Jan Tremel
Flexible Systems for Permanent Magnet Assembly and Magnetic Rotor Measurement / Flexible Systeme zur Montage von Permanentmagneten und zur Messung magnetischer Rotoren
FAPS, 152 Seiten, 91 Bilder, 12 Tab. 2016. ISBN 978-3-87525-419-8.

Band 294: Ioannis Tsoupis
Schädigungs- und Versagensverhalten hochfester Leichtbauwerkstoffe unter Biegebeanspruchung
LFT, 176 Seiten, 51 Bilder, 6 Tab. 2017. ISBN 978-3-87525-420-4.

Band 295: Sven Hildering
Grundlegende Untersuchungen zum Prozessverhalten von Silizium als Werkzeugwerkstoff für das Mikroscherschneiden metallischer Folien
LFT, 177 Seiten, 74 Bilder, 17 Tab. 2017. ISBN 978-3-87525-422-8.

Band 296: Sasia Mareike Hertweck
Zeitliche Pulsformung in der Lasermikromaterialbearbeitung – Grundlegende Untersuchungen und Anwendungen
LPT, 146 Seiten, 67 Bilder, 5 Tab. 2017. ISBN 978-3-87525-423-5.

Band 297: Paryanto
Mechatronic Simulation Approach for the Process Planning of Energy-Efficient Handling Systems
FAPS, 162 Seiten, 86 Bilder, 13 Tab. 2017. ISBN 978-3-87525-424-2.

Band 298: Peer Stenzel
Großserientaugliche Nadelwickeltechnik für verteilte Wicklungen im Anwendungsfall der E-Traktionsantriebe
FAPS, 239 Seiten, 147 Bilder, 20 Tab. 2017.
ISBN 978-3-87525-425-9.

Band 299: Mario Lušić
Ein Vorgehensmodell zur Erstellung montageführender Werkerinformationssysteme simultan zum Produktentstehungsprozess
FAPS, 174 Seiten, 79 Bilder, 22 Tab. 2017.
ISBN 978-3-87525-426-6.

Band 300: Arnd Buschhaus
Hochpräzise adaptive Steuerung und Regelung robotergeführter Prozesse
FAPS, 202 Seiten, 96 Bilder, 4 Tab. 2017. ISBN 978-3-87525-427-3.

Band 301: Tobias Laumer
Erzeugung von thermoplastischen Werkstoffverbunden mittels simultanem, intensitätsselektivem Laserstrahlschmelzen
LPT, 140 Seiten, 82 Bilder, 0 Tab. 2017. ISBN 978-3-87525-428-0.

Band 302: Nora Unger
Untersuchung einer thermisch unterstützten Fertigungskette zur Herstellung umgeformter Bauteile aus der höherfesten Aluminiumlegierung EN AW-7020
LFT, 142 Seiten, 53 Bilder, 8 Tab. 2017. ISBN 978-3-87525-429-7.

Band 303: Tommaso Stellin
Design of Manufacturing Processes for the Cold Bulk Forming of Small Metal Components from Metal Strip
LFT, 146 Seiten, 67 Bilder, 7 Tab. 2017. ISBN 978-3-87525-430-3.

Band 304: Bassim Bachy
Experimental Investigation, Modeling, Simulation and Optimization of Molded Interconnect Devices (MID) Based on Laser Direct Structuring (LDS) / Experimentelle Untersuchung, Modellierung, Simulation und Optimierung von Molded Interconnect Devices (MID) basierend auf Laser Direktstrukturierung (LDS)
FAPS, 168 Seiten, 120 Bilder, 26 Tab. 2017.
ISBN 978-3-87525-431-0.

Band 305: Michael Spahr
Automatisierte Kontaktierungsverfahren für flachleiterbasierte Pkw-Bordnetzsysteme
FAPS, 197 Seiten, 98 Bilder, 17 Tab. 2017. ISBN 978-3-87525-432-7.

Band 306: Sebastian Suttner
Charakterisierung und Modellierung des spannungszustandsabhängigen Werkstoffverhaltens der Magnesiumlegierung AZ31B für die numerische Prozessauslegung
LFT, 150 Seiten, 84 Bilder, 19 Tab. 2017. ISBN 978-3-87525-433-4.

Band 307: Bhargav Potdar
A reliable methodology to deduce thermo-mechanical flow behaviour of hot stamping steels
LFT, 203 Seiten, 98 Bilder, 27 Tab. 2017. ISBN 978-3-87525-436-5.

Band 308: Maria Löffler
Steuerung von Blechmassivumformprozessen durch maßgeschneiderte tribologische Systeme
LFT, viii u. 166 Seiten, 90 Bilder, 5 Tab. 2018. ISBN 978-3-96147-133-1.

Band 309: Martin Müller
Untersuchung des kombinierten Trenn- und Umformprozesses beim Fügen artungleicher Werkstoffe mittels Schneidclinchverfahren
LFT, xi u. 149 Seiten, 89 Bilder, 6 Tab. 2018.
ISBN: 978-3-96147-135-5.

Band 310: Christopher Kästle
Qualifizierung der Kupfer-Drahtbondtechnologie für integrierte Leistungsmodule in harschen Umgebungsbedingungen
FAPS, xii u. 167 Seiten, 70 Bilder, 18 Tab. 2018.
ISBN 978-3-96147-145-4.

Band 311: Daniel Vipavc
Eine Simulationsmethode für das 3-Rollen-Schubbiegen
LFT, xiii u. 121 Seiten, 56 Bilder, 17 Tab. 2018. ISBN 978-3-96147-147-8.

Band 312: Christina Ramer
Arbeitsraumüberwachung und autonome Bahnplanung für ein sicheres und flexibles Roboter-Assistenzsystem in der Fertigung
FAPS, xiv u. 188 Seiten, 57 Bilder, 9 Tab. 2018.
ISBN 978-3-96147-153-9.

Band 313: Miriam Rauer
Der Einfluss von Poren auf die Zuverlässigkeit der Lötverbindungen von Hochleistungs-Leuchtdioden
FAPS, xii u. 209 Seiten, 108 Bilder, 21 Tab. 2018.
ISBN 978-3-96147-157-7.

Band 314: Felix Tenner
Kamerabasierte Untersuchungen der Schmelze und Gasströmungen beim Laserstrahlschweißen verzinkter Stahlbleche
LPT, xxiii u. 184 Seiten, 94 Bilder, 7 Tab. 2018.
ISBN 978-3-96147-160-7.

Band 315: Aarief Syed-Khaja
Diffusion Soldering for High-temperature Packaging of Power Electronics
FAPS, x u. 202 Seiten, 144 Bilder, 32 Tab. 2018.
ISBN 978-3-87525-162-1.

Band 316: Adam Schaub
Grundlagenwissenschaftliche Untersuchung der kombinierten Prozesskette aus Umformen und Additive Fertigung
LFT, xi u. 192 Seiten, 72 Bilder, 27 Tab. 2019.
ISBN 978-3-96147-166-9.

Band 317: Daniel Gröbel
Herstellung von Nebenformelementen unterschiedlicher Geometrie an Blechen mittels Fließpressverfahren der Blechmassivumformung
LFT, x u. 165 Seiten, 96 Bilder, 13 Tab. 2019. ISBN 978-3-96147-168-3.

Band 318: Philipp Hildenbrand
Entwicklung einer Methodik zur Herstellung von Tailored Blanks mit definierten Halbzeugeigenschaften durch einen Taumelprozess
LFT, ix u. 153 Seiten, 77 Bilder, 4 Tab. 2019. ISBN 978-3-96147-174-4.

Band 319: Tobias Konrad
Simulative Auslegung der Spann- und Fixierkonzepte im Karosserierohbau: Bewertung der Baugruppenmaßhaltigkeit unter Berücksichtigung schwankender Einflussgrößen
LFT, x u. 203 Seiten, 134 Bilder, 32 Tab. 2019.
ISBN 978-3-96147-176-8.

Band 320: David Meinel
Architektur applikationsspezifischer Multi-Physics-Simulationskonfiguratoren am Beispiel modularer Triebzüge
FAPS, xii u. 166 Seiten, 82 Bilder, 25 Tab. 2019.
ISBN 978-3-96147-184-3.

Band 321: Andrea Zimmermann
Grundlegende Untersuchungen zum Einfluss fertigungsbedingter Eigenschaften auf die Ermüdungsfestigkeit kaltmassivumgeformter Bauteile
LFT, ix u. 160 Seiten, 66 Bilder, 5 Tab. 2019.
ISBN 978-3-96147-190-4.

Band 322: Christoph Amann
Simulative Prognose der Geometrie nassgepresster Karosseriebauteile aus Gelege-Mehrschichtverbunden
LFT, xvi u. 169 Seiten, 80 Bilder, 13 Tab. 2019.
ISBN 978-3-96147-194-2.

Band 323: Jennifer Tenner
Realisierung schmierstofffreier Tiefziehprozesse durch maßgeschneiderte Werkzeugoberflächen
LFT, x u. 187 Seiten, 68 Bilder, 13 Tab. 2019.
ISBN 978-3-96147-196-6.

Band 324: Susan Zöller
Mapping Individual Subjective Values to Product Design
KTmfk, xi u. 223 Seiten, 81 Bilder, 25 Tab. 2019.
ISBN 978-3-96147-202-4.

Band 325: Stefan Lutz
Erarbeitung einer Methodik zur semiempirischen Ermittlung der Umwandlungskinetik durchhärtender Wälzlagerstähle für die Wärmebehandlungssimulation
LFT, xiv u. 189 Seiten, 75 Bilder, 32 Tab. 2019.
ISBN 978-3-96147-209-3.

Band 326: Tobias Gnibl
Modellbasierte Prozesskettenabbildung rührreibgeschweißter Aluminiumhalbzeuge zur umformtechnischen Herstellung höchstfester Leichtbau-strukturteile
LFT, xii u. 167 Seiten, 68 Bilder, 17 Tab. 2019.
ISBN 978-3-96147-217-8.

Band 327: Johannes Bürner
Technisch-wirtschaftliche Optionen zur Lastflexibilisierung durch intelligente elektrische Wärmespeicher
FAPS, xiv u. 233 Seiten, 89 Bilder, 27 Tab. 2019.
ISBN 978-3-96147-219-2.

Band 328: Wolfgang Böhm
Verbesserung des Umformverhaltens von mehrlagigen Aluminiumblechwerkstoffen mit ultrafeinkörnigem Gefüge
LFT, ix u. 160 Seiten, 88 Bilder, 14 Tab. 2019.
ISBN 978-3-96147-227-7.

Band 329: Stefan Landkammer
Grundsatzuntersuchungen, mathematische Modellierung und Ableitung einer Auslegungsmethodik für Gelenkantriebe nach dem Spinnenbeinprinzip
LFT, xii u. 200 Seiten, 83 Bilder, 13 Tab. 2019.
ISBN 978-3-96147-229-1.

Band 330: Stephan Rapp
Pump-Probe-Ellipsometrie zur Messung transienter optischer Materialeigen-schaften bei der Ultrakurzpuls-Lasermaterialbearbeitung
LPT, xi u. 143 Seiten, 49 Bilder, 2 Tab. 2019.
ISBN 978-3-96147-235-2.

Band 331: Michael Scholz
Intralogistics Execution System mit integrierten autonomen, servicebasierten Transportentitäten
FAPS, xi u. 195 Seiten, 55 Bilder, 11 Tab. 2019.
ISBN 978-3-96147-237-6.

Band 332: Eva Bogner
Strategien der Produktindividualisierung in der produzierenden Industrie im Kontext der Digitalisierung
FAPS, ix u. 201 Seiten, 55 Bilder, 28 Tab. 2019.
ISBN 978-3-96147-246-8.

Band 333: Daniel Benjamin Krüger
Ein Ansatz zur CAD-integrierten muskuloskelettalen Analyse der Mensch-Maschine-Interaktion
KTmfk, x u. 217 Seiten, 102 Bilder, 7 Tab. 2019.
ISBN 978-3-96147-250-5.

Band 334: Thomas Kuhn
Qualität und Zuverlässigkeit laserdirektstrukturierter mechatronisch integrierter Baugruppen (LDS-MID)
FAPS, ix u. 152 Seiten, 69 Bilder, 12 Tab. 2019.
ISBN: 978-3-96147-252-9.

Band 335: Hans Fleischmann
Modellbasierte Zustands- und Prozessüberwachung auf Basis soziocyber-physischer Systeme
FAPS, xi u. 214 Seiten, 111 Bilder, 18 Tab. 2019.
ISBN: 978-3-96147-256-7.

Band 336: Markus Michalski
Grundlegende Untersuchungen zum Prozess- und Werkstoffverhalten bei schwingungsüberlagerter Umformung
LFT, xii u. 197 Seiten, 93 Bilder, 11 Tab. 2019.
ISBN: 978-3-96147-270-3.

Band 337: Markus Brandmeier
Ganzheitliches ontologiebasiertes Wissensmanagement im Umfeld der industriellen Produktion
FAPS, xi u. 255 Seiten, 77 Bilder, 33 Tab. 2020.
ISBN: 978-3-96147-275-8.

Band 338: Stephan Purr
Datenerfassung für die Anwendung lernender Algorithmen bei der Herstellung von Blechformteilen
LFT, ix u. 165 Seiten, 48 Bilder, 4 Tab. 2020.
ISBN: 978-3-96147-281-9.

Band 339: Christoph Kiener
Kaltfließpressen von gerad- und schrägverzahnten Zahnrädern
LFT, viii u. 151 Seiten, 81 Bilder, 3 Tab. 2020.
ISBN 978-3-96147-287-1.

Band 340: Simon Spreng
Numerische, analytische und empirische Modellierung des Heißcrimpprozesses
FAPS, xix u. 204 Seiten, 91 Bilder, 27 Tab. 2020.
ISBN 978-3-96147-293-2.

Band 341: Patrik Schwingenschlögl
Erarbeitung eines Prozessverständnisses zur Verbesserung der tribologischen Bedingungen beim Presshärten
LFT, x u. 177 Seiten, 81 Bilder, 8 Tab. 2020.
ISBN 978-3-96147-297-0.

Band 342: Emanuela Affronti
Evaluation of failure behaviour of sheet metals
LFT, ix u. 136 Seiten, 57 Bilder, 20 Tab. 2020.
ISBN 978-3-96147-303-8.

Band 343: Julia Degner
Grundlegende Untersuchungen zur Herstellung hochfester Aluminiumblechbauteile in einem kombinierten Umform- und Abschreckprozess
LFT, x u. 172 Seiten, 61 Bilder, 9 Tab. 2020.
ISBN 978-3-96147-307-6.

Band 344: Maximilian Wagner
Automatische Bahnplanung für die Aufteilung von Prozessbewegungen in synchrone Werkstück- und Werkzeugbewegungen mittels Multi-Roboter-Systemen
FAPS, xxi u. 181 Seiten, 111 Bilder, 15 Tab. 2020.
ISBN 978-3-96147-309-0.

Band 345: Stefan Härter
Qualifizierung des Montageprozesses hochminiaturisierter elektronischer Bauelemente
FAPS, ix u. 194 Seiten, 97 Bilder, 28 Tab. 2020.
ISBN 978-3-96147-314-4.

Band 346: Toni Donhauser
Ressourcenorientierte Auftragsregelung in einer hybriden Produktion mittels betriebsbegleitender Simulation
FAPS, xix u. 242 Seiten, 97 Bilder, 17 Tab. 2020.
ISBN 978-3-96147-316-8.

Band 347: Philipp Amend
Laserbasiertes Schmelzkleben von Thermoplasten mit Metallen
LPT, xv u. 154 Seiten, 67 Bilder. 2020. ISBN 978-3-96147-326-7.

Band 348: Matthias Ehlert
Simulationsunterstützte funktionale Grenzlagenabsicherung
KTmfk, xvi u. 300 Seiten, 101 Bilder, 73 Tab. 2020.
ISBN 978-3-96147-328-1.

Band 349: Thomas Sander
Ein Beitrag zur Charakterisierung und Auslegung des Verbundes von Kunststoffsubstraten mit harten Dünnschichten
KTmfk, xiv u. 178 Seiten, 88 Bilder, 21 Tab. 2020.
ISBN 978-3-96147-330-4.

Band 350: Florian Pilz
Fließpressen von Verzahnungselementen an Blechen
LFT, x u. 170 Seiten, 103Bilder, 4 Tab. 2020.
ISBN 978-3-96147-332-8.

Band 351: Sebastian Josef Katona
Evaluation und Aufbereitung von Produktsimulationen mittels abweichungsbehafteter Geometriemodelle
KTmfk, ix u. 147 Seiten, 73 Bilder, 11 Tab. 2020.
ISBN 978-3-96147-336-6.

Band 352: Jürgen Herrmann
Kumulatives Walzplattieren. Bewertung der Umformeigenschaften mehrlagiger Blechwerkstoffe der ausscheidungshärtbaren Legierung AA6014
LFT, x u. 157 Seiten, 64 Bilder, 5 Tab. 2020.
ISBN 978-3-96147-344-1.

Band 353: Christof Küstner
Assistenzsystem zur Unterstützung der datengetriebenen Produktentwicklung
KTmfk, xii u. 219 Seiten, 63 Bilder, 14 Tab. 2020.
ISBN 978-3-96147-348-9.

Band 354: Tobias Gläßel
Prozessketten zum Laserstrahlschweißen von flachleiterbasierten Formspulenwicklungen für automobile Traktionsantriebe
FAPS, xiv u. 206 Seiten, 89 Bilder, 11 Tab. 2020.
ISBN 978-3-96147-356-4.

Band 355: Andreas Meinel
Experimentelle Untersuchung der Auswirkungen von Axialschwingungen auf Reibung und Verschleiß in Zylinderrol-lenlagern
KTmfk, xii u. 162 Seiten, 56 Bilder, 7 Tab. 2020.
ISBN 978-3-96147-358-8.

Band 356: Hannah Riedle
Haptische, generische Modelle weicher anatomischer Strukturen für die chirurgische Simulation
FAPS, xxx u. 179 Seiten, 82 Bilder, 35 Tab. 2020.
ISBN 978-3-96147-367-0.

Band 357: Maximilian Landgraf
Leistungselektronik für den Einsatz dielektrischer Elastomere in aktorischen, sensorischen und integrierten sensomotorischen Systemen
FAPS, xxiii u. 166 Seiten, 71 Bilder, 10 Tab. 2020.
ISBN 978-3-96147-380-9.

Band 358: Alireza Esfandyari
Multi-Objective Process Optimization for Overpressure Reflow Soldering in Electronics Production
FAPS, xviii u. 175 Seiten, 57 Bilder, 23 Tab. 2020.
ISBN 978-3-96147-382-3.

Band 359: Christian Sand
Prozessübergreifende Analyse komplexer Montageprozessketten mittels Data Mining
FAPS, XV u. 168 Seiten, 61 Bilder, 12 Tab. 2021.
ISBN 978-3-96147-398-4.

Band 360: Ralf Merkl
Closed-Loop Control of a Storage-Supported Hybrid Compensation System for Improving the Power Quality in Medium Voltage Networks
FAPS, xxvii u. 200 Seiten, 102 Bilder, 2 Tab. 2021.
ISBN 978-3-96147-402-8.

Band 361: Thomas Reitberger
Additive Fertigung polymerer optischer Wellenleiter im Aerosol-Jet-Verfahren
FAPS, xix u. 141 Seiten, 65 Bilder, 11 Tab. 2021.
ISBN 978-3-96147-400-4.

Band 362: Marius Christian Fechter
Modellierung von Vorentwürfen in der virtuellen Realität mit natürlicher Fingerinteraktion
KTmfk, x u. 188 Seiten, 67 Bilder, 19 Tab. 2021.
ISBN 978-3-96147-404-2.

Band 363: Franziska Neubauer
Oberflächenmodifizierung und Entwicklung einer Auswertemethodik zur Verschleißcharakterisierung im Presshärteprozess
LFT, ix u. 177 Seiten, 42 Bilder, 6 Tab. 2021.
ISBN 978-3-96147-406-6.

Band 364: Eike Wolfram Schäffer
Web- und wissensbasierter Engineering-Konfigurator für roboterzentrierte Automatisierungslösungen
FAPS, xxiv u. 195 Seiten, 108 Bilder, 25 Tab. 2021.
ISBN 978-3-96147-410-3.

Band 365: Daniel Gross
Untersuchungen zur kohlenstoffdioxidbasierten kryogenen Minimalmengenschmierung
REP, xii u. 184 Seiten, 56 Bilder, 18 Tab. 2021.
ISBN 978-3-96147-412-7.

Band 366: Daniel Junker
Qualifizierung laser-additiv gefertigter Komponenten für den Einsatz im Werkzeugbau der Massivumformung
LFT, vii u. 142 Seiten, 62 Bilder, 5 Tab. 2021.
ISBN 978-3-96147-416-5.

Band 367: Tallal Javied
Totally Integrated Ecology Management for Resource Efficient and Eco-Friendly Production
FAPS, xv u. 160 Seiten, 60 Bilder, 13 Tab. 2021.
ISBN 978-3-96147-418-9.

Band 368: David Marco Hochrein
Wälzlager im Beschleunigungsfeld – Eine Analysestrategie zur Bestimmung des Reibungs-, Axialschub- und Temperaturverhaltens von Nadelkränzen –
KTmfk, xiii u. 279 Seiten, 108 Bilder, 39 Tab. 2021.
ISBN 978-3-96147-420-2.

Band 369: Daniel Gräf
Funktionalisierung technischer Oberflächen mittels prozessüberwachter aerosolbasierter Drucktechnologie
FAPS, xxii u. 175 Seiten, 97 Bilder, 6 Tab. 2021.
ISBN 978-3-96147-433-2.

Band 370: Andreas Gröschl
Hochfrequent fokusabstandsmodulierte Konfokalsensoren für die Nanokoordinatenmesstechnik
FMT, x u. 144 Seiten, 98 Bilder, 6 Tab. 2021.
ISBN 978-3-96147-435-6.

Band 371: Johann Tüchsen
Konzeption, Entwicklung und Einführung des Assistenzsystems D-DAS für die Produktentwicklung elektrischer Motoren
KTmfk, xii u. 178 Seiten, 92 Bilder, 12 Tab. 2021.
ISBN 978-3-96147-437-0.

Band 372: Max Marian
Numerische Auslegung von Oberflächenmikrotexturen für geschmierte tribologische Kontakte
KTmfk, xviii u. 276 Seiten, 85 Bilder, 45 Tab. 2021.
ISBN 978-3-96147-439-4.

Band 373: Johannes Strauß
Die akustooptische Strahlformung in der Lasermaterialbearbeitung
LPT, xvi u. 113 Seiten, 48 Bilder. 2021. ISBN 978-3-96147-441-7.

Band 374: Martin Hohmann
Machine learning and hyper spectral imaging: Multi Spectral Endoscopy in the Gastro Intestinal Tract towards Hyper Spectral Endoscopy
LPT, x u. 137 Seiten, 62 Bilder, 29 Tab. 2021.
ISBN 978-3-96147-445-5.

Band 375: Timo Kordaß
Lasergestütztes Verfahren zur selektiven Metallisierung von epoxidharzbasierten Duromeren zur Steigerung der Integrationsdichte für dreidimensionale mechatronische Package-Baugruppen
FAPS, xviii u. 198 Seiten, 92 Bilder, 24 Tab. 2021.
ISBN 978-3-96147-443-1.

Band 376: Philipp Kestel
Assistenzsystem für den wissensbasierten Aufbau konstruktionsbegleitender Finite-Elemente-Analysen
KTmfk, xviii u. 209 Seiten, 57 Bilder, 17 Tab. 2021.
ISBN 978-3-96147-457-8.

Band 377: Martin Lerchen
Messverfahren für die pulverbettbasierte additive Fertigung zur Sicherstellung der Konformität mit geometrischen Produktspezifikationen
FMT, x u. 150 Seiten, 60 Bilder, 9 Tab. 2021.
ISBN 978-3- 96147-463-9.

Band 378: Michael Schneider
Inline-Prüfung der Permeabilität in weichmagnetischen Komponenten
FAPS, xxii u. 189 Seiten, 79 Bilder, 14 Tab. 2021.
ISBN 978-3-96147-465-3.

Band 379: Tobias Sprügel
Sphärische Detektorflächen als Unterstützung der Produktentwicklung zur Datenanalyse im Rahmen des Digital Engineering
KTmfk, xiii u. 213 Seiten, 84 Bilder, 33 Tab. 2021.
ISBN 978-3-96147-475-2.

Band 380: Tom Häfner
Multipulseffekte beim Mikro-Materialabtrag von Stahllegierungen mit Pikosekunden-Laserpulsen
LPT, xxviii u. 159 Seiten, 57 Bilder, 13 Tab. 2021.
ISBN 978-3-96147-479-0.

Band 381: Björn Heling
Einsatz und Validierung virtueller Absicherungsmethoden für abweichungs-behaftete Mechanismen im Kontext des Robust Design
KTmfk, xi u. 169 Seiten, 63 Bilder, 27 Tab. 2021.
ISBN 978-3-96147-487-5.

Band 382: Tobias Kolb
Laserstrahl-Schmelzen von Metallen mit einer Serienanlage – Prozesscharakterisierung und Erweiterung eines Überwachungssystems
LPT, xv u. 170 Seiten, 128 Bilder, 16 Tab. 2021.
ISBN 978-3-96147-491-2.

Band 383: Mario Meinhardt
Widerstandselementschweißen mit gestauchten Hilfsfügeelementen - Umformtechnische Wirkzusammenhänge zur Beeinflussung der Verbindungsfestigkeit
LFT, xii u. 189 Seiten, 87 Bilder, 4 Tab. 2022.
ISBN 978-3-96147-473-8.

Band 384: Felix Bauer
Ein Beitrag zur digitalen Auslegung von Fügeprozessen im Karosseriebau mit Fokus auf das Remote-Laserstrahlschweißen unter Einsatz flexibler Spanntechnik
LFT, xi u. 185 Seiten, 74 Bilder, 12 Tab. 2022.
ISBN 978-3-96147-498-1.

Band 385: Jochen Zeitler
Konzeption eines rechnergestützten Konstruktionssystems für optomechatronische Baugruppen
FAPS, xix u. 172 Seiten, 88 Bilder, 11 Tab. 2022.
ISBN 978-3-96147-499-8.

Band 386: Vincent Mann
Einfluss von Strahloszillation auf das Laserstrahlschweißen hochfester Stähle
LPT, xiii u. 172 Seiten, 103 Bilder, 18 Tab. 2022.
ISBN 978-3-96147-503-2.

Band 387: Chen Chen
Skin-equivalent opto-/elastofluidic in-vitro microphysiological vascular models for translational studies of optical biopsies
LPT, xx u. 126 Seiten, 60 Bilder, 10 Tab. 2022.
ISBN 978-3-96147-505-6.

Band 388: Stefan Stein
Laser drop on demand joining as bonding method for electronics assembly and packaging with high thermal requirements
LPT, x u. 112 Seiten, 54 Bilder, 10 Tab. 2022.
ISBN 978-3-96147-507-0

Band 389: Nikolaus Urban
Untersuchung des Laserstrahlschmelzens von Neodym-Eisen-Bor zur additiven Herstellung von Permanentmagneten
FAPS, x u. 174 Seiten, 88 Bilder, 18 Tab. 2022.
ISBN: 978-3-96147-501-8.

Band 390: Yiting Wu
Großflächige Topographiemessungen mit einem Weißlichtinterferenzmikroskop und einem metrologischen Rasterkraftmikroskop
FMT, xii u. 142 Seiten, 68 Bilder, 11 Tab. 2022.
ISBN: 978-3-96147-513-1.

Band 391: Thomas Papke
Untersuchungen zur Umformbarkeit hybrider Bauteile aus Blechgrundkörper und additiv gefertigter Struktur
LFT, xii u. 194 Seiten, 71 Bilder, 16 Tab. 2022.
ISBN 978-3-96147-515-5.

Band 392: Bastian Zimmermann
Einfluss des Vormaterials auf die mehrstufige Kaltumformung vom Draht
LFT, xi u. 182 Seiten, 36 Bilder, 6 Tab. 2022.
ISBN 978-3-96147-519-3.

Band 393: Harald Völkl
Ein simulationsbasierter Ansatz zur Auslegung additiv gefertigter FLM-Faserverbundstrukturen
KTmfk, xx u. 204 Seiten, 95 Bilder, 22 Tab. 2022.
ISBN 978-3-96147-523-0.

Band 394: Robert Schulte
Auslegung und Anwendung prozessangepasster Halbzeuge für Verfahren der Blechmassivumformung
LFT, x u. 163 Seiten, 93 Bilder, 5 Tab. 2022.
ISBN 978-3-96147-525-4.

Band 395: Philipp Frey
Umformtechnische Strukturierung metallischer Einleger im Folgeverbund für mediendichte Kunststoff-Metall-Hybridbauteile
LFT, ix u. 180 Seiten, 83 Bilder, 7 Tab. 2022.
ISBN 978-3-96147-534-6.

Band 396: Thomas Johann Luft
Komplexitätsmanagement in der Produktentwicklung - Holistische Modellierung, Analyse, Visualisierung und Bewertung komplexer Systeme
KTmfk, xiii u. 510 Seiten, 166 Bilder, 16 Tab. 2022.
ISBN 978-3-96147-540-7.

Band 397: Li Wang
Evaluierung der Einsetzbarkeit des lasergestützten Verfahrens zur selektiven Metallisierung für die Verbesserung passiver Intermodulation in Hochfrequenzanwendungen
FAPS, xxii u.151 Seiten, 72 Bilder, 22 Tab. 2022.
ISBN 978-3-96147-542-1.

Band 398: Sebastian Reitelshöfer
Der Aerosol-Jet-Druck Dielektrischer Elastomere als additives Fertigungsverfahren für elastische mechatronische Komponenten
FAPS, xxv u. 206 Seiten, 87 Bilder, 13 Tab. 2022.
ISBN 978-3-96147-547-6.

Band 399: Alexander Meyer
Selektive Magnetmontage zur Verringerung des Rastmomentes permanenterregter Synchronmotoren
FAPS, xv u. 164 Seiten, 90 Bilder, 18 Tab. 2022.
ISBN 978-3-96147-555-1.

Band 400: Rong Zhao
Design verschleißreduzierender amorpher Kohlenstoffschichtsysteme für trockene tribologische Gleitkontakte
KTmfk, x u. 148 Seiten, 69 Bilder, 14 Tab. 2022.
ISBN 978-3-96147-557-5.

Band 401: Christian P. J. Schwarzer
Kupfersintern als Fügetechnologie für Leistungselektronik
FAPS, xxvii u. 234 Seiten, 125 Bilder, 24 Tab. 2022.
ISBN 978-3-96147-566-7.

Band 402: Alexander Horn
Grundlegende Untersuchungen zur Gradierung der mechanischen Eigenschaften pressgehärteter Bauteile durch eine örtlich begrenzte Aufkohlung
LFT, xii u. 204 Seiten, 58 Bilder, 6 Tab. 2022.
ISBN 978-3-96147-568-1.

Band 403: Artur Klos
Werkstoff- und umformtechnische Bewertung von hochfesten Aluminiumblechwerkstoffen für den Karosseriebau
LFT, x u. 192 Seiten, 73 Bilder, 12 Tab. 2022.
ISBN 978-3-96147-572-8.

Band 404: Harald Schmid
Ganzheitliche Erarbeitung eines Prozessverständnisses von Tiefziehprozessen mit Ziehsicken auf Basis mechanischer und tribologischer Analysen
LFT, xiii u. 211 Seiten, 78 Bilder, 5 Tab. 2022.
ISBN 978-3-96147-577-3.

Band 405: Johannes Henneberg
Blechmassivumformung von Funktionsbauteilen aus Bandmaterial
LFT, viii u. 176 Seiten, 101 Bilder, 2 Tab. 2022.
ISBN 978-3-96147-579-7.

Band 406: Anton Schmailzl
Festigkeits- und zeitoptimierte Prozessführung beim quasi-simultanen Laser-Durchstrahlschweißen
LPT, xiii u. 157 Seiten, 84 Bilder, 7 Tab. 2022.
ISBN 978-3-96147-583-4.

Band 407: Alexander Wolf
Modellierung und Vorhersage menschlichen Interaktionsverhaltens zur Analyse der Mensch-Produkt Interaktion
KTmfk, x u. 207 Seiten, 69 Bilder, 10 Tab. 2022.
ISBN 978-3-96147-585-8.

Band 408: Tim Weikert
Modifikationen amorpher Kohlenstoffschichten zur Anpassung der Reibungsbedingungen und zur Erhöhung des Verschleißschutzes
KTmfk, xvii u. 258 Seiten, 91 Bilder, 9 Tab. 2022.
ISBN 978-3-96147-589-6.

Band 409: Stefan Götz
Frühzeitiges konstruktionsbegleitendes Toleranzmanagement
KTmfk, ix u. 276 Seiten, 127 Bilder, 13 Tab. 2022.
ISBN 978-3-96147-593-3.

Band 410: Markus Hubert
Einsatzpotenziale der Rotationsschneidtechnologie in der Verarbeitung von metallischen Funktionsfolien für mechatronische Produkte
FAPS, xviii u. 139 Seiten, 86 Bilder, 7 Tab. 2022
ISBN 978-3-96147-603-9.

Band 411: Manfred Vogel
Grundlagenuntersuchungen und Erarbeitung einer Methodik zur Herstellung maßgeschneiderter Halbzeuge auf Basis eines neuartigen flexiblen Walzprozesses
LFT, ix u. 176 Seiten, 61 Bilder, 11 Tab. 2022.
ISBN 978-3-96147-605-3.

Band 412: Michael Weigelt
Multidimensionale Optionenanalyse alternativer Antriebskonzepte für die individuelle Langstreckenmobilität
FAPS, xv u. 222 Seiten, 89 Bilder, 38 Tab. 2022
ISBN 978-3-96147-607-7.

Band 413: Frank Bodendorf
Machine Learning im Cost Engineering des Supply Managements
FAPS, xiii u. 166 Seiten, 75 Bilder, 13 Tab. 2022
ISBN 978-3-96147-609-1.

Band 414: Maximilian Metzler
Planung und Simulation taktiler, intelligenter und kollaborativer Roboterfähigkeiten in der Montage
FAPS, xix u. 174 Seiten, 72 Bilder, 3 Tab. 2022.
ISBN 978-3-96147-611-4.

Band 415: Tina Buker
Ein Ansatz zur Reduktion produktinduzierter Nutzerstigmatisierung durch Förderung einer gleichermaßen gebrauchstauglichen wie emotionalen Produktgestalt
KTmfk, x u. 236 Seiten, 54 Bilder, 44 Tab. 2022.
ISBN: 978-3-96147-613-8.

Band 416: Marlene Kuhn
Model-based Traceability System Development for Complex Manufacturing Applying Blockchain and Graphs
FAPS, xv u. 167 Seiten, 63 Bilder, 10 Tab. 2022.
ISBN: 978-3-96147-615-2

Kurzzusammenfassung

Rückverfolgbarkeitssysteme sind der Schlüssel zu einer intelligenten Fertigung, da sie für Transparenz entlang der Wertschöpfungskette sorgen und eine strukturierte Dokumentation der Produkthistorie ermöglichen. Mit den steigenden regulatorischen und organisatorischen Anforderungen haben sich Rückverfolgbarkeitssysteme über die letzten Jahre von einem Instrument zur Risikominimierung zu einer wesentlichen Säule der Datenrevolution im Kontext von Industrie 4.0 entwickelt. Insbesondere in der Automobilindustrie steigt ihre Bedeutung an, da das Rückverfolgbarkeitssystem es ermöglicht, die Kosten durch eine gezieltere Eingrenzung der Rückrufe zu reduzieren und die Sicherheit der Produkte durch eine schnelle Fehlersuche zu gewährleisten.

In dieser Arbeit wird eine Methodik zur systematischen Modellierung von Rückverfolgbarkeitssystemen vorgestellt. Das entwickelte Modell baut auf einer standardisierten Terminologie auf, die aus Trace Objekten, Trace Links, Trace Akteuren sowie Tracking und Tracing Funktionen besteht, und umfasst sowohl Produktions- als auch Lieferkettendaten. Basierend auf einer Graphdatenbank und einer Blockchain wird das Modell für einen Anwendungsfall aus der Automobilindustrie spezifiziert und ganzheitlich implementiert. Die Graphdatenbank ermöglicht die Verknüpfung semantisch angereicherter und detaillierter Fertigungsdaten, während die Ethereum-basierte Blockchain-Lösung Daten über verschiedene Produktionsstandorte hinweg vernetzt und aggregiert.

Traceability systems are the key enablers to smart manufacturing, as they provide transparency and structured documentation along with a product's value generation flow. With rising regulatory and organizational require-ments, traceability systems have developed from a pure risk mitigation tool to an essential pillar of the data revolution in the context of Industry 4.0. Especially in the automotive industry, recall costs are growing exponen-tially with particularly high growth rates for electrified and autonomous vehicles. A traceability system helps to reduce these costs through a more targeted containment of the recalls.

This thesis presents a modeling methodology to systematically develop traceability in manufacturing industries. In alignment with the proposed methodology, a traceability model for complex manufacturing systems is developed. The model builds on a standardized traceability terminology consisting of trace objects, trace links, trace actors as well as tracking and tracing functions, and encompasses manufacturing data and supply chain data. The model is implemented for an automotive use case through a ho-listic application based on a graph database and a blockchain. The graph database allows to connect and store semantically rich and detailed manu-facturing data, while the Ethereum-based permissioned blockchain enables tracing macro data for products as they move through the supply chain. The developed solution thus provides full transparency and safe documen-tation to complex and opaque production networks.

ISBN 978-3-96147-615-2

9 783961 476152

FAU UNIVERSITY PRESS 2022