Otto-von-Guericke-Universität Magdeburg

Recovery planning method for production systems

Dissertation

zur Erlangung des akademischen Grades

Doktor-Ingenieurin

(Dr.-Ing.)

von Dipl.-Ing. Nicole Schmidt
geboren am 20.11.1986 in Magdeburg

genehmigt durch die Fakultät für Maschinenbau
der Otto-von-Guericke-Universität Magdeburg

Gutachter: apl. Prof. Dr.-Ing. habil. Arndt Lüder
Prof. Dr.-Ing. Rainer Stark

Promotionskolloquium am 18.05.2018

Preface

This thesis has been created during my work at the Institute of Ergonomics, Manufacturing Systems and Automation (IAF) of the Otto-von-Guericke University Magdeburg.

I would like to thank apl. Prof. Dr.-Ing. habil. Arndt Lüder for his supervision, his time for the many positive and fruitful discussions, the possibilities he gave me to present my work at various occasions, and for taking over the primary assessment. Furthermore, I thank Prof. Dr.-Ing. Rainer Stark for his secondary assessment and his valuable feedback on my work.

Moreover, I would like to thank all the experts, who were willing to invest their time for the insightful and informative discussions as well as their patience to answer all of my questions. Their expert knowledge was essential for this thesis.

For reviewing my thesis I would like to thank Anka Gerling, Carola and André Kösser.

Also I would like thank all employees, colleagues, and students of the chair of Factory Operations and Manufacturing Systems, who inspired me and, therefore, contributed to my work.

I thank my whole family, especially my parents and my beloved Erik Syllwasschy, who have always been on my side and gave me the strength and support that I needed to write and also to complete this thesis.

Finally, I want to thank the German state, which paid for my education. Without this I would not have been able to get this far.

I would like to conclude this preface with a quote of a wise man, whose words became more and more true with time:

"The more you know, the more you know you don't know."

- Aristotle

Magdeburg, 17.01.2018 Nicole Schmidt

Vorwort

Diese Arbeit entstand während meiner Tätigkeit am Institut für Arbeitswissenschaft, Fabrikautomatisierung und Fabrikbetrieb (IAF) der Otto-von-Guericke Universität Magdeburg.

Ich danke apl. Prof. Dr.-Ing. habil. Arndt Lüder für seine Betreuung, seine Zeit für die vielen konstruktiven und ergiebigen Diskussionen, die Möglichkeiten, die er mir bot, meine Arbeit bei unterschiedlichen Gelegenheiten zu präsentieren und die Übernahme des Erstgutachtens. Weiterhin gilt mein Dank Prof. Dr.-Ing. Rainer Stark für die Übernahme des Zweitgutachtens und sein wertvolles Feedback zu meiner Arbeit.

Außerdem möchte ich all den Experten danken, die sich Zeit für die aufschlussreichen und lehrreichen Diskussionen nahmen sowie für ihre Geduld, all meine Fragen zu beantworten. Ihr Expertenwissen war grundlegend für die Arbeit.

Für das Korrekturlesen meiner Arbeit möchte ich Anka Gerling, Carola und André Kösser danken.

Darüber hinaus möchte ich mich bei allen Mitarbeitern, Kollegen und Studenten des Lehrstuhls für Fabrikbetrieb und Produktionssysteme bedanken, die mich inspiriert haben und damit einen Teil zu meiner Arbeit beitrugen.

Ich danke meiner gesamten Familie, insbesondere meinen Eltern und meinem geliebten Erik Syllwasschy, welche mir während der gesamten Zeit zur Seite standen und mir die notwendige Kraft und Unterstützung zu Teil kommen ließen, diese Arbeit zu schreiben und auch zu schaffen.

Zu guter Letzt möchte ich mich beim deutschen Staat bedanken, der für die Kosten meiner Ausbildung aufkam. Ohne diese wäre es mir nicht möglich gewesen, so weit zu kommen.

Das Vorwort möchte ich mit den Worten eines weisen Mannes abschließen, die mit der Zeit immer mehr an Wahrheit gewonnen haben:

“Je mehr man weiß, desto mehr weiß man, dass man es nicht weiß.”
(eigene Übersetzung)

- Aristoteles

Magdeburg, 17.01.2018 Nicole Schmidt

Abstract

The optimization of the production system life cycle has been a focus of research and industry for many years. With the paradigm Industrie 4.0 it was put on the agenda again by demanding the digitization of the industry and the optimization of the entire value chain along life cycles.
To enable such typical Industrie 4.0 use cases, like 'Adaptable Factory', 'Order-Controlled Production', or 'Value-Based Services', a life cycle thinking is required, i.e. the optimization of the life cycle with its interconnections is requested rather than the optimization of isolated life cycle phases. However, one phase - often forgotten by academia and industry - is the last phase of the production system life cycle: the End-of-Life (EoL) phase. This is addressed by the less prominent Industrie 4.0 use case 'Circular Economy'. Here, the scope is to recover machines, products, components, and materials to keep them in cycles, while recovery involves concepts, like recycling and reuse. This results in the connection of the EoL phase with multiple life cycle phases and can, therefore, be another and promising starting point for optimization potentials of the production system life cycle. The author thinks it is worth investigating the range and impact of this last phase, which can benefit from the Industrie 4.0 paradigm and can contribute to transform the industry from today's continuous economy to a circular one.

This thesis strives to understand and describe the EoL phase of production systems in its entirety including the possible EoL treatments, their corresponding recovery processes, and necessary information. Based on this a data model and method is proposed, which could support the proper choice of an EoL treatment for a production system. Additionally, a realization for the data representation is developed that can be a basis for future software tools supporting systemically and holistically the EoL phase.

For this, the first part of this thesis analyzes and evaluates the state of the art by conducting a literature review as well as seven case studies. This results in an EoL data model, insights into possible methodical EoL support, and in a detailed description of the EoL business ecosystem.
The second part uses the conclusions and imitations of the state of the art to develop a method to support the EoL phase in finding a proper EoL treatment for production systems. In addition, the data representation is developed and exemplarily applied.
In the third part three of this thesis the findings are critically discussed and possible future research work is derived.

Kurzfassung

Die Optimierung des Lebenszyklus von Produktionssystemen steht bereits seit mehreren Jahren im Fokus von Forschung und Industrie. Mit dem Trend Industrie 4.0 steht es nun durch die Forderung nach Digitalisierung der Industrie und der Optimierung der gesamten Wertschöpfungskette entlang beteiligter Lebenszyklen erneut auf der Tagesordnung.
Um die typischen Industrie 4.0 Anwendungsfälle, wie 'Wandlungsfähige Fabrik', 'Auftragsgesteuerte Produktion' oder 'Value-Based Services', zu ermöglichen, ist ein Lebenszyklus-denken vonnöten. Das heißt, es steht nicht die Optimierung von einzelnen Lebenszyklusphasen im Vordergrund, sondern die Optimierung des gesamten Lebenszyklus inklusive seiner Vernetzung. Jedoch wird eine Lebenszyklusphase regelmäßig von Forschung und Industrie vergessen. Es handelt sich um die letzte Phase im Lebenszyklus von Produktionssystemen: die End-of-Life (EoL) Phase. Diese wird auch im weniger bekannten Industrie 4.0 Anwendungsfall 'Kreislaufwirtschaft' adressiert. Hier steht die Rückgewinnung von Maschinen, Produkten, Komponenten und Materialien im Vordergrund, wobei Rückgewinnung Konzepte, wie Recycling und Wiederverwendung, beinhalten. Damit ist die EoL Phase mit diversen Lebenszyklusphasen verbunden und kann damit einen anderen und vielversprechenden Ansatzpunkt für Optimierungen des Lebenszyklus von Produktionssystemen bieten. Der Autor denkt, es ist sinnvoll, die Ausmaße und den Einfluss dieser letzten Phase zu untersuchen, die vom Trend Industrie 4.0 profitieren und zur Transformation der Industrie von einer Linearwirtschaft zu einer Kreislaufwirtschaft beitragen kann.

Diese Arbeit strebt an, die EoL Phase in ihrer Gesamtheit zu verstehen und zu beschreiben inklusive ihrer möglichen EoL Optionen, den dazugehörigen Rückgewinnungsprozessen sowie den dazu erforderlichen Informationen. Darauf basierend werden ein Datenmodell und eine Methode vorgeschlagen, die dabei unterstützen kann, eine geeignete EoL Option für ein Produktionssystem zu finden. Zusätzlich wird eine Umsetzung der Datenmodellierung entwickelt, die als Grundlage für zukünftige Softwarewerkzeuge zur systematischen und ganzheitlichen Unterstützung der EoL Phase dienen kann.

Dafür analysiert und bewertet der erste Teil der Arbeit den Stand der Technik, der durch eine Literaturrecherche sowie durch sieben Fallstudien zusammengetragen wurde. Daraus werden dann ein EoL Datenmodell, Einblicke in eine denkbare, methodische Unterstützung der EoL Phase und einer detaillierten Beschreibung des EoL Business Ecosystem erarbeitet.
Der zweite Teil nutzt die Erkenntnisse und Forschungslücken, die im ersten Teil gefunden wurden, und entwickelt eine Methode, die dabei unterstützen kann, geeignete EoL Optionen für Produktionssystem zu finden. Zusätzlich wird die Datenmodellierung realisiert und exemplarisch angewandt.
Im dritten und letzten Teil der Arbeit werden die Ergebnisse der Arbeit kritisch hinterfragt und zukünftiger Forschungsbedarf abgeleitet.

Contents

List of Figures

List of Tables

Glossary

Artifact An artifact comprises information and/or a physical asset.

Asset An asset is “a useful or valuable thing ...” [Oxf17a]. In this thesis the two asset types ’information asset’ and ’physical asset’ are used [ISO14b]. In this thesis ’information asset’ is simply called ’information’.

Commissioning Commissioning is the transition of a production system from a non-operating state into a permanent operating state [Web08].

Component In this thesis, a component is considered as a subsystem or system element - depending on the viewpoint. “A System Element is an identifiable part of a system providing essential parts of the overall system behavior by its own and enabling the system to fulfill its aim. A Subsystem is a system itself. Besides it is part of and fully contained by another upper system. In the scope of the upper system the subsystem is similar to a system element.” [SLR+15]

Connection A connection is “a relationship in which a ... thing is linked or associated with something else” [Oxf17b]. In this thesis connection is understood as a physical link between components or between the production system and components.

Data Data is a “reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing” [Int98]. Data is somehow transported on a medium.

Data model A data model is a formal representation of data properties (based on [Lin09]).

Decommissioning Decommissioning is the transition of a production system from a permanent operating state to a non-operating state.

Digital representation In this thesis, digital is not understood as the contrast of analog as “... signals or data expressed as series of the digits 0 and 1, typically represented by values of a physical quantity ...” [Oxf17c]. Instead digital is understood in the context of Industrie 4.0 with digitalization and digital shadow [Pla16a], [Fra]. A digital representation of a production system is the model of the production system that physically exists.

Disassembly Disassembly is a (planned) sequence of processes that leads to a separation of a multi-body system into components and/or amorphous materials [VDI01a].

Disposal Disposal is the dumping of physical assets or the incineration without energy recovery. [VDI02]

Engineering Engineering is the task to solve technical problems and to optimize "those solutions within the requirements and constraints set by material, technological, economic, legal, environmental and human-related considerations" [PBFG07]. In this thesis, mainly the engineering of production systems is considered.

Equipment Equipment is a "single apparatus or set of devices or apparatuses, or the set of main devices of an installation, or all devices necessary to perform a specific task" [Int01].

Information Information is "data that have been organized or given structure - that is, placed in context - and thus endowed with meaning" [Gla91]. Information is somehow transported on a medium.

Material Material is "... matter from which a thing is ... made" [Oxf17d].

Matter Matter is a "physical substance ... which occupies space" [Oxf17e] and consists of "corpuscles tied together by their interactions" [Int11].

Memory In a memory information or data is stored [Web08].

Method A method is a systematic procedure with the aim to gain knowledge or to achieve practical results. What are the necessary steps to solve a problem? [Sch99] [Lin09]

Operation Operation (of a production system) is the functioning. The production system is producing products.

Physical asset "Physical assets usually refer to equipment, inventory and properties owned by the organization" [ISO14b]. In the context of this thesis, physical assets refer to production systems, components, and materials.

Preprocessing Preprocessing is a treatment of a physical asset using mainly mechanical or processing techniques (like shredding) with the intention to recycle the physical asset. [Her10] [VDI02] In this thesis, production systems, components, and materials are preprocessed.

Process A process is a "... set of interacting operations ... by which matter, energy or information is transformed, transported or stored" [Int15].

Quality According to [DIN05] quality is understood as the “degree to which a set of inherent characteristics fulfills requirements”. In this thesis it is applied to the technical characteristics of a physical asset.

Reconditioning Reconditioning is a treatment of a physical asset using mainly manufacturing process (like mechanical manufacturing steps or exchange of wear parts) with the intention to reuse the physical asset. Sometimes also: reprocessing. [Her10] [VDI02] In this thesis, production systems, components, and materials are reconditioned.

Recovery Recovery is the retrieval of the inherent value of a physical asset when it is no longer fulfilling the needs of the customer. Recovery involves concepts, like recycling and reuse. Recovery is the contrast of ordering a new physical asset. [LSsB06] [NC94] In this thesis, the recovery of production systems, components, and materials is considered.

Recovery process Recovery process is the process to recover physical assets. In this thesis, production systems, components, and materials are recovered.
Recovery process comprises all required activities from decommissioning, over disassembly, up to the final treatment (based on [Web08]).

Recycling Recycling of a physical asset comprises activities that recover it from the solid waste stream to generate out of it materials for the manufacturing of new physical assets [KM93].
[VDI02] differentiates the additional use of recovered materials, where they are used again for the same manufacturing processes (e.g. glass bottles used for bottles from recycled glass [JAA+93]) and where they are used for different manufacturing processes (e.g. glass windows used for bottles from recycled glass [JAA+93]). This differentiation is irrelevant for this thesis and, therefore, not applied.

Relocation With relocation the moving of a physical asset from one location to another is understood.

Remanufacturing Remanufacturing is not just restoring the production system or a component to their original specification instead it is about to modernize it to new specifications (in quality and functionality). For it, certain parts of the production system or component are kept, some are disposed and replaced by new ones. (Adapted from [RSI00], [KM93], and [LSsB06].)

Removal With removal the clearance of the production site is understood (based on [VDI16b]).

Repair Repair is the replacement of failed, badly worn or dysfunctional components or parts with the goal to restore the functionality of a system. The identity of the system is retained. [KM93]

Reuse Reuse of a physical asset "is the additional use ... after it is retired from a clearly defined duty" [KM93].
[VDI02] differentiates the additional use for the same intended use (e.g. bottle reused as refilled bottle [JAA+93]) or for a different intended use (e.g. milk bottle reused as flower vase [JAA+93]). This differentiation is irrelevant for this thesis and, therefore, not applied.
Reuse of information is the additional use of already created or gathered information.

Storage Storage means to stock physical assets, e.g. in a warehouse [Web08].

System "A System is an entirety of elements which interact in a way enabling the entirety to fulfill a defined aim. A system has a system border delimiting the entirety of system elements and its interaction from the rest of the environment." [SLR+15] In this thesis, this term is considered in the context of production systems.

Virtual representation Virtual means that a physical asset is "not physically existing as such but made by software to appear to do so" [Oxf17f]. A virtual representation of a production system is the model of the production system that does still not exist physically.

1 Introduction

This chapter provides the basic motivation of this thesis and derives research questions from that. The methodical approach, how those research questions are addressed, and the structure of the thesis are given additionally.

1.1 Motivation

The German plant engineering industry is faced with a decrease of incoming orders and an increased focus on mid-scale projects. For those, the customers demand more often modular plants with digital interfaces, with which small lot sizes can be flexible realized. Especially the orders of new plants in mature markets are stagnating. As a consequence, the service business is developed and expanded continuously on the one hand. Examples for such services are the exchange of components, spare part business as well as process optimizations and actions for the enhancement of the efficiency of operating plants. On the other hand the German plant engineering industry strives to offer customers a ’full package’. Examples are the installation and assembly management. [VDM17a]
But the German plant engineering industry is also affected by the global climate policy and must cope with more strict environmental protection laws to get the permissions to build plants. To save CO2, more energy supply systems are installed based on renewable energy or low-CO2 production technologies are used. [VDM17a] [Pla16b]
All in all, the German plant engineering industry is realigning concerning decrease of costs, process optimization, and development of new business models. Therefore, established workflows, methods, and processes are questioned. Industrie 4.0 is seen as an enabler to do so and as a chance to remain competitive. [VDM17a] [IEC15b]

According to [Pla16a] Industrie 4.0 is a new paradigm for the management and organization of the entire value chain along life cycles. But a life cycle is not considered isolated. Instead it is focused on the network of and linkage between the different life cycles, like the production system life cycle, the ones of technologies, components, or products. It is stated that this paradigm can be achieved by means of digitization. [Sen13] points out that for a successful accomplishment of Industrie 4.0 and an application of the new paradigm a comprehensive and intensive usage of software is necessary that interconnects the industry with its products and services.
Besides the Industrie 4.0 use cases, like ’Adaptable Factory’, ’Order-Controlled Production’, or ’Value-Based Services’, there is also the use case ’Circular Economy’ [Pla16a]. The scope of it is to keep machines, products, components, and materials in cycles [Pla16b]. That means the inherent value of machines, products, components and materials is kept and, thereby, energy and resources are saved as well as waste and CO2 emission are reduced [DSK+08] [Pla16a] [LSsB06] [Hub01a]. By that, the industry could make a contribution to the EU Action Plan [Eur17] to transform from today’s continuous economy to a circular economy.
Regarding a sustainable development the circular economy should be preferred to the continuous economy, since the latter one is striving to generate as much output

as possible from resources as input. The World Commission on Environment and Development defined sustainable development as a development that "meets the needs of the present without compromising the ability of future generations to meet their own needs" in 1987 [Hau87]. In contrast, at a certain point in time the continuous economy will run low before it runs out of resources.
The global demand of, e.g. metals, had increased by 87% up to more than 6 billion tons from 1980 until 2008 [Pan15]. Even though the market price for resources (such as copper, iron, and silver) is subject to fluctuations [VDM17a], resources are finite. According to [the17] copper will be globally depleted before 2045 and silver around 2035.
Some companies are facing the finiteness of resources and fluctuations of their prices by putting sustainability and a sustainable development on their agenda. Also to make them independent from (natural) resources suppliers. [Pla16a]
But also the ecological awareness of the people has been increased [VDM15] regarding their ecological footprint, i.e. the impact of the people on earth. And German companies can benefit from this demand for 'green' services and products [Fed14a]. And they did. They produced products, which consume less resources and produce less waste during product use. Those environmentally relevant innovations are mainly targeting the optimization of the product use phase [Hub01a]. Even though the use phase is essential, there exist other phases before and after [SL16], which could benefit from environmentally relevant innovations. Examples are the usage of appropriate materials (fair trade, long lasting, recyclable well) or the establishment of take-back systems.
This basically corresponds to the general life cycle definition with the analogous phases of 'birth', 'life', and 'death' and is typical - not only for products but also for production systems. Production systems play a central role in the sustainable development [Sch04], as they produce the products to be used by the customers. Compared with products, the same demands could be made for production systems. According to [SL16], the following three approaches can be pursued to get a 'green' production system:

- **Birth: considerations within engineering phase**
  Within the engineering phase, the later physical representation and operation of a production system is defined. Here, several optimization approaches respectively design guidelines are available, basically emerging from product design, see [PBFG07], [VDI02], [Nah13], or [PL11], targeting at e.g. CO2 emissions, energy consumption, production waste, or recyclability.

- **Life: considerations within operation phase**
  Within the operation phase a production system is producing products and potentially production waste (by-product), as it was defined in the engineering phase. For dealing with production waste, approaches can be found regarding reusing and recycling it for the same or different manufacturing processes, within the same company or across companies (called 'industrial ecology' or 'industrial symbiosis'), see [SPPR97], [PBFG07], [MCK$^+$10], [HLS16], [HMP$^+$10], or [WW10]. Thereby, the requirement on closing cycles emerging from sustainable economy can be met [Sch04].

- **Death: considerations within End-of-Life (EoL) phase**
  Devaluations bring a production system's operation phase to an end, when flexibility, adaptability, or reconfigurability (see [GXN09], [WRN15], or [CT05]) are

exhausted, or when manufacturing changes [KGR16] or modernization activities of the production system [DLH+10] are no longer implementable beneficially [SLH+16]. Then, an optimal EoL treatment or EoL scenario should be chosen to close the cycle, accordingly. It is usually strived to remain as much of the inherent value as possible [VDI02] [Fed16b].

Following a sustainable development and considering the finiteness of resources, the establishment of cycles is the next logical step and according to [Sch04] inevitable. That means that the life cycle phases should be interconnected appropriately. These interconnections can also be established across lives or life cycles [SL17b]. But what is a good cycle or interconnection for a production system when it had reached its end of life and how can this decision-making process be supported?

This thesis will focus on the EoL phase of production systems and, thereby, address the Industrie 4.0 use case 'Circular Economy', see [Pla16a] and [Pla16b]. The author thinks that the EoL phase can benefit from the Industrie 4.0 paradigm, because it is the last phase in the production system life cycle and, thereby, highly dependent on the prior phases regarding information. But a support is only profitable if all necessary information from the prior phases is provided in a digital form and is accessible for years. Today, the missing digitization of life cycle information needed in the EoL phase as well as the question of ownership of the information are main issues that hinder the EoL phase from being supported with specific methods or software tools. With Industrie 4.0, this can be changed.

Having this in mind, the next section will derive three research questions, on which this thesis is based on.

1.2 Research questions

When the life cycle of production systems is thematized in publications, mainly only the phases up to the operation phase are considered (see [AG12], [WSCL13], [FSD+16], or [Dra10]). But that what is coming after the operation phase - the EoL phase - is often neglected. A reason could be that the EoL phase is out of the scope of these publications and does not affect the results. That means publications can be barely found. And this makes it difficult to put a thesis on a sound basis. This results in the first research question:

RQ1: How is the EoL phase of production systems characterized?

With this question involved stakeholders and activities done in this phase are identified, so that a holistic view can be created. This thesis will only focus on production systems from the manufacturing industry.
After having understood and characterized the EoL phase, it can be analyzed how this last phase of the production system life cycle can be supported. This results in the second research question:

RQ2: How can the decision making-process be supported, so that a proper EoL treatment for a production system can be found and assessed?

To answer this question possible EoL treatments and characterization approaches to make them comparable should be identified. Thereby, the decision-making process requires a holistic, generalized, and life cycle thinking to support it systematically. Even though the development of a supportive software tool is out of scope of this thesis, the results of this thesis should be the basis for such a software based realization. This leads to the next and last research question:

RQ3: Which types of information are required to enable or support this decision-making process and how can it be made available?

On the one hand this question should answer the question, which types of information need to be provided by the previous life cycle phases to enable or support the decision-making process, since the performance of the EoL phase and this process are highly dependent on it. On the other hand, it should answer the question, which types of information need to be gathered during EoL phase to perform it well. How this information can be made available and transferred is the second part of this research question that is answered in this thesis.
This life cycle consideration (including the EoL phase) could benefit the Industrie 4.0 related discussions on necessary describing information of physical assets and could potentially result in adaptations of existing software tools of each life cycle phase (as they are the information providers) to do so. By that, the results of this thesis could be one step towards the circular economy.

The next section will describe the methodical approach applied to tackle the research questions.

1.3 Methodical approach

The bottom line of the research questions is to understand and describe the EoL phase of production systems in its entirety including the possible EoL treatments as well as their recovery processes. With recovery the retrieval of the inherent value of a physical asset is understood when it is no longer fulfilling the needs of the customer. Recovery involves concepts, like recycling and reuse. To enable this holistic view, the qualitative research method is chosen. This is usually applied when there is no or just few insights or knowledge on a topic. It comprises the inductive scientific approach, which is the generalization from the particular. [SR16] [Bla15]
The first activity to capture the state of the art is a literature review. For this, several publications were read comprising PhD theses, papers, books, reports, EU projects, standards, and guidelines. Those were found through GoogleScholar, IEEExplore, Springer link, ScienceDirect, CORDIS, and through university web pages. However, these publications were focusing on products, because there is barely literature found that is especially focusing on production systems (i.e. production system from manufacturing industry). But due to the fact that a production system (e.g. a packaging system) could be generally considered as a product (e.g. washing machine), it was identified as a viable approach.
The second activity that should complement the literature review is a case study analysis. For this, several experts were interviewed that do the recovery process of production systems as their daily business.

Based on both activities the author is able to generate a holistic view on the EoL phase (including possible EoL treatments or EoL scenarios) and is, thereby, able to identify how the EoL phase could be supported. For this, a recovery planning method (RPM) is developed that should support decision makers in the early phase of the recovery process to decide on a proper EoL treatment for a particular production system, i.e. which physical assets should be recovered. And by additionally considering the exchange of the recovered physical assets with their information and data between the EoL phase and the other phases of the production system life cycle, the author is able to highlight the impact and relevance of the usually forgotten EoL phase of production systems.

The next section will provide the structure of this thesis presenting how the research questions will be answered.

1.4 Thesis structure

This thesis is structured into three parts as visualized in Figure 1.1. The first part analyzes and evaluates the state of the art. The second part uses the conclusions and limitations of the state of the art to develop a method to support the EoL phase in finding a proper EoL treatment or EoL scenario. Due to the inductive scientific approach of this thesis, there is no use case specific application of this method, since the pursuit of the author is the generalization. Part three discusses the developed method regarding its applicability, benefits, and drawbacks.

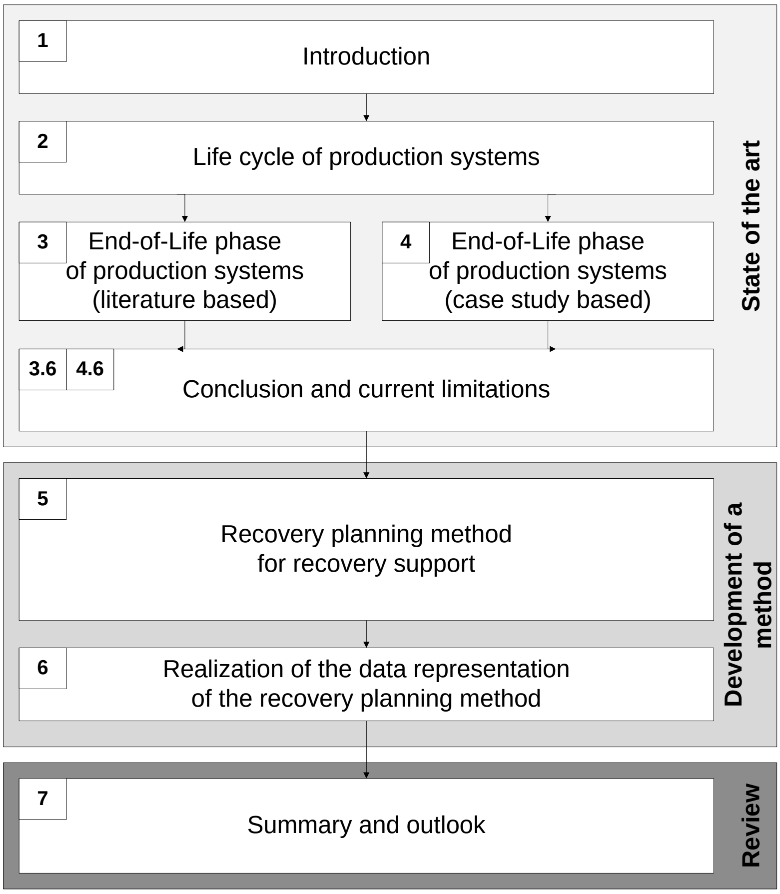

Figure 1.1: Thesis structure

After having introduced this thesis, Chapter 2 will provide a general overview of the life cycle of production systems found in the discrete manufacturing industry. This chapter will also give a definition of a production system as well as how its life cycle phases are characterized and interconnected with each other (called recovery possibilities). This is the basic understanding for this thesis.

Chapter 3 will provide the literature based analysis of the state of the art regarding the EoL phase of production systems. In detail the recovery possibilities will be examined. Another result of this chapter will be the EoL data model. Step by step the model will be developed by analyzing several publications and extracting requirements and the corresponding information from them. Those will be, then, transferred to an UML class diagram - also step by step and comprehensible.
Also, methods that could be used to support the decision-making process on a proper EoL scenario for production system will be analyzed and compared. Advantages and limitations will be discussed.
With an analysis of software and IT support this chapter will end. Here, the paradigm Industrie 4.0 will be explained. Also possible software tools (emerging from previously presented methods) and a software system architecture extended by Industrie 4.0 considerations will be presented. The latter also includes the discussion of possible data transfer mechanisms.

Chapter 4 will complement the literature based analysis from Chapter 3. As already mentioned, there are barely publications about the EoL phase of production systems available. That is why thematically close publications are chosen and adapted to the EoL of production systems. As a counterpart for the theoretical analysis, a case study based approach will be chosen. For this, the author had interviewed several experts, who are involved in the EoL phase of production systems with their daily work. This will result in a characterization of the EoL phase from the practical point of view, an overview of reasons why a production system reaches its end of life, an overview and description of the business ecosystem for the EoL phase that shows the involved stakeholders, and the legal basis for recovery in Germany.

Chapter 3 as well as Chapter 4 will end with a conclusion and an identification of current limitations. These are the basis for Chapter 5, in which the method will be developed that intends to support the EoL phase in deciding on a proper EoL treatment or EoL scenario for a certain production system. Besides the development and description of the recovery planning method (RPM), the EoL data model from Chapter 3 will be considered again to get unified and generalized. This model is the data based foundation of the RPM, which stores all necessary information.

In Chapter 6 this EoL data model will be mapped onto a data exchange format, since this transfer mechanism was chosen in this thesis to be applicable. In particular, the data exchange format AutomationML was chosen due to the expertise of the author in this standard.

With a summary and outlook in Chapter 7 this thesis will end. Additionally, an intense discussion about applicability, benefits, and drawbacks will be provided.

2 Life cycle of production systems

Even though, this thesis focuses on the last phase - the EoL phase - with its interconnections, this chapter introduces, in general, production systems and the production system life cycle with its phases. It also shows how all these phases can be interconnected with each other and what this means. By that, this chapter provides the basic understanding for this thesis.

2.1 Production system

A *production system* is a system with a production function or process that converts incoming material, energy, and signals into outgoing material, energy, and signals [PBFG07]. With this, production systems finally produce the products, which are demanded by the market and customers.

The system theory defines an abstract model of a system consisting of interacting elements which are separated from the system environment:
A *system* is defined as an entirety of elements which interact to enable the system to fulfill a defined purpose it is set up or used for. In addition, the system itself is (per definition) embedded in a *system environment*. The system and the system environment can establish together a kind of upper system they belong to as well. It is a matter of the point of view.
Within the system, a *system element* is an identifiable part of a system enabling or supporting the overall system behavior, so that the system can fulfill its purpose. A system element cannot be divided into further parts - They are considered as atomic.
Interaction between system elements is the process of mutual influencing the internal element behavior by using the element interfaces.
A *system function* is considered as the entirety of interactions of system elements exposed to the environment by a single element of the systems' interface. It may have various characteristics ranging from mathematical functions over physical and logical dependencies to common behavior.
Technical systems, such as production systems, are usually too complex to consider and engineer them only as one set of elements to fulfill a certain purpose. Thereby, a hierarchy is needed, which can be established by the term of a subsystem. A *subsystem* is a system itself and is often denoted as *component*. Besides, it is part of and fully contained by the upper system. In the scope of the upper system, the subsystem is considered as system element. A *system level* is, then, an abstraction for a set of elements considered on the same level of the system hierarchy. A schematic representation of a system can be found in Figure 2.1 and in the corresponding UML representation in Figure 2.2.
[LRS+14] [SLR+15]

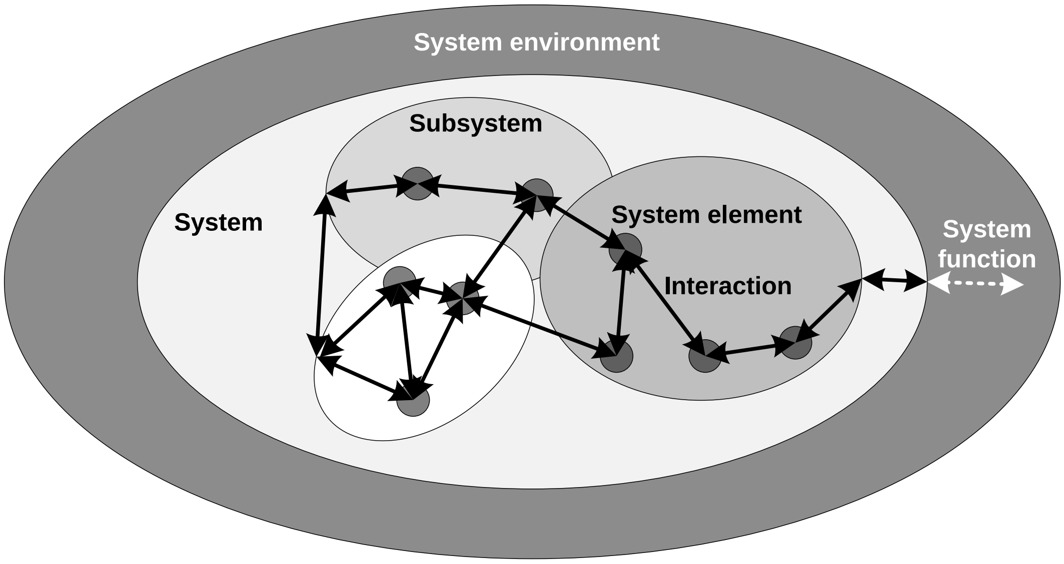

Figure 2.1: Schematic representation of the structure of a system

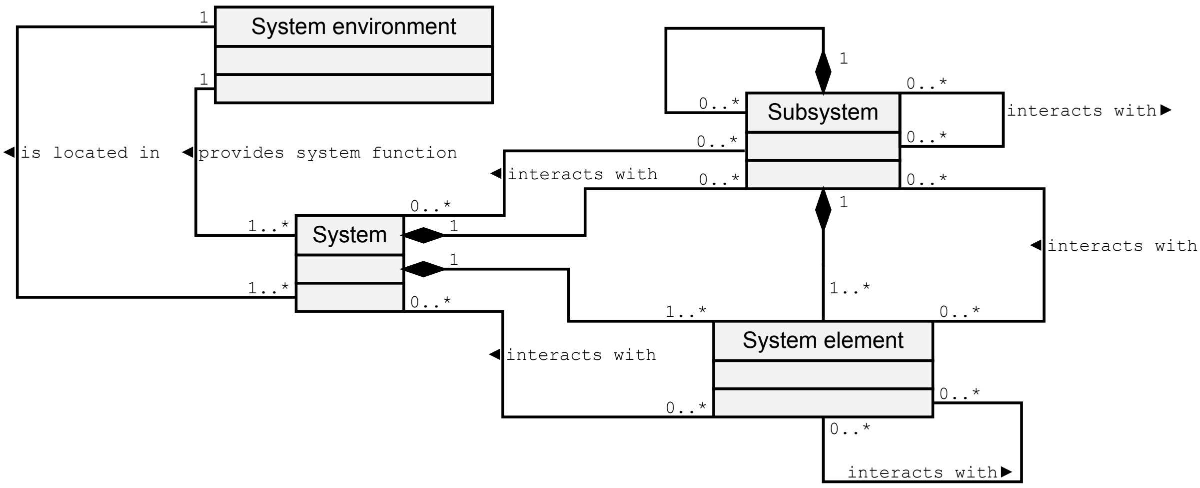

Figure 2.2: UML representation of the structure of a system

Within this thesis, a production system is understood as a system which is located in a system environment, the factory. A *factory* combines resources (e.g. production systems), personnel, buildings, material, capital, qualifications, and knowledge [WRN15]. A production system, which produces a product, can contain other subsystems - further called components - which produce intermediate goods or provide supportive functions. In this thesis there will be no distinction made between subsystem and system element. All of those system levels are considered as components. How many of such components or production system levels exist, depends on the industry sector and point of view. Additionally, each level has usually its own, industry dependent naming, like production line and work station [LSH^{+}17b] or assembly groups and process technologies [Gep14]. In the course of this thesis, materials will play an important role as well. Thus, each component consists of *materials*. The adaptation of a generic system term to the scope of this thesis is done in Figure 2.3. All of them are physical assets.

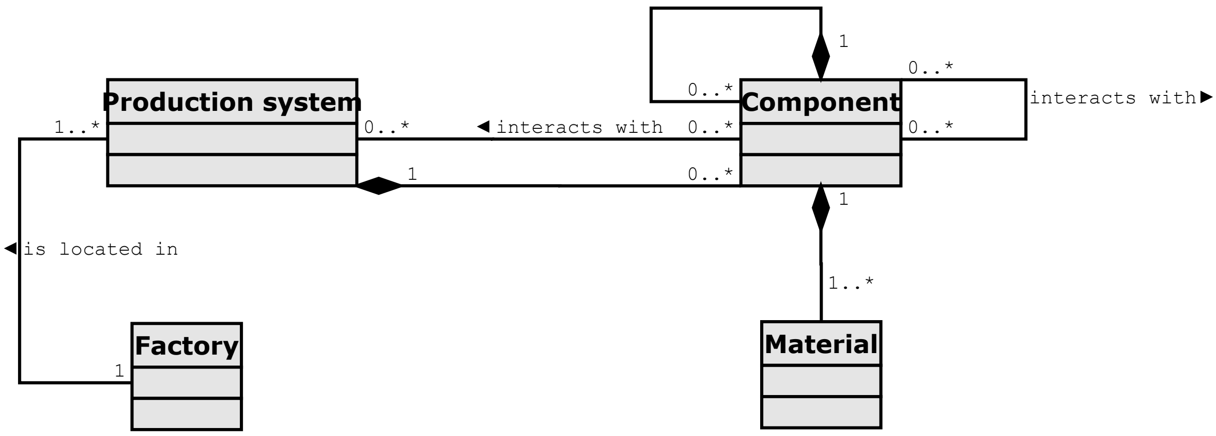

Figure 2.3: UML representation of the structure of a production system

Due to the author's expertise, this thesis considers only production systems of the discrete manufacturing industry. The following section will analyze the life cycle of such a production system.

2.2 Life cycle of production systems

The term 'life cycle' is a widely used term almost existing in every field of work. It always needs to come with a prefix describing the life cycle of what is meant, like the factory life cycle, production system life cycle, product life cycle, order life cycle, technology life cycle etc. But as common the term 'life cycle' is, the more definition variations exist, e.g. [AG12] or [WSCL13]. It appears that only those phases of a life cycle are considered by authors that are relevant for their current publication. In some publications the life cycle ends with the use/operation phase. Since this thesis investigates the EoL phase or the end of a production system a holistic consideration is necessary.
According to electropedia [Int13] a life cycle comprises, in general, "consecutive and interlinked stages of a product system, from raw material acquisition or generation from natural resources to the final disposal". In this definition the life cycle starts with the earth (mining natural resources from it) and ends with the earth (disposing on a landfill onto it). This is on the one hand the holistic consideration this thesis needs regarding the circular economy. And on the other hand this definition also focuses on the physical assets that are in the scope of this thesis.
As a production system can also be seen as a system of products or components, this definition will be applied. The life cycle with its stages or phases, as considered in this thesis, is depicted in Figure 2.4 - with the earth with its resources as source and sink to get a closed loop.

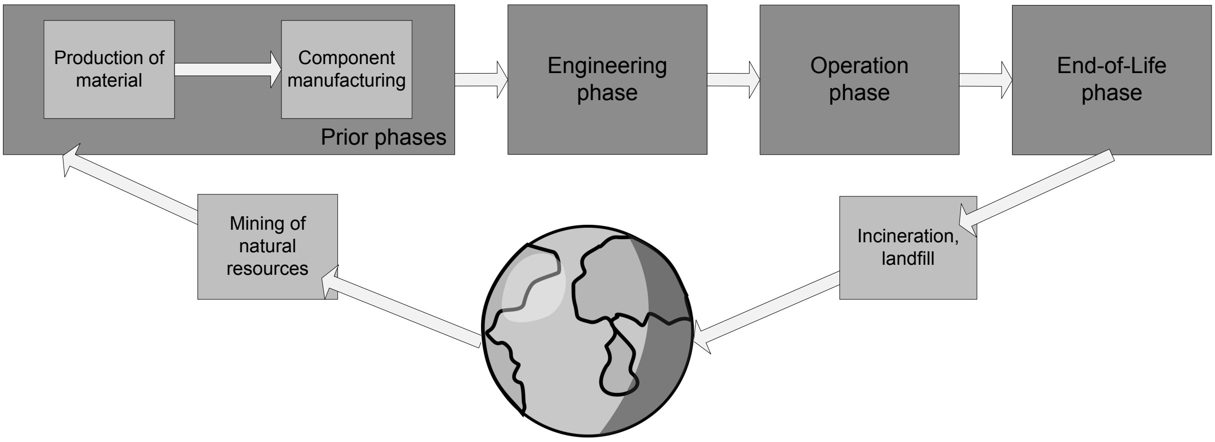

Figure 2.4: Life cycle of a production system

The life cycle of a production system begins with mining natural resources. These are processed to materials that are used to manufacture single components from it. Both phases are summarized and further considered as 'prior phases'. Because in this thesis the engineering phase is considered as 'birth' of the production system and the prior phases or the result of them are needed to execute the engineering phase, where the components are aggregated to the production system, which can, then, produce the products for the market (operation phase 'life'). Due to requirement changes, like trends or technological progress [SBK01], the operation phase of a production system can end and it is removed to make room for a new production system, which meets the changed requirements ('death' of the production system). With the incineration and/or disposal of the production system the cycle is closed [Web08].
All in all, it can be boiled down to, generally speaking, a production system engineering phase, that needs certain inputs from prior phases, an operations phase, and an EoL phase. These three phases are described in the following subsections in more detail.

2.2.1 Engineering phase

The engineering phase comprises the set of activities necessary to engineer production systems. This is often called engineering process.
In literature, there are many of such engineering processes described, e.g. [PBFG07], [VDI04], [Lin09], [Foe13], [LFH+11], [Hun12], or [LSR+13].
According to [LSR+13], at first, the initial situation is analyzed and the target situation is defined, which results in requirements on the production system. Additionally, a possible draft solution structure is developed that fulfills the intended functionality of the overall production system (analysis phase). Afterwards, the draft solution structure is further defined with concrete components providing the system functionality. In a next step, the results are refined, i.e. the production system is developed completely, so that all information necessary to build the production system is available. This includes the aspects of mechanical engineering, electrical engineering, HMI development, robot and PLC programming, and virtual commissioning among others.
Subsequently, individual production system components are installed at the customer's location and are tested for their correct functionality. During the following commissioning each component is integrated into the overall production system, the production system is built up at its intended location, all control functions are tested and modified if needed,

and the documentation is completed.
In practice, this process is not strictly sequential. Some activities can be overlapped or done in parallel.

The engineering phase gets its inputs, as mentioned before, from the material and/or components suppliers (prior phases).

2.2.2 Operation phase

The engineering phase of a production system ends with the installation, commissioning, and the ramp-up of the production system. With the ramp-up it is a smooth transition to the operation phase of the production system.
The operation phase is an important phase since it represents the useful life of a production system where actually products are produced. Due to operation wear and tear are caused, which can trigger maintenance activities. Also conversions, caused e.g. by changed requirements on the production system, can occur. Usually the conversion is comparable with the engineering phase with the difference: The engineered solution must be integrable into the existing production system. [WRN15]

2.2.3 End-of-Life phase

The operation phase of a production system can come to an end or reaches the EoL phase when it can no longer fulfill its indented purpose or required function, in particular when it can no longer fulfill the needs of the owner or operator [LSsB06]. When maintenance and repair efforts as well as conversion efforts are not useful anymore [WRN15] or an operation phase extension is not wanted, the production system has to be decommissioned, disassembled, removed, reused, recycled, or disposed [Web08], i.e. the life of a production system or its function ends at a certain geographically location because it is no longer needed at this place. Thereby, it is made room for another production system. This can be, then, engineered to meet the changed requirements.
At this point the production system turns into waste for the production system owner or operator. This is in line also with the most general definition for waste given by the German law on waste called KrWG [Fed16b]. It says that objects become waste when the owner gets rid of them, wants to get rid of them, or needs to get rid of them. This happens when the original purpose is no longer fulfilled or is no longer required. [RSI00]
The reasons for a production system losing its purpose or its function could be the same than the general change drivers described in [NRA08] and [WRN15]. Changes are disturbances, influencing the production system, that are caused by external and internal drivers. External drivers can be related to world economy, environment, politics, society, or technology. Internal drivers are mainly caused by a change in the corporate strategy, e.g. due to a change in the management or ownership or due to the wish to enter new markets. [NRA08] [WRN15]
In the EoL phase it is decided about recovery options or the EoL treatment [KM93].

Of course, the production system life cycle is interconnected with other life cycles [Pla16a], e.g. with the product life cycle, since products are produced by production systems, the component life cycle, since components are used to build the production system, or the technology life cycle, since technology is used to produce the products with the production system. However, this thesis will only analyze the production system life cycle, but will consider relationships to the other life cycles.

After having described each relevant life cycle phase, the next section will analyze the interconnections or recovery possibilities among those life cycle phases.

2.3 Recovery possibilities within the life cycle of production systems

Recovery possibilities represent artifacts, i.e. information and/or physical assets, which are valuable enough to be gathered and used afterward to enhance the life cycle of production systems in some way.

Since the market or customer regularly demands new and individualized products and is also willing to replace products in less time [BtHVH14] [WLH10] [IEC15b], for example, the product life cycle is reduced. Due to the interconnection of it with the production system life cycle, its configuration or setup is also reduced [DLH+10]. To cope with this reduced life cycle, two general approaches can be named (according to [SL17b] and [Sen09]):

- **Optimize one life cycle phase**
  For example, when shortening the engineering phase, the production system can produce the products earlier. In contrary: When the production system is engineered more flexible and adaptable, the production system may also be able to produce the new products. This causes an extension of the operation phase, because different production system configurations are possible. But when this lifetime extension is not possible, the production system reaches its end of life. By enabling a quick removal of the production system, to make room for a new production system capable to meet the changed demands of the market, the EoL phase is reduced.
- **Optimize the interconnection of life cycle phases (within a life as well as across lives)**
  For example, when engineering information about the installation is available, it makes maintenance and repair during the subsequent operation phase easier. When information about the current state of wear and tear of the production system is available, the subsequent recovery process (EoL phase) becomes easier. But those interconnections can also be established across lives. For example, when information about errors, malfunctions or recycling issues of the production system is considered in the engineering phase of the subsequent production system (second cycle), operation and EoL phase of the second cycle could benefit from that. All those interconnections can support a production system life cycle reduction.

The shorter the life of a production system becomes, on the one hand, the more interconnected the phases become (it is likely). On the other hand, the more beneficial it is to find proper interconnections between the life cycle phases [Sen09] of a production system. A cycle is established, when there is information or a physical asset worth

being used again to benefit the life cycle of the production system. These are called recovery possibilities in this thesis. [SL17b]
Figure 2.5 gives an overview of recovery possibilities in and among the life cycle phases - based on [LSH+17b]. This is no exhaustive representation of recovery possibilities. It is assumed that all artifacts from the prior phases are aggregated in the engineering phase of a production system and are propagated to the EoL phase, even though, those artifacts from the prior phases may be also used in the operation phase, like operation fluids or spare parts for repair.

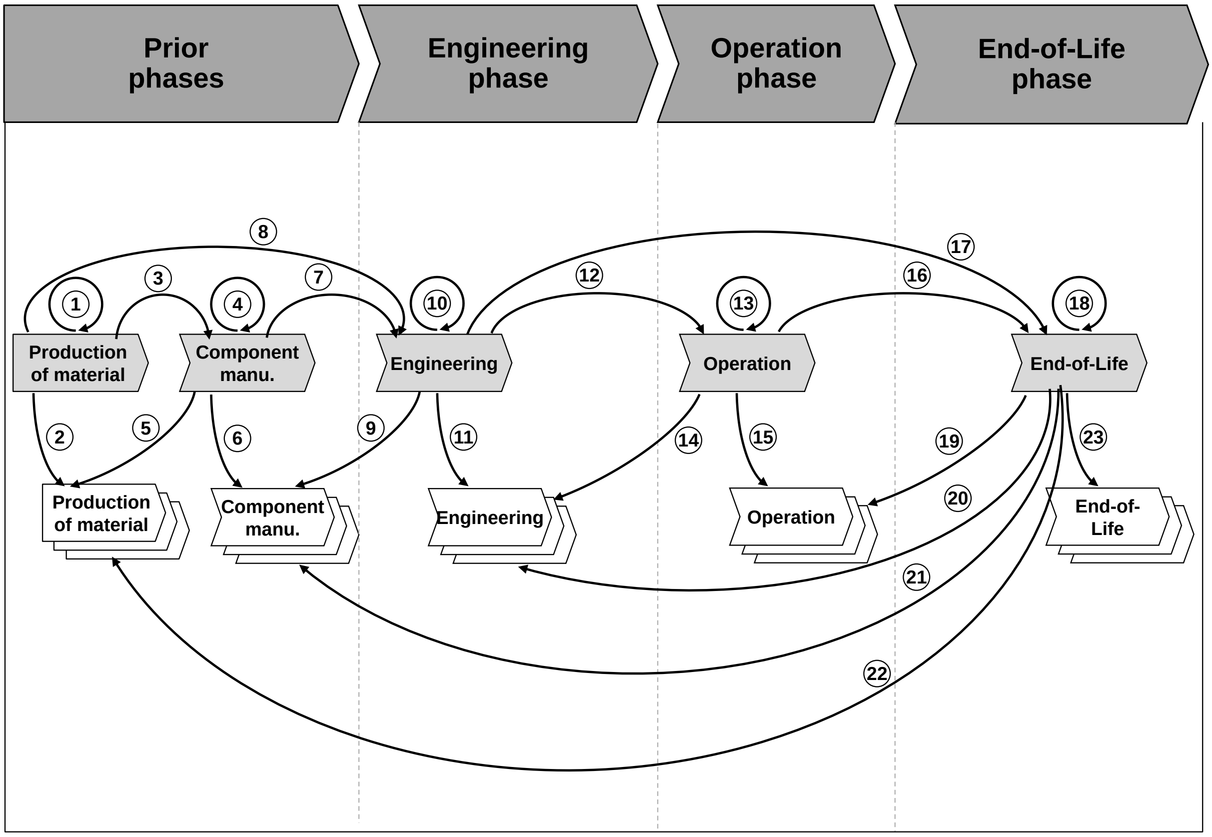

Figure 2.5: Recovery possibilities within the production system life cycle (based on [LSH+17b])

There are three types of recovery possibilities:

1. Recovery among life cycle phases (feed forward): An artifact created in one phase has an effect on another artifact of another, subsequent phase - but of the same cycle (case 3, 7, 8, 12, 16, and 17 in Figure 2.5).

2. Recovery within the same life cycle phase (feed within): An artifact created in one phase has an effect on another artifact of the same phase (case 1, 4, 10, 13, and 18 in Figure 2.5).

3. Recovery among life cycle phases (feed back): An artifact created in one phase has an effect on another artifact of a phase in a subsequent cycle (case 2, 5, 6, 9, 11, 14, 15, 19, 20, 21, 22, and 23 in Figure 2.5). Technically speaking, this is also a feed forward from a temporal point of view. But in this thesis it is considered as feedback.

This thesis will only focus on the EoL phase of production systems and, thus, only consider those recovery possibilities related to this phase. For the others see [LSH+17c]. This reduces the recovery possibilities as follows and shown in Figure 2.6:

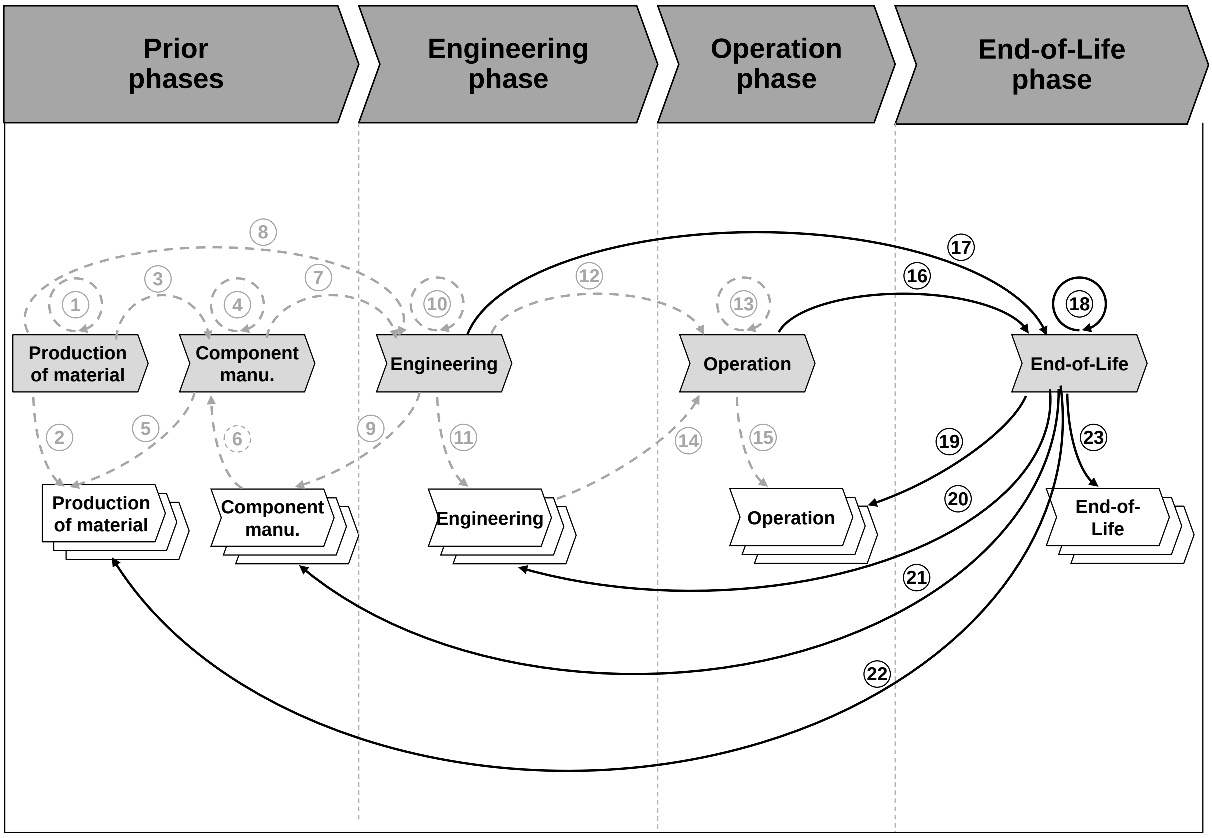

Figure 2.6: Recovery possibilities related to End-of-Life phase in black (based on [LSH+17b])

Case 16 and 17 (feed forward) represent artifacts that are needed to execute the EoL phase properly, like the connection structure of the production system or the state of wear and tear of a production system. Case 18 (feed within) represents, e.g. certain knowledge that is gathered during EoL that helps to enhance the current EoL phase, or the cost and time of a disassembly activity. Case 19, 20, 21, 22, and 23 (feed back) represent artifacts that are created in the EoL phase and are used in the life cycle of another production system, like a certain component as spare part. The next chapter will, therefore, analyze those recovery possibilities related to the EoL phase in more detail.

3 End-of-Life phase of production systems (literature based)

This chapter analyzes the state of the art of the EoL phase of production systems from a literature based point of view. The recovery possibilities, which emerge from the EoL phase are discussed. This implies a consideration of the recovery possibilities including information, which is necessary for them, as well as methods, which support the recovery in the EoL phase. Software and IT support in the context of EoL are also discussed.

3.1 Methodical approach

The state of the art was captured by having done a literature review. Several publications were read comprising PhD theses, papers, books, reports, EU projects, standards, and guidelines. Those were found through GoogleScholar, IEEExplore, Springer link, ScienceDirect, CORDIS, and through university web pages. In the following the main search keywords are listed (comprising English ones as well as German ones): recycling (of products/production systems); EOL/End-of-Life (of products/production systems); disassembly (also in German: Demontage); factory life cycle (also in German: Fabriklebenszyklus); product/production system life cycle; life cycle costs; life cycle assessment, recycling information, recycling model, sustainability information, sustainability model, product/production system recovery; product/component/production system reuse; sustainability of products, EOL/End-of-Life support, EOL/End-of-Life information model, EOL/End-of-Life data exchange, material recovery, component recovery, recycling systems, disassembly planning, recycling planning, and circular economy (also in German: Kreislaufwirtschaft).
Publications referring to products were considered, because there was barely literature found that is especially focusing on production systems (i.e. production system from manufacturing industry). Due to the fact that a production system (e.g. a packaging system) can be also seen as a product (e.g. washing machine) (question of perspective), it was identified as a viable approach. Of course, there are differences between both in uniqueness and the number of instances, existing distribution and reverse logistics networks/systems, size, purchase and operating costs, duration of life cycle, technical complexity, frequency of reconfiguration, and, therefore, different requirements on design [Tit16] [LMF16]. Even though, the approach remains viable.
Publications about energy generation (like nuclear, wind, solar, or water power plants) or production systems from process industry (producing gas and oil or chemicals) were explicitly excluded. Instead this thesis is focusing on production systems from the manufacturing industry. [MBS$^+$11] shows the differences between the manufacturing and process industry. For example, that the latter one engineers production systems that intend to operate 40-50 years while modernization activities are common. The manufacturing industry on the other hand does not typically have such long operating production systems, i.e. they reach more often their end of life, e.g. with a product change.

The next section will describe the EoL related recovery possibilities from Chapter 2 in more detail.

3.2 Recovery possibilities

As mentioned before, a recovery possibility can be information or a physical asset, but in general an artifact [LSH+17b]. In this thesis it is focused on the physical assets, which emerge from the EoL phase, not just on (isolated) information. The recovery of 'pure' information [WSCL13], e.g. experience gained during the disassembly of a production system, is not considered (case 23). However, each physical asset is always considered together with its digital description or representation. All in all, the cases 19, 20, 21, and 22 - depicted in Figure 2.6 - are considered.

3.2.1 Recovery goals

Except for some certain physical assets (batteries, cars, etc.) with specific restrictions on the recovery, there are usually more general terms used to express the recovery of physical assets.
[PBFG07] mentions production waste recycling, product recycling, and used material recycling. [DSK+08] names the recovery of products, components, parts, and material. [NTW09] names products, modules, components, and material. [SBK01] mentions products, components, and material. [VDI02] and [RSI00] have made the same distinction. [Pla16a] names besides products, components, and material also machinery that can be recovered. This publication also considers the product producing machines or production systems.

A general consideration of the physical asset and what of it can be possibly recovered follows the system theory approach presented in Chapter 2. Based on the literature and in adaptation of the term product to production system, this thesis will consider the physical assets *production system*, *component*, and *material*. Those are the physical assets, which can be recovered from a production system at its end of life (feedback cycle). This is also in accordance with the structure of a production system presented as a UML class diagram in Figure 2.3.
All in all, the physical asset can also be seen as the *recovery goal*, since it describes 'what' is recovered from the production system. A description for the three recovery goals follows:

EoL3 - production system recovery *Production system recovery* comprises the reuse of the entire used production system. It is disassembled out of the actual environment while its shape is retrained. The function of the production system might have changed due to its operation. In this thesis it is called EoL3.

EoL2 - component recovery *Component recovery* comprises the reuse of used components of the production system, e.g. as spare parts [Sch05]. As the life span of those might be longer than the life span of the production system itself [Hub01a] [WRN15]. The components are disassembled out of the actual production system - The shape of the production system is dissolved - while the shape of the components is retained. The function of the components might have changed due to their operation. In this thesis it is called EoL2.

EoL1 - material recovery *Material recovery* comprises the recycling of material, the components are made of, by disassembling the components out of the actual production system and dissolving their shape [DSK$^+$08] [PBFG07]. The function of the material might have changed due to its operation. In this thesis it is called EoL1.

EoL0 - no recovery *No recovery* means that the production system is send to the landfill where it is dumped. Of course, the waste could be incinerated and energy could be recovered. [RSI00] But this is out of the scope of this thesis. It is just mentioned for the sake of completeness. It is called EoL0.

According to [VDM15] the demand for incineration capacities is decreasing, i.e. recovery is increasing. Also the existence of several market places for selling industrial goods, like eBay [eba17], asset orb [Ass17], or GoIndustry DoveBid [Dov17] shows that it is not only material recovery that decreases the aforementioned incineration capacities, but also component and potentially production system recovery.

To recover a physical asset several processes, in this thesis called *End-of-Life strategies*, are applied. Those are also taking the actual functionality of the production system into account as well as the future application/purpose of it. Because an EoL strategy prepares a physical asset for its future usage, which is, then, called *EoL scenario*. Figure 3.1 depicts this.
The next section is dedicated to the different EoL strategies for production systems.

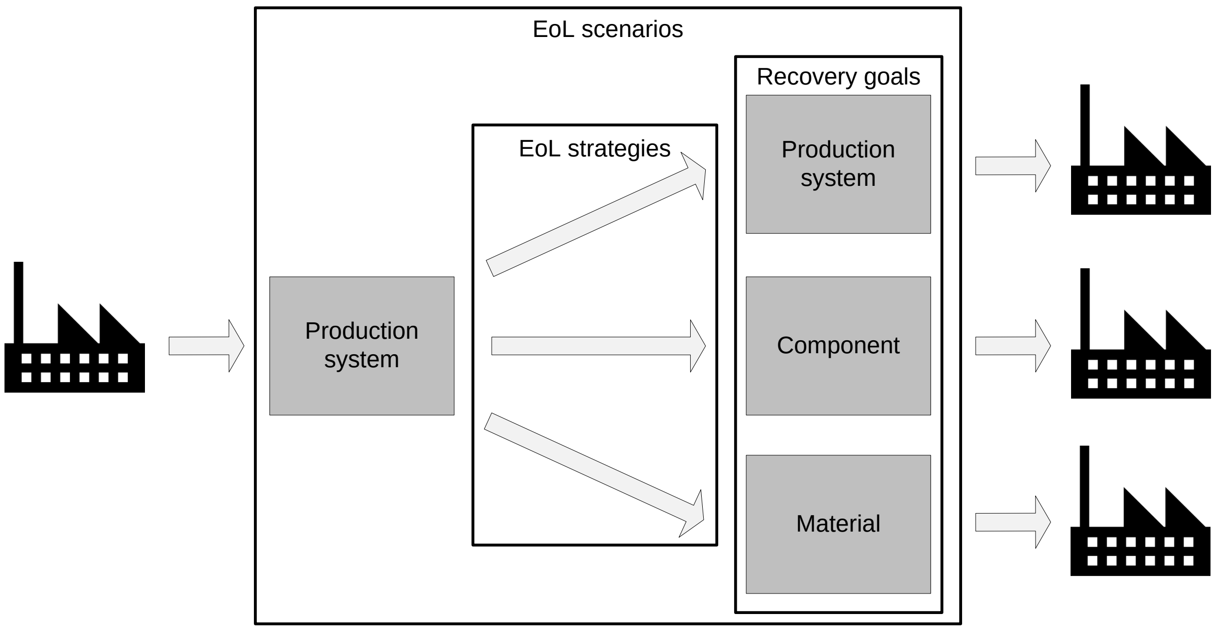

Figure 3.1: Meaning of End-of-Life scenarios

3.2.2 End-of-Life strategies

To successfully execute a certain recovery, different processes are needed - depending on the future application/purpose - that enable the recovery of a physical asset from a production system. EoL strategies describe 'how' it is recovered.

EoL strategies can be seen as treatment options for a production system, component, or material. They often point to the process rather than explicitly mention the object to be recovered. A vast number of EoL strategies can be found in literature - also with varying definitions. Due to the little number of appropriate publications regarding production systems, also similar publications regarding the EoL of products were analyzed. The following shall give an overview.
[DSK+08] names maintenance and repair as processes to extend the life of a product at its actual location. Thus, no transportation (reverse logistics) is needed. In contrast, product reuse, product upgrade, product downgrade, product remanufacturing, component reuse, and material recycling require transportation as well as incineration and landfill. [KIK08] mentions, besides disposal, reuse, recycling, and natural recycling for products. [VDI02] names maintenance and refurbishment as treatments for reuse. Additionally, the treatments material and energetic recycling are named. [SBK01] identifies treatments (called: adaptation processes), which lead to a product's reuse. These are: maintenance, repair, remanufacturing, upgrading/downgrading, enlargement/reduction, rearrangement, and modernization. [Pla16b] lists reuse, repair, refurbishment, remanufacturing, recycling (downcycling), and upcycling as possible treatments. [SSZ+14] names for the post-use stage of a product the following treatments: reuse, recycle, remanufacture, and landfill. Also [MT02] proposes those four treatments. [RSI00] lists reuse, service, remanufacture, recycle (separate first), recycle (shred first), and disposal. [SWM10] groups possible treatments into categories: plant improvement (e.g. updating, upgrading, retrofitting, modernization), maintenance (e.g. repair, overhauling), reuse (e.g. sale of used parts), reconditioning (e.g. revamping, conversion), and disposal (e.g. recycling of materials). [Pla16a] names remanufacturing, reuse, repair, and upcycling. [Pla16b] extends this by refurbishment and recycling (downcycling). [LSsB06] finds out, that those treatments are differently understood in academia and industry. Several definitions were identified for: reuse, remanufacturing, reconditioning, refurbishment, component cannibalization, and material recycling. [PBFG07] names reuse, reconstitution, disposal, and final storage. Additionally, it differentiates reuse: reuse for different function and reuse for same function. Also for recycling: recycling for different application and recycling for same application. For recovering material, preprocessing processes are required. For reuse, reconditioning processes. This kind of classification is also proposed by [VDI02], i.e. the future application is essential to decide for a proper treatment of the production system at the end of its life.

As mentioned before, EoL strategies are also taking the actual functionality of the production system/component/material into account as well as the requirements on the future application of it. This is used to decide, which the relevant ones are:

1. The actual production system/component **exceeds** the requirements on the future application.
    - A *direct reuse* of the production system/component is possible. But it is decreased in its value since not all of its functionality is used.

2. The actual production system/component **matches** the requirements on the future application.
   - A *direct reuse* of the production system/component is possible. It keeps its value since all of its functionality is used.
   - In case that the usage has changed the production system/component, a *reuse after repair* is possible. The value is restored but basically kept.
3. The actual production system/component **deceeds** the requirements on the future application.
   - A *reuse after remanufacturing* of the production system/component is possible. The value is increased.
   - A *recycling* of the material is possible. The value is decreased.
4. The actual production system/component shall be disposed. All value is lost.

[LSsB06] states that the terminology in academia and industry for EoL and recovery related terms is not consistent. The author can confirm this. In this thesis, direct reuse, reuse after repair, reuse after remanufacturing, and recycling are identified as relevant. For the sake of completeness also the EoL strategy disposing is considered. For a common understanding, all of the EoL strategies are described in the following.

Direct reuse *Direct reuse* is the additional use of a production system or component after the production system has reached its EoL, i.e. the current life of a production system or component has ended but it is considered for another life. A production system or component is recovered without having any improvement processes executed before reusing it.
In this thesis it is called EoLxa. EoL3a (for production system recovery) or EoL2a (for component recovery) is possible.

Reuse after repair *Reuse after repair* is the additional use of a production system or component after the production system has reached its EoL but with the necessity to replace failed or badly worn part. The identity of the production system or component is retained. [KM93]
In this thesis it is called EoLxb. EoL3b (for production system recovery) or EoL2b (for component recovery) is possible.

Reuse after remanufacturing *Reuse after remanufacturing* describes also the additional use of a production system or component after the production system has reached its EoL - by not just restoring it to its original specification but by modernizing it to new specifications (in quality and functionality). For it, certain parts of the production system or component are kept, some are disposed and replaced by new ones. (Adapted from [RSI00], [KM93], and [LSsB06].)
In this thesis it is called EoLxc. EoL3c (for production system recovery) or EoL2c (for component recovery) is possible.

Recycling *Recycling* is the additional use of materials after the production system has reached its EoL. For it, processes are used that dissolve the shape of a production system or component to recover the materials. Since the recycling is directly assigned to material recovery (see Subsection 3.2.1), no further distinction is made.

Disposal In this case any inherent value of the production system, its components, and materials gets lost. The production system is send to the landfill where it is dumped. Of course, the waste could be incinerated and energy could be recovered. [RSI00] But this is out of the scope of this thesis. It is just mentioned for the sake of completeness. Since the disposal is directly assigned to no recovery (see Subsection 3.2.1), no further distinction is made.

A recovery goal together with an EoL strategy equals to an *EoL scenario*. How the EoL scenarios are located in the life cycle of production systems this is described in the next section.

3.2.3 End-of-Life scenarios

All possible EoL scenarios, which result from the combination of recovery goal with EoL strategy, are depicted as arrows in Figure 3.2. Here, the number indicates the recovery goal and the letter the EoL strategy.
Each one of the upper arrows represents a loop closing the life cycle of a production system. Even though, just disposing or incinerating the production system represents a valid loop, all inherent value of the production system, its components and material, and know-how is lost, because all phases (production of material, component manufacturing etc.) has to be done a second time [Hub01a]. It is of economic and environmental interest to choose a better loop, to keep the loop small, to recover the inherent value [VDM15][Hub01a][DSK^{+}08]. EoL scenarios can also be seen as different possible treatments of a production system at the end of its life.

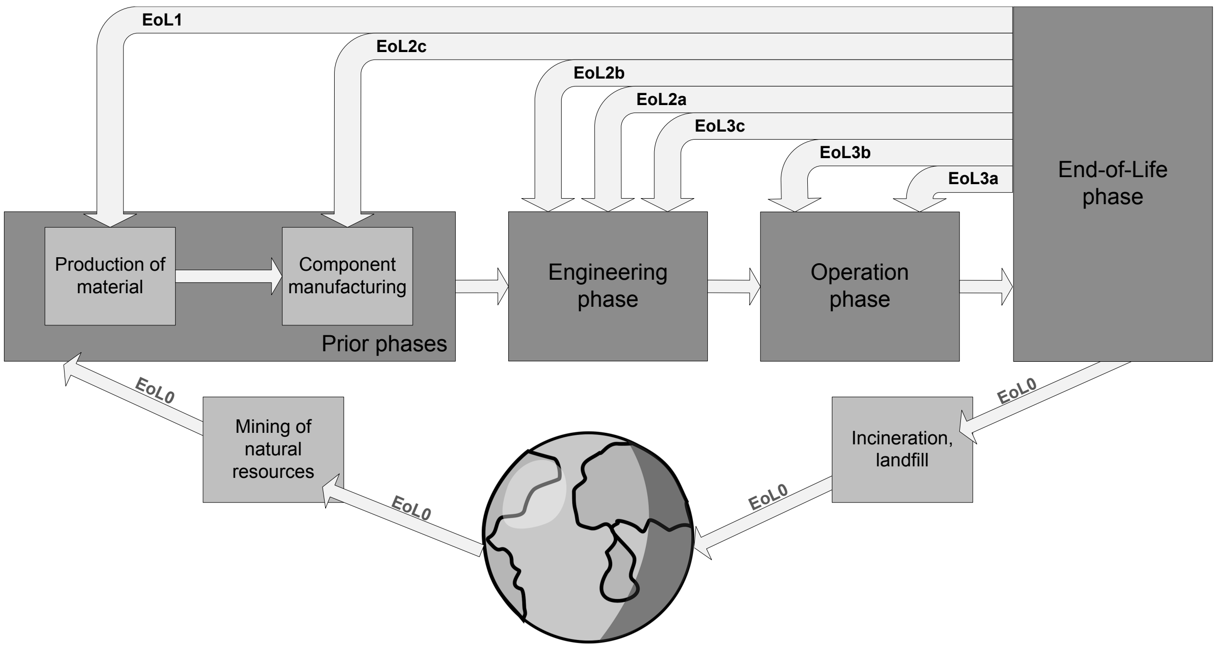

Figure 3.2: End-of-Life scenarios of production systems visualized in the life cycle (based on [DSK+08], [PBFG07], [VDI02], [KIK08], [GAB+11], [KM08], and [SBK01])

Figure 3.2 is in accordance with Figure 2.6 and covers the recovery possibility cases 19 (corresponds to EoL3a and EoL3b), 20 (corresponds to EoL3c, EoL2a, and EoL2b), 21 (corresponds to EoL2c), and 22 (corresponds to EoL1).

In order to enable the described EoL scenarios, certain information from the life cycle phases of the production system is required. These are identified in the following section by developing an EoL data model. For it, only the information about the production system is considered - not the physical production system itself. However, this is also required to enable the EoL scenarios.

3.3 End-of-Life data model

By reviewing literature, a vast number of requirements on the recovery of production systems were identified. There was also information identified that was, then, assigned to each requirement. By that, the EoL data model was developed.
Based on the gathered requirements, a classification could be made: system structure (SS), connection structure (CS), material structure (MS), and system design and characterization (SD). This follows basically a general consideration used in the recycling oriented product development (see e.g. [CRM12] and [VDI02]). The *system structure* describes the entire system with its components and the relations between them. The *connection structure* focuses on the description of the interfaces / connections among the components and between the system and the components. The *material structure* describes or characterizes the materials used. The relation between the materials is not considered. The *system design and characterization* was added by the author since the scope of this thesis is not just recycling (i.e. material recovery) but recovery in general. Besides the characterization of the material for the material recovery, the system and its

components also need to be characterized for the production system and component recovery. System structure and connection structure are essential for every recovery goal.

In the following sections it will be distinguished between the information that needs to be provided for the EoL by the previous life cycle phases (in accordance with the cases 16 and 17 from Figure 2.6) and this information that needs to be provided by the EoL phase itself (in accordance with case 18 from Figure 2.6). As a result, the base EoL data model is developed first and then the EoL process data model itself. Both converge into the EoL data model.

3.3.1 Feed forward

In this section the requirements and information are identified, which need to come from the engineering phase and operation phase in order to enable a proper EoL phase. These represent the cases 16 and 17 from Figure 2.6 - feed forward. In Annex A, all requirements with their information are grouped according to the given classification. Besides this, the author has made a proposal how each information could be modeled in a UML class diagram. Annex A also contains a UML class model for each category of the classification. The four UML class models are aggregated to one UML class model called base EoL data model. This is shown in Figure 3.3.

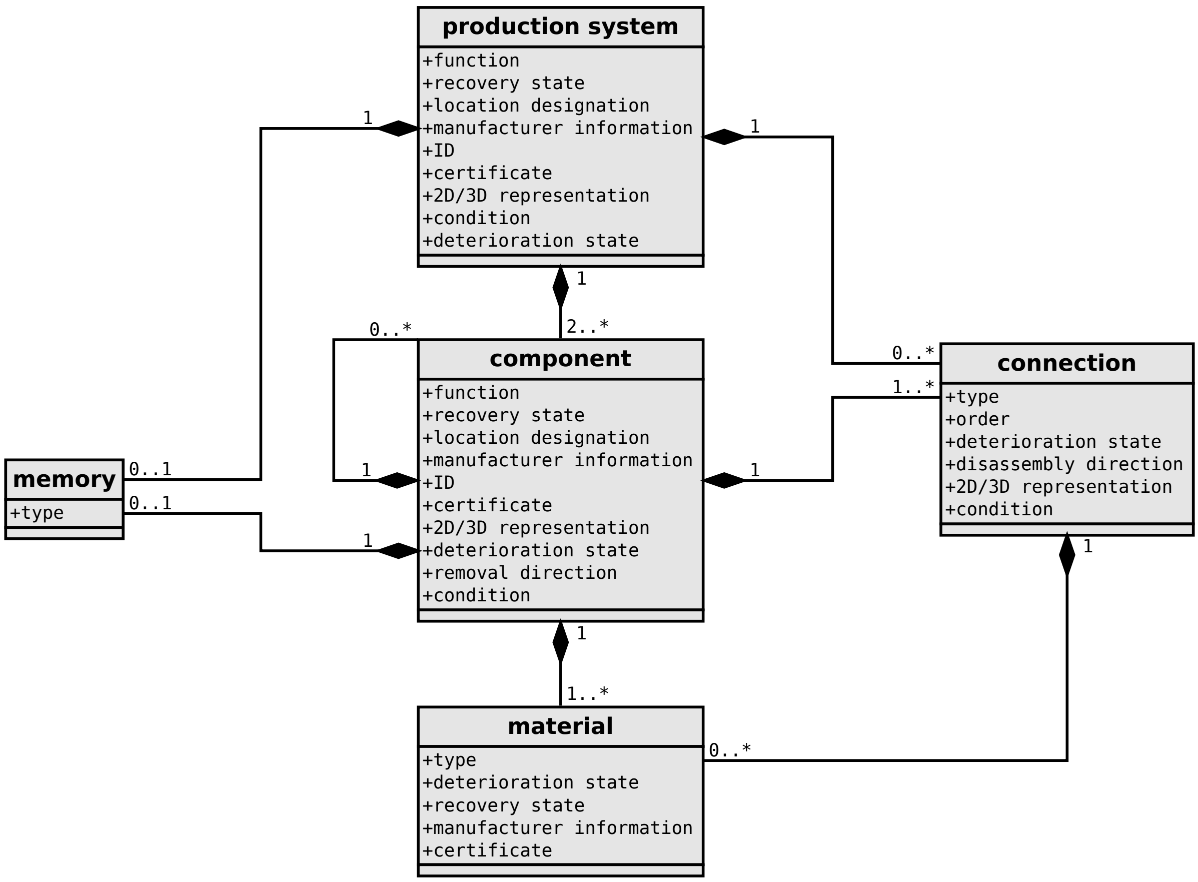

Figure 3.3: Resulting UML class diagram based on requirements and information - called base End-of-Life data model

Besides the information about the deterioration state and condition, information need to come mainly from the engineering phase. However, the engineering information needs to be kept up to date during the operation phase, even after repair or maintenance activities had been executed [SFR02], [Hub01b], [Ruh06], [FTM99], [DSK+08], [SO13], [OHB+13], [Hub01a], [SBK01]. This up to date kept engineering information must, then, be extended by operation phase specific information. For recovery purposes this is only the deterioration state and condition. Basically, Figure 3.3 represents a data filter. The EoL phase does not need all information neither from the engineering phase nor from the operation phase.

In literature such base data models, which focus on providing this data necessary to enable a certain EoL scenario, can be found.
[PAS04] proposes a recycling passport that shall be delivered with a certain product so that the recycling companies have all the information they need to recycle the product.
The EU project SustainHub had developed a sustainability data management for the automotive and electronics sector to exchange compliance and sustainability data across the supply chain. In [BBS+13] an eco-/sustainability data model for SustainHub is developed.
The International Dismantling Information System (IDIS) provides dismantling information about cars. The system can be accessed by recycling companies to get the information they need to disassemble and recycle cars [VDI02].

As mentioned above the EoL phase does not need all information neither from the engineering phase nor from the operation phase. However, the EoL phase does need additional information that is EoL relevant. This will be discussed in the next section. The base EoL data model will be extended to the EoL process data model and, finally, to the EoL data model.

3.3.2 Feed within

In this section the requirements and information are identified, which need to come from the EoL phase itself. These requirements and information represent the case 18 from Figure 2.6 - feed within. In Annex A, all requirements with their information were grouped according to the classification and modeled in a UML class diagram. However, within the column 'Representation in UML class diagram' at certain information it is said that it is process relevant and will be considered later or that it can be derived by certain attributes. This information will be analyzed in more detail in this section to create the EoL process data model and, finally, the EoL data model.

Figure 3.3 represents the information that needs to be provided to the EoL phase. However, in order to enable a recovery of the production system or its components or materials, additional information from the EoL phase is needed. Those are either derived from the base EoL data model (shown in Figure 3.3) or from other external sources (like market analysis or external service providers).
In Annex B the remaining requirements with their information are grouped according to the classification. Besides this, the author has made a proposal how this information could be modeled in a UML class diagram. Annex B also contains a UML class model for each category of the classification. The four UML class models are aggregated to

one UML class model that represents the EoL process data model. This is shown in Figure 3.4.

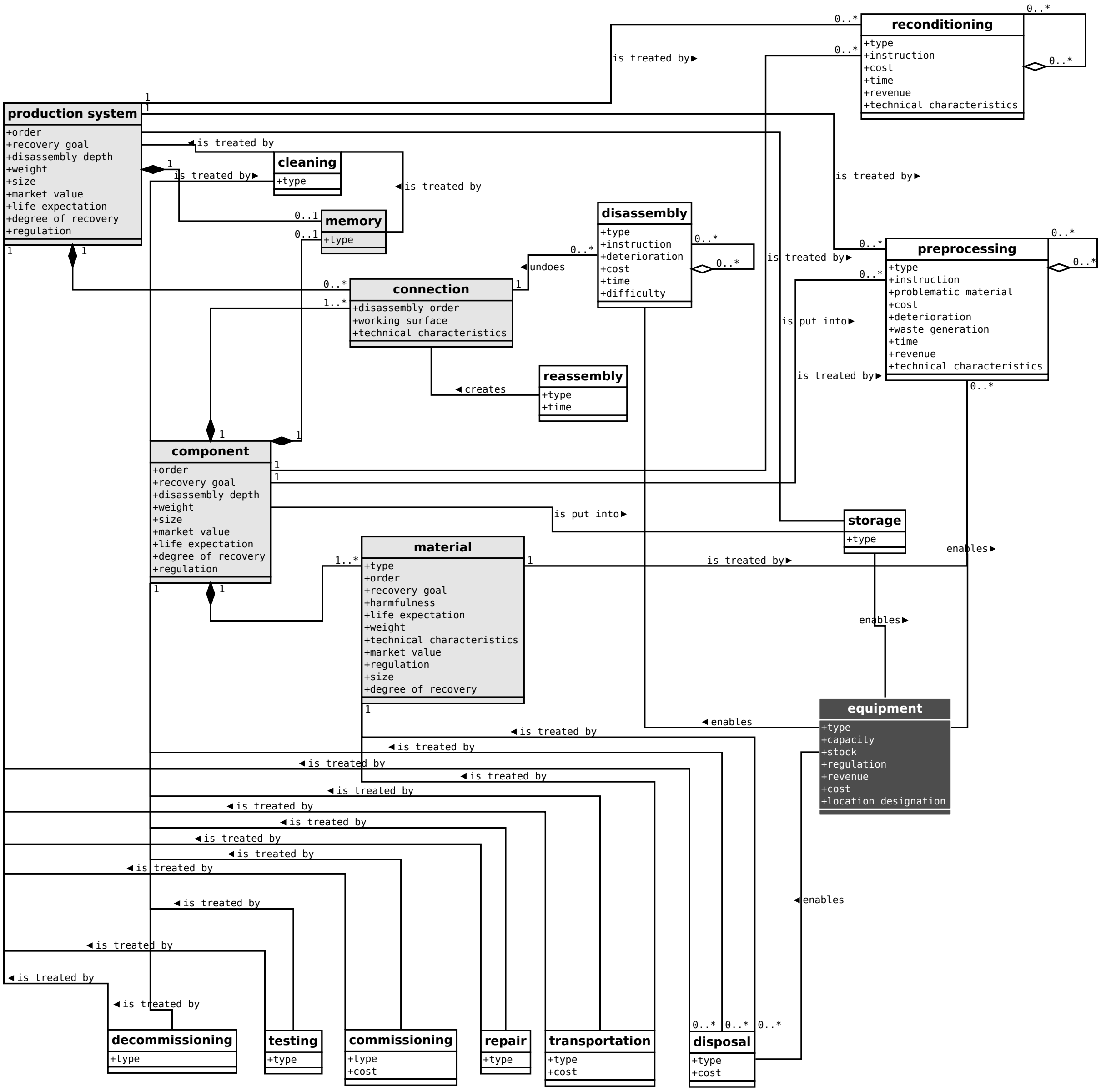

Figure 3.4: Resulting UML class diagram based on requirements and information resulting in the End-of-Life process data model

Figure 3.4 not just represents the EoL process data model with its classes, attributes, and relations. The author had also highlighted the three different meanings of the classes. The PPR concept (Product - Process - Resource) is applied - see [IEC14], [MMB02], [CDYM+07], [WSCL13], [SD09], [Dra10]. When seeing this PPR concept from a different perspective and adapting it to the context of this thesis, the base EoL data model represents the ’product view’. This is indicated by coloring the corresponding classes in light gray. In the scope of this thesis, the product, i.e. the object that is intended to be produced, is actually the production system, its components, or materials that get recovered. Since the recovery process is a process with certain processes that uses resources to produce the product, the ’process view’ and ’resource view’ needs to be added. In Figure 3.4 the processes are colored in white, the resources in dark gray.

In the next step the base EoL data model and the EoL process data model are merged into the EoL data model, which is shown in Figure 3.5.

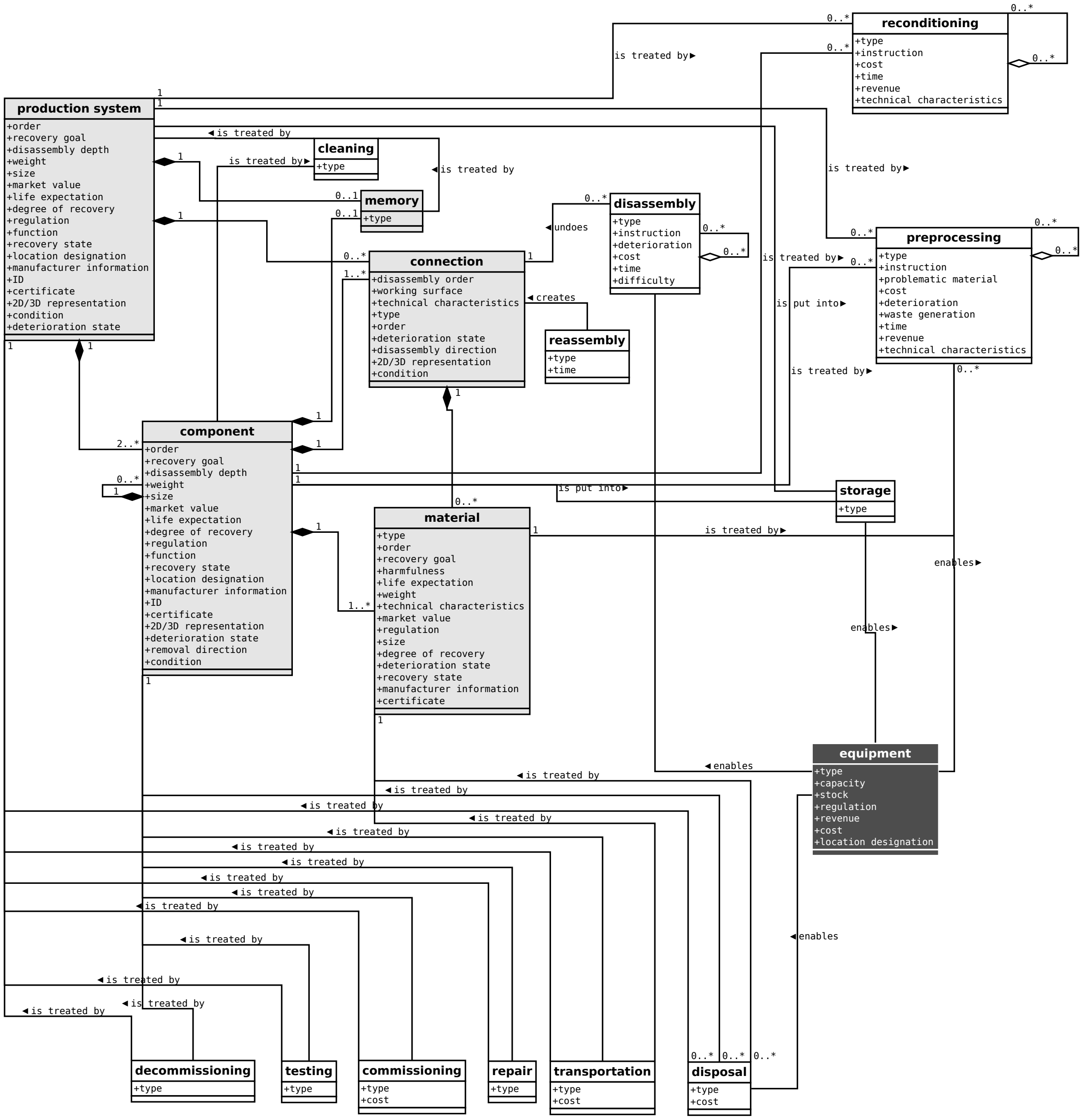

Figure 3.5: Resulting End-of-Life data model: Base End-of-Life data model integrated into the End-of-Life process data model (as UML class diagram)

In literature such data models, which incorporate the PPR view, can be found. They can provide this EoL phase specific data by only considering one EoL scenario or EoL strategy up to considering several ones.
[SFR02] identifies the necessary data for a disassembly planning and control system and how this could be modeled in SAP. The use case is a disassembly factory. Here, products of many different kinds (like washing machines) are the input and components or fractions of material are the output. Knowledge about the product structure and its connections is required.
[LSsB06] develops a generic life cycle model with which EoL scenarios can be modeled.

Regarding the EoL strategy, it is mainly focused on remanufacturing.
The EU project PROMISE had developed concepts, models, methodologies, and technologies to enable an information flow throughout the entire life cycle of a product across the value and supply chain [KM08]. It distinguished the life cycle into BOL (beginning of life), MOL (middle of life), and EOL (end of life). In [CTMM07] a life cycle data model was developed that covers those three phases.

In Annex B, within the column 'Representation in UML class diagram' at certain information it is said, that it is decision-making process relevant and will be considered later. These are the information or requirements:

- Provide information if a component is reusable, recyclable, or needs to be disposed [SFR02]
- Which components can be recovered [Hub01b], [Pla16a]
- Evaluation of suitability for further use phases [SBK01]
- Provide guided benchmarking including clearly defined KPIs [SBSW12]
- Give information about market [Hub01a]
- Determine residual value of materials, components, or production system [VDM06], [Hub01a]
- Give information about probability that the system is used again [Hub01a]
- Provide information about condition and life time dependent quality properties [SFR02], [SBK01]
- Assessment of the load of the used systems includes properties like mobility, functionality, degree of contamination, harmfulness, deterioration state, deformation state, corrosion state [Hub01a]

The bottom line of this is that the production system needs to be evaluated regarding the EoL scenario that can be initiated from the EoL phase. These represent the cases 19, 20, 21, and 22 from Figure 2.6. The recoverable physical assets need to be identified. However, to answer the question what EoL scenario is optimal for a specific production system, it is not enough to just identify them. They must be characterized consistently (based on the EoL data model) to become comparable. This is the focus of the next section.

3.3.3 Feed back

This section focuses on the decision-making process. How can it be evaluated, which EoL scenario is the best one? For this, a characterization of the different scenarios is required. In order to find an optimal EoL scenario it may not be sufficient to only estimate the residual value or market value of the production system, or certain components of it, or the material. Nevertheless, the quality (e.g. reliability) should be considered [MT02], [Gar87]. However, as explained above, each EoL scenario consists, besides of the recovery goal, also of the EoL strategy. That means, the processes are also relevant and should be characterized (see Figure 3.1). Processes need to be planned,

coordinated, and executed. That takes time and costs money. For example [VDI02] compares the costs to buy a new system, component, or material and sell the old one with the process costs to get the old one recovered. By this, the suitability for recovery can be calculated.
In Annex B, all remaining requirements with their information were grouped according to the classification and modeled in a UML class diagram. However, within the column 'Representation in UML class diagram' at certain information it is said that it is relevant for characterization and will be considered later. This information is now analyzed in more detail in this section to find a characterization or indices for the processes.
This is the information in particular:

- Degree of quality reduction due to be processed [PBFG07], [SBK01]
- Economic assessment of the recycling processes including disassembly, separation, preprocessing, and logistics [VDI02], [SBK01]
- Provide information about costs and revenues [PAS04], [Ruh06], [RMTF99], [DSK$^+$08], [HS94], [SPPR97], [VDM06], [Hub01a]
- Assess environmental impact [HS94]

When looking at this information as well as at the attributes of the process classes in Figure 3.4, it can be seen that the economic (e.g. cost, revenue) and ecological impact (problematic material, waste generation) are relevant. These indices can be found in publications on, in general, assessment methods e.g. regarding costs (see e.g. [VDM06], [otZAD12], [VDI02], [VDI05]) and regarding ecology (see e.g. [HS94], [DIN13]). Also a combined consideration can be found, like [RABW09] and [MT02]. When the economic and ecological impact is extended by the societal impact, it is called sustainability. Publications that consider the three impacts are e.g. [SSZ$^+$14], [BNS$^+$15], and [HJA05a], but also considering only societal impact, like [Lif09], can be found.
In addition to them, two other impacts can be extracted from the process classes in Figure 3.4: time and quality (e.g. deterioration) impact. The time impact is alike the economic impact, since a recovery process is also about making room on site, e.g. for another production system. In the case that the removal needs to be done quickly, time can be a critical factor. The quality impact indicates whether the process has an effect on the quality of the production system, component, or material. For example, the quality of a certain component is assessed and it is fully functional. Then, the component needs to be transported on a truck from location A to location B. After that, the component is assessed again and it is not working anymore. In this case the transportation process has an effect on the component.

In the following section, methods are analyzed that deal on the one hand with the characterization of processes necessary for the recovery, namely EoL strategies. On the other hand with methods that characterizes the system itself. If the system is characterized it can be compared with the requirements on the future application of the recovery goal. By having both characterized, a suitable EoL scenario can be chosen.

3.4 Methods for recovery support

This section analyzes methods that support the EoL phase by enabling a characterization of different EoL scenarios. This is the basis for the decision what is happening with a certain production system at the end of its life.
In literature several methods can be found. They differ in the life cycle phase in which the method is applied but also in the scope. Therefore, they are analyzed whether the method is providing a characterization of the system itself and/or EoL strategy. And if so, it is listed, which assessment indices for characterization are proposed.

[VDI02] proposes not just a checklist that helps to do a recycling-optimized product development. It also proposes criteria for recycling namely environment, technology, ecology, and economy. It focuses on reuse, recycling, and disposal and is applied in the engineering phase and EoL phase. But a consideration of product or system itself is not given.
[RABW09] and [ABR08] propose a recycling planning method. The goal is to find a disassembly path for a certain product (e.g. washing machine, building) that generates the best output from the recycling point of view. Costs and revenues, available recycling techniques, legislative compliance, environmental impact are aspects considered - in general the ecological and economic impact. A consideration of the recovery goal, a characterization regarding the requirements on the future application, is not given.
[SBK01] proposes an adaptation planning method. Adaptation processes mean processes that increase the use time of a product (reuse), like repair, modernization, remanufacturing, or downgrading. Depending on the chosen adaptation process different subprocesses are required, also to a certain extent. For example, repairing a product does not need such a deep disassembly path than it is required to remanufacture it. The adaptation planning method uses this information as well as the information of the product (regarding product characteristics and current state) and information for which purpose the product will be used again. Out of this set of information the optimal adaptation process for a certain product at the end of its life is calculated. It is said that adaptation process information is needed and that they should provide information regarding economic aspects, but a detailed consideration cannot be found. It is also said that the different levels of reuse need to be identified before they can be evaluated. However, a detailed consideration of the recovery goal is not given.
[RSI00] develops an End-of-Life Design Advisor (ELDA) that uses a classification schema to identify suitable EoL scenarios within the engineering phase. Here, it is distinguished between reuse, service, remanufacturing, recycling (separate first), recycling (shred first), and disposal. Technical product characteristics are provided as a list and used in this characterization schema. ELDA is intended to be used before the economic and ecological assessment to, first of all, decide for a suitable EoL scenario from a technical or quality point of view.
[MT02] proposes a planning method for component reuse in product portfolio design. It aims to support the engineer to design a product portfolio that incorporates recovery as the criterion to address certain market segments. For example 'green consumers' are on the one hand willing to spend more money on a 'green' products that has some recovered components used and are on the other hand willing to accept a not so good reliability. With the focus on those market segments, the different EoL scenarios are assessed (regarding costs, reliability, and environmental impact). Besides the EoL

strategies reuse, and remanufacturing, recycling, the strategy new is considered.
[SSZ+14] defines a product sustainability index (ProdSI) that allows the assessment of the entire life cycle of a product regarding its impact of sustainability. It is applied in the engineering phase to make products comparable regarding sustainability. It is not focusing on any specific process. Processes are considered on a high level, like 'product remanufacturability' or 'waste management regulation compliance'. The product is assessed, not explicitly the EoL strategy. The product is characterized.

There are also other publications that have similar EoL support methods, like [SPPR97] and [RMTF99], [FTM99], [SO13], [AMR13], [FMK+13], and [Hub01a] for disassembly and recycling. The author tried to choose publications that show the variety of EoL support methods without having a certain scope several times.

A method that explicitly focuses on production systems with this scope of this thesis could not be found.
The EU project ReBORN actually focuses on production system with its equipment, but the scope is different. It is about extending the life time of production equipment. For this, ReBORN had developed strategies and technologies to decide, which reused production equipment can be used in the factory design process to enhance this process on the one hand. On the other hand how the usage of reused production equipment can, then, shorten the ramp up time of the new designed factory. [COR16] This is a different point in time when the decisions need to be made and it is out of the scope of this thesis.
The two different scopes are shown in Figure 3.6. The difference is the point in time, when the decision-making process is executed.

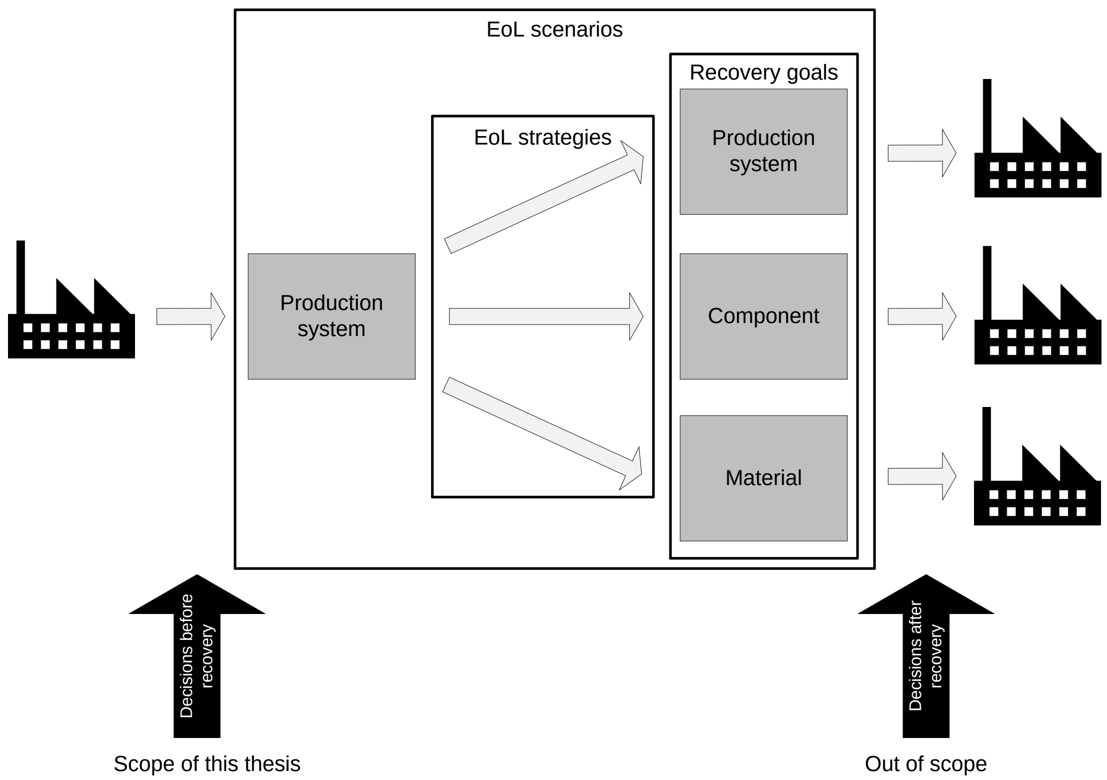

Figure 3.6: Scope of this thesis related to the point in time of the decision-making

To import, store, handle, manipulate, and possible export data or to enable the application of those methods for recovery support, usually software and information technologies (IT) are used. Therefore, the next section will analyze available software and information technologies.

3.5 Software and IT support

This section is dedicated to present the currently available software tools that support decision-making processes in the EoL phase on finding an optimal EoL scenario for production systems. But as it is described above, the EoL phase is the last phase of the life cycle of a production system and needs information from the previous phases. Therefore, EoL software tools need to import production system information from the previous phases.
In order to engineer and operate a production system a lot of different software tools from different tool vendors are used. For their functionality they need to import certain data. If this software tool has not just monitoring purposes, it usually manipulates the imported data according to its functionality and exports it afterwards. Thus, the possible support of IT is analyzed. Those two aspects are highlighted in Figure 3.7 by a dotted box.

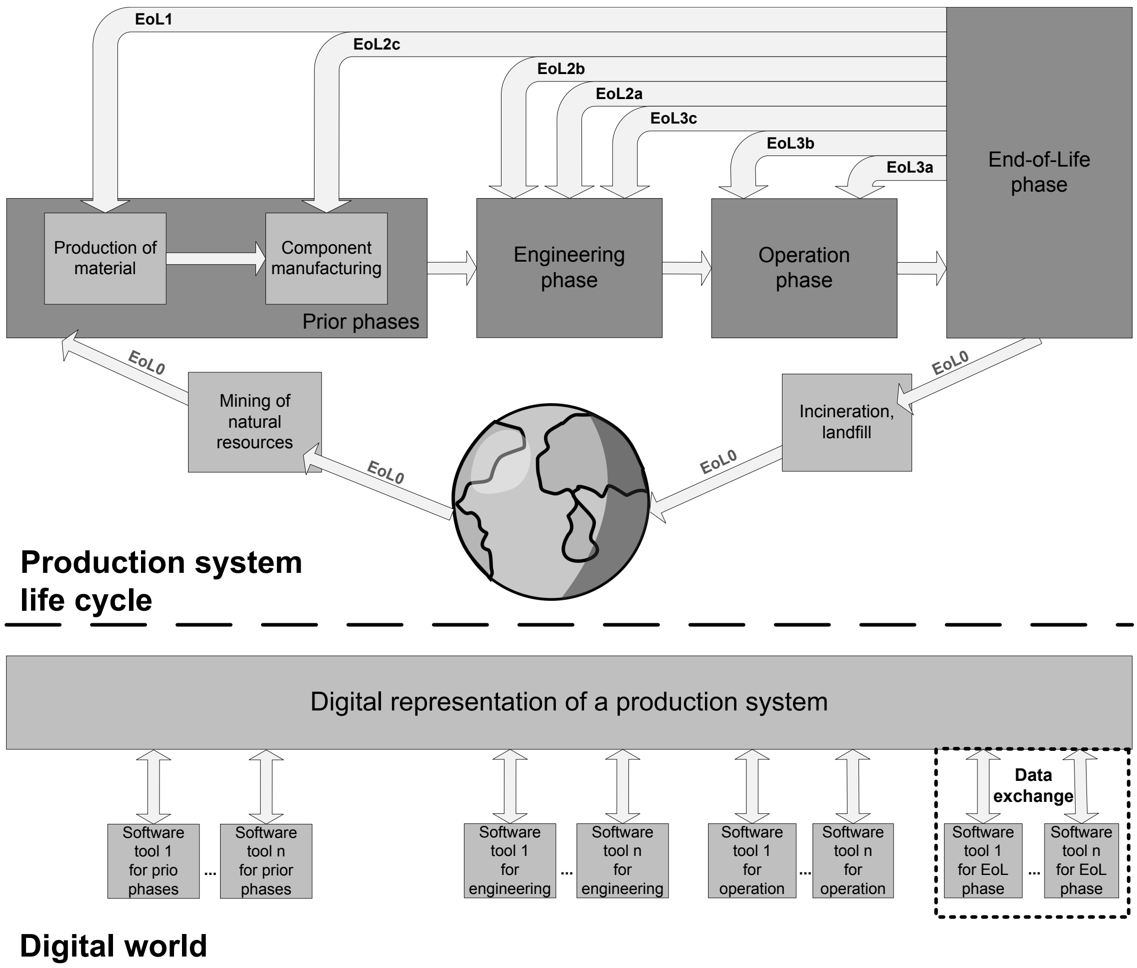

Figure 3.7: Software system architecture (based on [Her10]) combined with the production system life cycle with its EoL scenarios

3.5.1 Software tools

This thesis is focusing on the software tools that are capable to support the decision-making process for the recovery of production systems. Here, only those software tools are discussed which are used in the EoL phase as indicated in Figure 3.7. Needless to say, that there were also software tools developed that are used in the engineering phase to compare different constructions or designs with each other. Such software tools are for example DEMROP that evaluates recycling oriented designs [BMB00]. Or the Recycling-Graph-Editor (ReGrEd) that allows the modeling of products with disassembly and recycling information [FTM99] [RMTF99]. [SDF02] had surveyed 15 disassembly planning tools and [LE13] 28 methods and tools that focus on minimizing the environmental impact of product designs (called EcoDesign).

Another possibility is to follow or apply modeling guidelines used in general design software tools. Prominent examples are the Design-for-X guidelines, which are manifold, e.g. the Design-for-Environment, Design-for-Recycling, Design-for-Disassembly, Design-for-End-of-Life, Design-for-Remanufacturabilty, or the Design-for-Reuse [KIK08] [Hil01] [MMM+12] [BH96] [JAA+93]. But also Design-for-Assembly guidelines help for disassembly [Pan13], even though disassembly cannot be considered as the simple inversion of assembly [Ohl06].

In the following a list of EoL software tools gathered during the literature review is provided. Some methods described in Section 3.4 were also realized as software tools.
The Disassembly-Planning-System (DisPlay) calculates the optimal disassembly path and a suitable recycling strategy. For this, all possible disassembly paths are identified. Then, the material fractions of each resulting, disassembled component of the product are analyzed to identify a suitable recycling strategy. By that, the disassembly costs can be compared to the revenues created by the resulting material factions. [RMTF99] [FTM99]
The Computer aided Recycling Process Planner (CARPP) generates recycling plans and characterizes each plan by calculating the ecological and economic performance of it; Based on standardized recycling plans for WEEE products and the analysis of the materials of a specific product. After customizing a standard recycling plan to the specific product, the recycling process is assessed ecologically and economically. [RABW09] [ABR08] [ABR07]
The Eco-Advisor is a mobile application that is capable to aggregate recycling relevant information from different (web) sources for certain products, e.g. a soda can. The mobile application can, then, disassemble virtually the product into its materials. In doing so, the Eco-Advisor assesses the product's environmental impact as well as proposes the consumer a suitable recycling option for the product. [KK14]

The literature review resulted, besides the EoL data model, also in a non-exhaustive list of requirements on software support. These requirements are listed in Annex C. It is indicated whether a requirement applies to the software tool functionality or not. Since it is not the scope of this thesis to develop an EoL software tool, the requirements regarding software tool functionality are not considered further.

To import, store, handle, manipulate, and possible export data or to enable the application such EoL software tools, information technologies (IT) are used. Before information technologies will be analyzed, the next section will deal with the Industrie 4.0 paradigm since it implies certain requirements on the digitalization and, therefore, information technologies.

3.5.2 Industrie 4.0 paradigm

Industrie 4.0 as the fourth industrial revolution is a term - for the digitization of the industry and the usage of IT system within the entire value chain - made by the German government. Other countries have similar initiatives with different names, e.g. advanced manufacturing (US), e-Factory (Japan), or intelligent manufacturing (China) [IEC15b]. But the term Industrie 4.0 is also known beyond Germany's borders [Sen16].
This digitization of the industry can make smart manufacturing systems possible. By making the manufacturing system or production system smart companies could cope with the changing market and requirements of the customers on their products regarding customization possibilities and shorter product life cycles. Those requirements have effects on the production system regarding production agility, quality, and efficiency. This smartness results in the abolishment of the conventional hierarchical automation pyramid with its control model and in the establishment of distributed production services, instead. This is called Cyber Physical Production System (CPPS). It consists of Cyber Physical Systems (CPS), which are systems that connect a physical asset and process with a digital asset or information asset and process by using open and interconnected information networks; Thus, they provide new features, properties, and/or services [Fra] [GB12]. Industrie 4.0 addresses the technical integration of those CPS into production [KWH13]. [LMF16] [Sen16]

Industrie 4.0 related terms often come with relative general definitions and are, therefore, not properly used in publications, like digital twin, administration shell, or I4.0 component. [WGE+17] analyzed the definition of those terms, found out the common concept, and mapped the terms onto the common concept. This common concept has three levels: asset, data, and administration. An asset is a resource that provide physical services. With data or data object it is understood the data based representation of the asset. The administration or administration facade is associated to an asset, has access to the data of the asset, and basically provides Industrie 4.0 conform interfaces.
In this thesis, the CPS is considered in a more simple way, because it is not required in that level of detail due to the scope of this thesis. Relevant is only the asset and the data. With asset it is understood the physical asset (production system, component, or material). And data represents the digital representation of the physical asset. The administration is out of the scope of this thesis.

Assuming such a smart production system as mentioned before, it would be possible to have a digital representation of the physical production system and, therefore, the as-is state of it (see Figure 3.8). However, this idea is not new. [Web08] mentioned in 2008 the life cycle document that is kept up to date during operation phase of the production system. Also related fields of work came up with this idea. [SBK01] mentioned in 2001 the life cycle unit (LCU), which collects and stores product related information. With the building information modeling (BIM) a similar approach is pursued only for buildings with their internal infrastructure [BKKB15]. The difference between now and then is, that the IT technologies had evolved and with them this idea could finally come true.

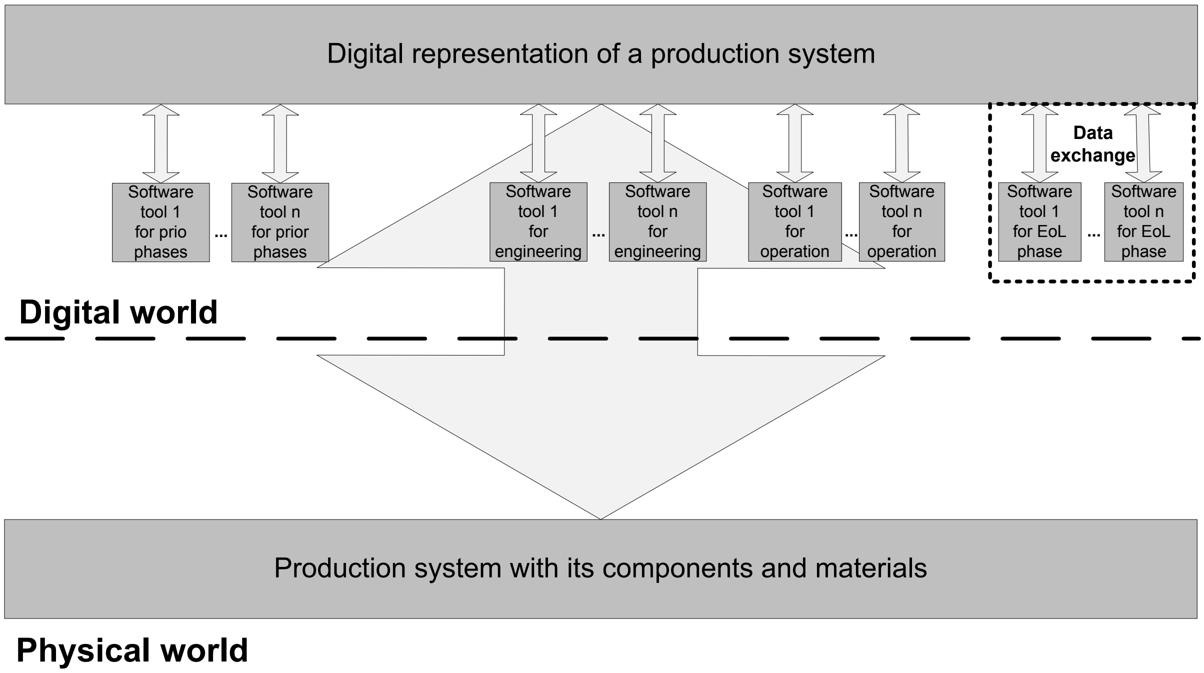

Figure 3.8: Physical and digital world of a production system

To reach this goal of a CPPS, (engineering) data consistency is important and efforts regarding reference architecture, standardization, security of distributed system, legal conditions need to be made [Sen16] [KWH13] [FSD$^+$16]. [LMF16] identifies the data transfer from shop floor to enterprise level as one important aspect where standard support is needed. In the scope of this thesis, the next section will, therefore, focus on the standardized data transfer. In this thesis it is not considered in detail how a production system or its components should be built up or communicate (reference architecture). However, in the EoL context the EU project ReBORN had stated that production equipment should be on the one hand versatile and modular, task-driven plug&produce devices and on the other hand self-aware and knowledge based so that they can collect information about their evolution over time. This includes information about repair, maintenance, upgrade, and deterioration. [COR16]

3.5.3 Information technologies

In a software tool chain it is difficult to stepwise create the EoL data model (described in Section 3.3.2), since information from engineering as well as operation phase are needed. If so, each software tool would need to export its imported data in addition to its added and manipulated data. For this life cycle phase crossing data storage, a central database is usually used - called PDM or industrial information systems [DLH$^+$10] - that would be based on an integrated EoL data model. A PDM system (product data management) is fed by different software tools from different life cycle phases. This technical information system is used to store, manage, and provide this production system describing data and information or documents [Her10].

PDM systems can come with different scopes regarding the product or production system information they store, manage, and provide. The International Dismantling Information System (IDIS), for example, focuses on providing disassembly information that are relevant to recycle cars at the end of their lives. The International Material Data

System (IMDS) handles material data for the materials that are used in cars, so that recycling companies can identify problematic materials. [Her10] [BOM] can support this by informing about the latest new substance regulations around the world. The EU project SustainHub developed its own data management system to store, manage, and provide compliance and sustainability data [Sus]. The Recycling-Data-Management-System (ReDaMa) stores, manages, and provides disassembly and recycling relevant data [RMTF99]. The e-Cycling platform supports the entire product life cycle [SBK01]. The Product Data and Knowledge Management system (PDKM) also stores, manages, and provide data from all lifecycle phases of products [KM08]. The AML.hub is focusing on production system data but only in the engineering phase [log17].
This thesis could propose to create an own data management system based on the EoL data model describe in Section 3.3.2. But this is not the scope of this thesis. Instead the author proposes, if needed, to use existing PDM systems that store, manage, and provide the overall life cycle data of a production system.

To enable the data exchange between software tools, directly or between the software tools with a PDM system in between, data exchange formats are required. It is assumed that the production system life cycle is dominated by a heterogeneous software tool landscape and is not entirely covered by only one tool suite, which would make a discussion about data exchange formats redundant, since all software tools would be 'under one roof'. As mentioned above the focus is on a data transfer or exchange that shall be standardized. Therefore, no own data format is developed in this thesis. [LMF16] provides a standards landscape for smart manufacturing systems. In the following, standards for the data exchange formats are described.

[LMF16] divides the standards into those for the product development and those for the production system.
At the product development side, there is ISO 10303-239, LOTAR, and PLM XML mentioned for product life cycle data management. For product model and data exchange the following standards are listed: ISO 10303-203/214/210/242, ISO 14306 (JT), ISO 14739 (PRC), IGES and DXF, ISO/ASTM 52915 (AMF) and STL. There are also standards available in the categories 'modeling practice', 'manufacturing model data', and 'product catalog data'.
Even though, standards for product development are applicable for production systems as well [LMF16] and a differentiation may be difficult to make, they are neglected at this point. Instead it is focused on the production system standards, since standards explicitly for production system had evolved over the last years.
At the production system side, there is ISO 10303-239, ISO 15926, ISO 16739, IEC 62890, and IEC 61987 mentioned for production life cycle data management. For production system modeling and data exchange there are several standards listed. Beside this category, also standards are available in the categories 'production system engineering' and 'production system maintenance'. The standards for production system modeling and data exchange are analyzed in the following.

ISO 10303-225 for 'building elements using explicit shape representation' [ISO99] is a part of the STEP family of standards that specifies a protocol (connecting computer systems) that can transfer the following information about: 3D geometric information, spatial configuration, structural information, nonstructural enclosure information, geo-

metric and spatial information of building services, geometric and spatial information of building fixtures, description of spaces, site information, building component properties, building component classification, building component changes, approval information related to building components, and as-built information [Aki11].
ISO 10303-227 for 'plant spatial configuration' [ISO05] is also a part of the STEP family of standards that specifies a protocol for the spatial configuration of process plants. The scope is on shape and spatial arrangements of the components of the piping system of a plant. [GDKL06]
STEP, standardized in the ISO 10303 family of standards, enables the modeling of products. For this, it provides a data model, a modeling language, a data exchange format, a protocol for data exchange, and model conformance mechanisms [Aki11].
ISO 16739 defines 'Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries' [ISO13]. It focuses on the exchange of information about construction projects throughout the life cycle that, in particular, includes information about building geometry, topology, spatial structure, building elements, and relations between building elements, but also building equipment, furniture, people, organizations, and project data [GDKL06].

According to [LMF16] those standards are from the building/facility domain and are, therefore, out of the scope of this thesis. The other group of standards is coming from the manufacturing resource and process domain. They are described in the following.
IEC TS 62832 for 'digital factory framework' specifies models and rules for the modeling of production systems [IEC16].
ISO 15746 for 'integration of advanced process control and optimization capabilities for manufacturing systems' focuses on the method to integrate those capabilities for manufacturing systems [ISO15].
IEC 62337 for 'commissioning of electrical, instrumentation and control systems in the process industry - Specific phases and milestones' specifies phases of milestones [IEC12].
IEC 61987 for 'Industrial-process measurement and control - Data structures and elements in process equipment catalogues' provides a generic structure for describing products of process measuring equipment with its features [IEC06].
ISO 10303-214 is mentioned by [LMF16] but it has been revised by ISO 10303-242.
ISO 10303-242 for 'managed model-based 3D engineering' is a part of the STEP family of standards that specifies a protocol (connecting computer systems) that can transfer the following information about: products with their parts, tools, assemblies of tools, and raw materials, engineering and product data, product data management, process planning, mechanical design, kinematics and composite design [ISO14a].
ISO 10303-221 for 'functional data and their schematic representation for process plants' [ISO07] is a part of the STEP family of standards that specifies a protocol (connecting computer systems) that can transfer the following information about piping and instrumentation diagram (P&ID). It comprises the identification of the plant system with its equipment, connectivity, classification, definition of standard functional and physical classes, materials, and project data [GDKL06].
ISO 18629 for 'Process specification language (PSL)' specifies the modeling and exchange of process information related to the discrete manufacturing industry [ISO04].
CMSD for 'core manufacturing simulation data' specifies an information model for the exchange of manufacturing data in simulation environments either as UML model or as

XML representation [Sim].
ISA 95 is the basis for IEC 62264. IEC 62264 for 'enterprise-control system integration' [IEC13] specifies the data exchange of production and order related data [Dic09].
IEC 62424 for 'representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools' specifies the modeling and data exchange of process control engineering request relevant data by using the XML based format called CAEX [DIN10].
ISO/PAS 17506 for 'COLLADA digital asset schema specification for 3D visualization of industrial data' specifies how geometry and kinematics data of digital assets can be modeled and exchanged by using the XML based format COLLADA [ISO12].
PLCopen XML for 'XML formats for IEC 61131-3' specifies how PLC programming project can be exchanged by using the XML based format PLCopen XML [Fou]. PLCopen XML is currently standardized within the IEC 61131-10.

The combination of IEC 62424, ISO/PAS 17506, together with PLCopen XML results in the data exchange format AutomationML, which is standardized in IEC 62714 (Engineering data exchange format for use in industrial automation systems engineering - Automation markup language) [IEC14]. [LMF16] had put AutomationML into the category 'production system engineering'.
AutomationML is a scalable, open, neutral, XML based, extendable, and free data exchange format, standardized in IEC 62714, which enables a discipline and company crossing transfer of engineering data between software tools. The data model, AutomationML, provides the production system structure expressed as a hierarchy of AML objects containing geometry and kinematics, logic information (behavior and sequencing), and relations among AutomationML objects as well as references to information that is stored in external documents. Each AutomationML object can integrate different information with different semantics related to different disciplines. [IEC14].

All standards on models and data exchange are again listed in the following table.

Table 3.1: List of standards on models and data exchange (based on [LMF16])

	Product development	**Production system**
Life cycle data management	ISO 10303-239, LOTAR, PLM XML	ISO 10303-203/214/210/242, ISO 14306 (JT), ISO 14739 (PRC), IGES and DXF, ISO/ASTM 52915 (AMF) and STL
Model and data exchange	ISO 10303-239, ISO 15926, ISO 16739, IEC 62890, IEC 61987	ISO 10303-221/225/227/242, ISO 16739, IEC TS 62832, ISO 15746, IEC 62337, IEC 61987, ISO 18629, CMSD, IEC 62264, IEC 62424, ISO/-PAS 17506, PLCopen XML, IEC 62714

Due to the expertise of the author (see [LSD17], [LS17a], or [LS17b]), AutomationML is chosen for modeling production systems and for data exchange among the presented standards. In addition, [LSH+17a] had identified typical description means or information types that can be found in the engineering, operation, and EoL phase. And it was found out that almost all types can be modeled with AutomationML. However, the author points out that also other standards might be suitable.

AutomationML is originally coming from the production system engineering field but with generic modeling capabilities. Schema files are available so that files can be created.

Engineering software tools can import those files and export them, so that at the end of the engineering phase an AutomationML document could be available that stores the entire production system model. During operation phase and with the Industrie 4.0 paradigm with the smart production systems in mind, the data transfer is different. It is not file based. The data transfer or communication is usually done by protocols. For this, a standard - namely DIN SPEC 16592 [DIN16b] - was developed that defines how AutomationML data models can be represented in an OPC UA information model [HS14] [LSS+17]. That means, components of a production system can be capable to describe themselves in AutomationML and communicate this during operation phase via OPC UA [VDM17b] to a superordinated system, which aggregates this creating an overall model - the digital representation of the production system. Newly installed, recently repaired, or still operating components - all could communicate their new description, changes, or state of wear and tear to the superordinated system. For example, [WUL+16] uses this idea. Here, AutomationML is used to realize a Module Type Package (MTP), which is a semantic description for process industry modules to have them fast and efficiently integrated into process control systems. MTP is mapped onto the OPC UA information model so that an online discovery of those process industry modules is possible.
Another example is the project CSC (Cyber System Connector) [MKD17] [LBL+16], which aims at the automated generation of the technical documentation of CPPS. For this, each CPS has its self-description as AutomationML and can communicate it, if requested, to a superordinated system, which generates the technical documentation of the overall CPPS or production system.
AutomationML and OPC UA used together could enable this scenario visualized in Figure 3.8 - connecting the physical world with the digital one.

Of course, it does not have to be OPC UA. It could also be Open Services for Lifecycle Collaboration (OSLC) or Data Distribution Service (DDS) [MP15]. But for those, the AutomationML e.V. has no mapping developed, yet.

As mentioned before, each physical asset that shall be recovered is always considered together with its digital representation or description. With the presented architecture it would be possible to not just recover the physical asset and, e.g. sell it to a company. The digital representation could be also provided and, thus, also sold to the company (together with the physical asset). And then, the EU project ReBORN [COR16] could use this digital representation as input for their decision-making process about which component fits the best into the new designed factory.

3.6 Conclusion and current limitations

According to the inductive scientific approach, the author strives for a holistic consideration of a production system at the end of its life. So, that the decision-making process can be supported in terms of finding a suitable EoL scenario for the production system. The input of the EoL phase is always the production system as physical asset with its digital representation.

Therefore, the author had analyzed main recovery possibilities or physical assets that can be generated by the EoL phase in Section 3.2. Here, it was identified as necessary to distinguished between the to be recovered physical asset (What?) and the process

how this is recovered (How?).
In Section 3.3 the author had, then, analyzed the necessary information, which needs to be provided by the engineering and operation phase to the EoL phase regarding the production system as well as information, which are only EoL relevant and need to be provided by the EoL phase itself. As mentioned before, the input of the EoL phase is always the production system as physical asset. All in all, an EoL data model was developed based on the literature review (see Figure 3.5). However, this gives only a fragmented view on the relevant data - especially for the process classes - since each publications came with a specific scope or use case. To develop a holistic approach, this fragmented view needs to be unified and generalized. That means, for example, that when the class *reconditioning* has the attributes *cost* and *technical characteristics* to give information about the process costs and the technical process parameters for the reconditioning process, also the other process classes, like the *cleaning* or *reassembly* class, should get those attributes. In doing so, a holistic and general overview could be provided that can, then, be adapted use case dependent.
Those fragmented view was not only found in the EoL data but also in the methods for recovery support, described in Section 3.4. Because those were also adapted to a specific scope or use case and the methods need certain input, EoL data model, to be applicable. The following table lists all the methods for recovery support that were described in Section 3.4.

Table 3.2: Comparison of EoL support methods

	Characterization of system		**Characterization of EoL strategy**			
Source	**Avai-lable?**	**Index**	**Avai-lable**	**Index**	**EoL strategy**	**Applied in life cycle phase**
[VDI02]	no	-	yes	environment, economic, technology, ecology, economy	reuse, recycling, disposal	engineering phase, EoL phase
[RABW09], [ABR08]	no	-	yes	economic, environmental	recycling	EoL phase
[SBK01]	addressed	-	addressed	economic	reuse	EoL phase
[RSI00]	yes	technical	addressed	economic, environmental	reuse, recycling, remanufacturing, service, disposal	engineering phase
[MT02]	yes	reliability	yes	economic, environmental	reuse, remanufacturing, recycling, new	engineering phase
[SSZ+14]	yes	environmental, economic, societal	no	-	-	engineering phase

This table analyzes the methods in terms of system characterization, EoL strategy characterization, and life cycle phase application.
In the first category it is analyzed whether the method provides a characterization of the system (product or production system), so that the current state of the system can be described somehow, so that the system can be compared with the future application for it and assessed, therefore, whether the system could ever reach the requirements on the future application. If a method provides such a characterization it is also listed, which index is used, i.e. is the system characterized in a technical way or an environmental

way. This characterization is important for the recovery goal of an EoL scenario.
In the second category it is analyzed whether the method provides a characterization of the EoL strategy. Is it described how a system can be recovered? When a method considers this, it is also listed whether it gives an index and which EoL strategy it names. This characterization is important for the EoL strategy of an EoL scenario.
The third category simply names in which life cycle phase the method is used (engineering, operation, or EoL phase).

Only [MT02] has a YES in both categories but this method is applied in the engineering phase. It is a viable approach to assess the different EoL scenarios in the engineering phase so that the system is engineered or designed in accordance with the chosen EoL scenarios. A disadvantage is that this decision is based on assumptions due to the lack of data at this point. Also, a production system compared to a product has a longer operation phase. Production systems can operate between several years up to decades. After years of operating, the production systems could have changed, market or legal regulations might have changed, new EoL technologies are available, or prices for materials have increased or decreased. This all could affect the decision for a proper EoL scenario. Therefore, such an assessment for a suitable EoL scenario should be executed or repeated when the production system has reached its end of life. In addition, [MT02] always has the same recovery goal: the product. For it, the different EoL strategies are considered. For example: Is it better to repair a component of the product, or is it better to recycle the material, process this recycled material again to a new component, or is it even better to throw the old component away, process natural resources/virgin materials to a new component which is, then, assembled again into the product? And which of these EoL strategies makes the product in the end cheaper, more green, or more technological advanced? But the author thinks another viewpoint is needed here and, therefore, different recovery goals should be available. This thesis is about analyzing the different options or treatments of a production system at the end of its life. And it is obvious that in case a lot of inherent value is destroyed the more effort is needed to produce an equivalent value. But components and material might not end up in another production system. Instead they could leave the production system life cycle and start a new life in another life cycle.
This thesis will develop a recovery planning method (RPM) for production systems that is mainly based on [MT02]. But according to the inductive scientific approach of this thesis, also aspects of the other methods are integrated into the RPM to get a general method. Use case dependent this general RPM may be extended or reduced. But as a starting point to enable the decision makers (in a systematic way) a holistic consideration of the recovery process, the author thinks, it can be a valuable tool.
Even though, approaches that extent the operation phase of the production system consider similar aspects and are faced with similar challenges (see [Str14], [WRN15], [DLS09], [Ski12], or [PER]), they are out of the scope of this thesis.

Besides the development of the RPM in this thesis, is it also considered how the EoL data model could be stored and exchanged. As already mentioned in Section 3.5 it is done with the data exchange format AutomationML - due to the author's expertise in this field. It could also be another format. But AutomationML is not just a data exchange format. Due to it modeling capabilities it can also be used for an internal data model of a future software tool.

The literature review resulted, besides the EoL data model, also in a list of requirements on software support. These are listed in Table C.1 in Annex C. It is also indicated whether a requirement applies to the software tool functionality or to the data exchange between software tools. In the last column it is described how AutomationML meets a requirement. Finally, Table C.1 in Annex C shows that AutomationML is capable to meet the requirements on the data exchange and also some on the software tool functionalities. The usage of AutomationML is extended in this thesis, i.e. AutomationML gets also applicable in the EoL phase. As a consequence, AutomationML becomes a good candidate to actually store the life cycle data of a production system and, thus, become a life cycle format.
The development of a software tool for the RPM is out of the scope of this thesis.

All in all, this thesis provides a data model (namely the EoL data model) and a method (namely the RPM). Based on this, companies can develop new business models. And by providing a mapping of the data model to a data exchange format in this thesis, the basis for developing new software tools is created. Those software tools could, then, support those new business models.

Before the RPM and the mapping of the EoL data model onto AutomationML are developed, the EoL phase of production systems is again analyzed. But this time not from a literature based point of view (as done in this chapter), but instead from a case study based point of view.
Due to the lack of publications about the EoL phase of production systems, the author decided to look for experts, which do the recovery process as their daily business, and interview them in case studies. The results of those interviews will be presented in the next chapter and these can be seen as an extension or an addition to the literature based elaborations of this chapter to get a holistic picture of the EoL phase of production systems.

4 End-of-Life phase of production systems (case study based)

This chapter analyzes the state of the art of the End-of-Life phase of production systems from a case study based point of view. It extends - with the interview results - the literature based elaborations from Chapter 3. The extensions comprise a closer look at the reasons why a production system can reach its end of life, at the EoL business ecosystem, which displays the involved stakeholders in the recovery processes, and also comprise a closer look at the legal basis for recovering production systems.

4.1 Methodical approach

The state of the art was captured by having looked for and found experts that do the recovery process of production systems as their daily business.
The inductive scientific approach emerges from the qualitative research, which is the basis for this chapter. It enables a holistic view on the recovery process - This is the analyzed object. The qualitative design used is case study (with multiple cases) and the data collection technique is problem centric interview with experts. Some experts additionally provided documents. A case study observes social entities and from that knowledge is generated, i.e. case studies enable a deeper understanding of complex phenomena and can answer the questions: Why? and How? The problem centric interview allows a purposeful data collection with the identification of dependencies and relations. [SR16] [Gep14] [Bla15] [EG07]
It was not possible for the author to accompany and observe a recovery process of the expert's companies due to availability of recovery processes and the duration of them. Therefore, the observations were done retrospectively. The experts were describing either one very specific recovery process (having in mind) or they were describing a generalized recovery process based on their experiences. The author assumes that also a combination of both could be possible.

4.2 Case studies for the analysis of the recovery process

In this thesis several case studies were done. Each case study represents one or two experts interviewed that do the recovery among life cycle phases (feed back) as described in Chapter 3. Table 4.1 gives an overview of the case studies.

Table 4.1: Overview of case studies

No.	Stakeholder	Data source	Recovery goal	Interview hours in total	Result
I	plant engineering company	interview	EoL3	2h	detailed process description, reasons, business ecosystem
II	demolition company	interview	EoL1	6h	detailed process description, reasons, business ecosystem
IIIa	component/ equipment manufacturer	interview, documents	EoL3	3h	detailed process description, reasons, business ecosystem
IIIb	component/ equipment manufacturer	interview, documents	EoL1	3h	detailed process description, reasons, business ecosystem
IV	manufacturer of products and consumables	interview	EoL2	4h	detailed process description, reasons, business ecosystem
V	marketplace provider	interview	EoL2	4h	detailed process description, reasons, business ecosystem
VIa	component/ equipment manufacturer	interview, documents	EoL3	3h	process description, reasons, business ecosystem
VIb	component/ equipment manufacturer	interview	EoL1	1h	overview of recovery process, reasons, business ecosystem
VII	assembly and relocation company	interview	EoL3, EoL2, EoL1	3h	overview of recovery process, reasons, business ecosystem

It is mentioned in which type of industry the stakeholder or interviewed expert is working. All interviews and details were anonymized to protect the knowledge and expertise of each one of them and, therefore, the company's potential competitive advantages. This table also indicates whether additional documents were handed to the author and about the duration of the interviews. The majority of the experts invited the author for the interviews. Two experts were interviewed by the author over the phone. One of the two were called several times to get the case study done. For the other interviews it was only one meeting for several hours (dependent on the availability of the interviewed experts).
As mentioned before, the analyzed object is the recovery process. Thus, the experts were asked by the author to describe one for the company typical recovery process. Each case study usually covers only the recovery of one physical asset (production system - EoL3, component - EoL2, or material - EoL1). Some companies were able to describe a recovery process for each of the physical assets. At this point, the author chose one to have for each physical asset at least two case studies. The distribution is also given in the table. A distinction between the EoL strategies was not explicitly made. The last column in Table 4.1 shows the results of each case study. Due to the problem centric interviews the author were able to collect information and knowledge about the reasons, why a production system can possibly reach its end of life, insights about the involved stakeholders in the EoL phase and the recovery process, and also a recovery process description from some experts. Dependent on the availability of the interviewed experts, it was not always possible to get a detailed description of their recovery processes.

There are many methods available to model processes and workflows. Formal methods can either be script based or graphically based. The author decided for the latter one due to their clearness, understandability during the interviews, and the author's skills. The graphically or diagram based methods can be divided into data flow oriented, control flow oriented, and object oriented. Due to the scope of this thesis and the data centric topic the data flow oriented method was chosen. Examples for those are Petri nets, swimlane diagrams, data flow diagrams, or IDEF diagrams. [Gad13] Based on the IDEF diagram [SFLS13] and [LFKB12] have adapted this data flow oriented method specifically to the engineering process by considering the categories: engineering (activities), artifacts, humans (engineers), and software tools. By using this called 4D method, possible fractions and gaps in the engineering process can be identified. Fraction means that there is a switch to another responsible entity in almost every category. This means it is very likely to lose information. A gap means that there is in one or more categories no responsible entity available, e.g. there is no tool at hand to execute a certain activity. [SFLS13] [LFKB12] Even though, this method is customized to the engineering phase it provides a holistic data centric view at the recovery process which is in line with the scope of this thesis. Therefore, it was assessed as suitable by the author to capture the recovery process together with the interviewed experts. The adaptation of it to the recovery process is as follows:

- engineering (activities) –> recovery activities
- artifacts –> artifacts (information and physical assets)
- humans (engineers) –> humans (engineers and technicians)
- software tools –> tools (software tools and hardware tools)

The method is visualized as shown in the following figure:

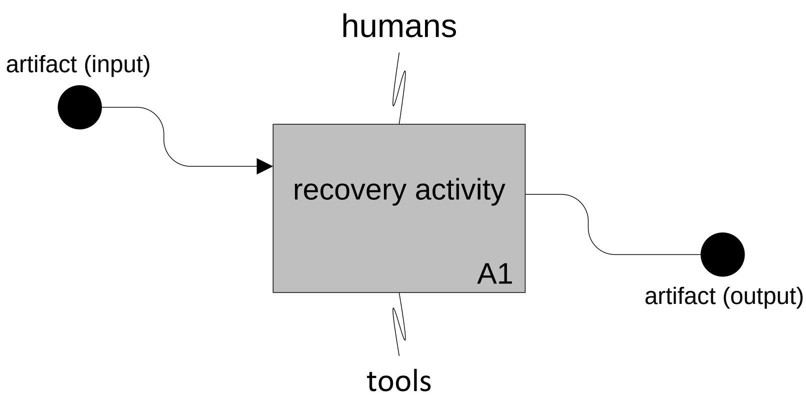

Figure 4.1: 4D method adapted to recovery process

In accordance with [SFLS13] and [LFKB12] the 4D comes, besides this visualization, also with a table for each activity that provides further information, e.g. description of the activity, description of the artifacts, and information about time (duration of the activity, starting time, and end point).
A complete description of the recovery processes from the case studies, covering the 4D visualization as well as the tables for each activity, can be found in Annex D. Additionally, each case study is textually described in more detail regarding the company's businesses and the recovery process. As mentioned before, the companies, experts, and descriptions are anonymized.

A look at the detailed recovery process descriptions in Annex D shows that besides the processes decommissioning, disassembling, transporting etc., which is basically the project execution, there is always a project planning needed before the execution. This planning phase is dependent on the EoL strategy applied and can, therefore, range from a simple project execution planning (EoLxa for direct reuse) up to an complicated project execution planning with an entire (re-)engineering of the system (EoLxc for reuse after remanufacturing). No matter what EoL strategy is used Annex D shows that there is always an inspection process required to get the “as-is” state of the production system, its components and/or materials as well as an examination of the available documents. Therefore, the availability of information is crucial. Dependent on the chosen EoL scenarios different information is necessary to do the project execution planning. The author sees potential to support this EoL phase with its recovery processes. A method with a holistic focus that supports the decision-making process, about which EoL scenario can be chosen, seems to be one possible and reasonable tool for the author. This method could shorten the inspection activity or could make it even redundant.

The next section will give an overview of possible reasons, why a production system reaches its end of life. These were collected during the interviews. It will show that there are good and relevant reasons why a production system reaches the EoL phase (according to the definition given in Subsection 2.2.3) and, therefore, should not be neglected.

4.3 Reasons for production system reaching the End-of-Life phase

The reasons why a production system reaches the end of its life are manifold. Based on interviews with experts of the case studies (see Table 4.1), which do production system recovery as their daily business, reasons were collected. They are grouped, in accordance with the general change drivers mentioned in Subsection 2.2.3, into *external drivers* and *internal drivers*. Those are listed in the following but are not intended to be exhaustive. The EoL phase of a production system can also be reached because of a combination of those reasons. In addition, the author indicates in these lists, which recovery goals could be achieved.

4.3.1 External drivers

Drivers that are coming from the outside and cause the EoL of a production system.

- Sales output is decreasing but the product shall not be discontinued.
  This can cause a relocation of the production system to another country (EoL3).
- Which country has the highest sales output of the product?
  This can cause a relocation of the production system to that country (EoL3).
- Operation costs at this location are too high.
  This can cause a relocation of the production system to another country (EoL3).
- Production system was destroyed, e.g. by a fire.
  This can cause a removal of the production system (EoL1).

- Production system does not meet (upcoming) demands, e.g. environmental restrictions or obsolete technology, and cannot be upgraded or it was never operating properly.
  This can cause a removal of the production system (EoL1/2/3).
- Components of the production systems are discontinued by a supplier. Therefore, spare parts are not available for repair purposes.
  This can cause a removal of the production system (EoL1/2).

4.3.2 Internal drivers

Drivers that are coming from the inside and cause the EoL of a production system.

- Which country manufactures the individual parts of the product?
  This can cause a relocation of the production system to that country (EoL3).
- Company is obligated to provide individual parts for this product for a certain time, e.g. as spare parts.
  This can cause a relocation of the production system to another country (EoL3).
- Product is manufactured in another way, e.g. another production system takes over the production or the product is manufactured in manual work.
  This can cause a removal of the production system (EoL1/2/3).
- Product is discontinued.
  This can cause a removal of the production system (EoL1/2/3).
- A company is producing the same product but at different locations as backup.
  This can cause a relocation of one production system to another location (EoL3).
- Production system is too worn out. Replacing it by a new one is more beneficial.
  This can cause a removal of the production system (EoL1/2/3).
- A company wants to strengthen a certain location/country.
  This can cause a relocation of a production system to that location/country (EoL3).
- Location of production system is needed for another production system - reorganization.
  This can cause a relocation of the production system to another location (EoL3).
- The company became insolvent.
  This can cause a removal of the production system (EoL1/2/3).
- The company was taken over by another company.
  This can cause a removal of the production system (EoL1/2/3).
- A company buys the company producing a product.
  This can cause a relocation of the production system to the location of the new owner (EoL3).
- Avoid direct competition for the same product.
  This can cause a relocation of the production system to specific countries or other markets, e.g. Russia (EoL3).

Those reasons or drivers are in line with the EoL definition provided in Subsection 2.2.3. Due to the variety of those drivers a holistic consideration and systematic support of the EoL phase seems reasonable for the author.

After having answered the question why a production system potentially reaches its end of life, the next section will deal with the question which stakeholders are involved or can be found when the EoL phase was entered. Which stakeholders are somehow involved in the recovery process?

4.4 Business ecosystem of the End-of-Life phase of production systems

In this section a business ecosystem for the EoL phase is developed based on the interviewed experts from the case studies. Therefore, it cannot be exhaustive. The business ecosystem is divided regarding the physical asset or recovery goal (production system, component, and material), since stakeholders can be involved in more than one recovery process of a certain physical asset. The resulting business ecosystem for the EoL phase is shown in Figure 4.2. In this it is also indicated how the loop can be closed or what the future application could be.

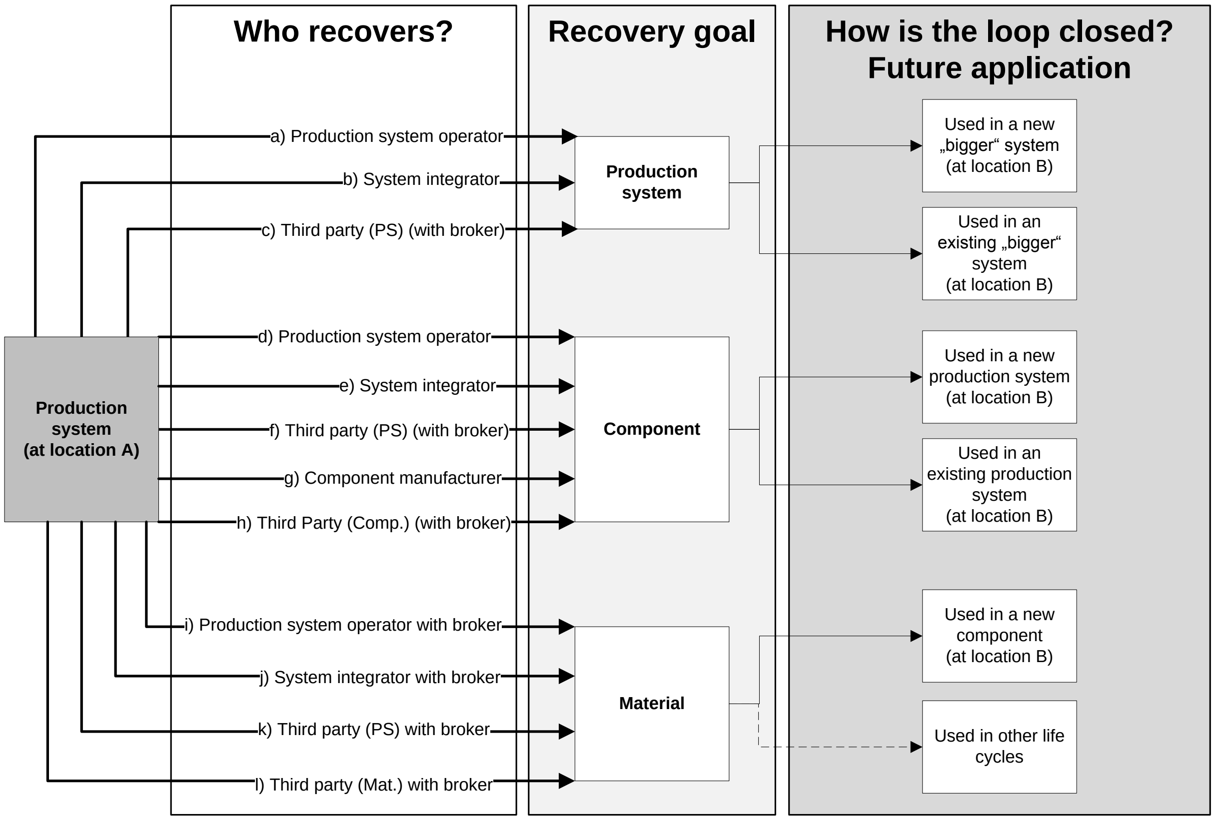

Figure 4.2: Business ecosystem in the End-of-Life phase

The *production system operator* is, here, the company that operates the production system or owns it. A *system integrator* is a company (and a service provider) that does the recovery process for the *production system operator*, but has knowledge about the production system and its components already before, because the *system integrator* and the *production system operator* have somehow a relationship. The *third party* is a company (and a service provider) that does the recovery process for the *production system operator*, but has no knowledge about the production system and its components, because the *third party* and the *production system operator* have no relationship. A *broker* can have different meanings, depending on the recovery process and physical asset. A *broker* can be seen as a 'service provider' for the both mentioned service providers. Some service providers need to have a *broker*, some can additionally have one. A *broker* is the interface between a service provider and the buying party of the recovered physical asset. Needless to say, that the hierarchy of the service providers (between the *production system operator* and the *broker*) can also be deeper than visualized in Figure 4.2. The *component manufacturer* is a company that had sold components to the *production system operator*.

Some recovery processes, characterized by the arrows in bold in Figure 4.2, correspond directly with a case study, i.e. it is described in Annex D. Other recovery processes correspond indirectly with the case studies, since the interviewed experts were giving the author their impression or understanding of the business ecosystem in the EoL phase.
The recovery processes associated to the recovery goals are briefly described in the following. On the one hand it is indicated which recovery process corresponds with which case study. On the other hand it is indicated where the author got the information from that a certain recovery processes exist. Needless to say, that also combinations of the recovery processes are possible (regarding the recovery goal or involved stakeholders). This is also addressed. A more detailed description of each stakeholder with its different recovery processes and potential availability of information can be found in Annex E.

4.4.1 Recovery goal: production system (EoL3)

The recovery of a production system basically means that it is decommissioned, disassembled, relocated, assembled, and finally commissioned. In the following the different arrows or business cases regarding production system recovery, which are shown in Figure 4.2, are explained.

The production system operator (a) does a relocation by itself, usually between the different locations of its company. Case study CS **IIIa** (see Annex D.3) and CS **VIa** (see Annex D.6) describe this recovery process.

The system integrator (b) does the relocation for the production system operator, also usually between the different locations of the contracting company. The system integrator has knowledge about the production system because it was engineered and built up by this company. Case study CS **I** (see Annex D.1) addresses this recovery process.

The third party/PS (c) is specialized on production systems. It can do the relocation of the production system either only for the production system operator or for the future production system operator (buying party). The relocation can, therefore, be done between the different locations of one company or between the locations of two different companies. In that case, it can be possible, that there is additionally a broker involved by the third party/PS. The broker would, then, try to find a potential buyer of the production system to be sold. After the broker has found a buyer, the third party/PS can start the recovery process. If no broker is involved the third party/PS needs to find a buyer on its own. Case study CS **VII** (see Annex D.7) and CS **V** as a broker (see Annex D.5) address this recovery process. Some recovery processes are done with or without the broker. But also the broker could be the first company in charge. That means that this time the buying party commissions the broker to find or to propose a proper third party/PS to do the relocation. This is addressed in case study CS **V** (see Annex D.5).

Dependent on the current state of the production system different EoL strategies are applied.
Basically, a recovery process focuses on one recovery goal. But when, e.g. a production system is relocated and needs to be remanufactured, this could result in a number of old, used, or defective components. In this case, component recovery and/or material recovery can be found as well.

4.4.2 Recovery goal: component (EoL2)

The recovery of components means that the production system has lost its purpose and is decomposed into its parts, which can be still used. In the following the different arrows or business cases regarding component recovery, which are shown in Figure 4.2, are explained.

The production system operator (d) does the disassembly of the components by itself and usually reuses them within the different locations of its company. Case study CS **IV** (see Annex D.4) describes this recovery process. But in this recovery process also a system integrator (e) can be involved. The reason could be, that special knowledge is necessary for decommissioning the component at location A and commissioning the component at location B. Therefore, a system integrator can be involved, since it had usually engineered and built up the component before.

The third party/PS (f) is specialized on the relocation of production systems. When it is commissioned to relocate the production system for a company and this also includes remanufacturing activities, it can happen that old components need to be replaced by new ones. In that case, those used components can be sold directly by the third party/PS or with a broker involved. Expert from case study CS **VII** (see Annex D.7) had provided the author this extension to the business ecosystem.

After the production system got decommissioned it is possible that the component manufacturer (g) is called by the production system operator and asked to remove its components. Usually the component manufacturer has sold exactly those components to the production system operator. After the component manufacturer has removed those components, it can sell them again. For example the robot manufacturer KUKA is doing this, see [KUK]. Expert from case study CS **I** (see Annex D.1) had provided the author this extension to the business ecosystem.

After the production system got decommissioned it is possible that a third party/Comp (h) is called by the production system operator and asked to remove certain components.

The third party/Comp is specialized in disassembling certain or, in general, components of one or more types. The third party/Comp can sell the components by itself or it cooperates with a broker. For example the company industrietec is disassembling and selling used components, see [ind10]. Expert from case study CS **V** (see Annex D.5) had provided the author this extension to the business ecosystem.

Dependent on the current state of the component different EoL strategies are applied. Basically, a recovery process focuses on one recovery goal. But when, e.g. a production system has lost its purpose and it should be decomposed into its useful and reusable parts, there are usually also no longer usable parts or waste. In this case, material recovery can be found as well.

4.4.3 Recovery goal: material (EoL1)

The recovery of material means that the production system has lost its purpose and there are no (more) reusable components existent. Then the shape of the remaining parts of the production system is dissolved. All of the stakeholders mentioned in this section cooperate with brokers. In this case the broker is not selling the production system or certain components, as the brokers from the previous two sections. Here, they are selling the materials, e.g. steel or copper. These brokers are scrap dealers. It is typical to have brokers involved in the recovery process since most of the companies don't have space to store materials. Another reason is, that the companies cannot easily transport the materials or scrap to, e.g. an ironworks, since the ironworks demands a constant delivery of a fixed amount of scrap. The crucial factor is constancy. Usually companies cannot fulfill this, because production systems do neither reach their end of life that often nor that constant, e.g. after a certain number of years. A steady waste stream can be found in the area of products or production waste. There is a higher frequency and constancy regarding the EoL phase. In the following the different arrows or business cases regarding material recovery, which are shown in Figure 4.2, are explained.

The production system operator (i) does the decommissioning and disassembling of the production system by itself. The production system is disassembled as much as it is necessary to fit onto a truck. Then, it can be transported to the broker. Case studies CS **IIIb** (see Annex D.3) and CS **VIb** (see Annex D.6) describe this recovery process.

The system integrator (j) is specialized on the relocation of production systems. When it is commissioned to relocate the production system for the production system operator and this also includes remanufacturing activities, it can happen that used materials and waste is the result. Those are, then, transported to a broker. It is also possible that the system integrator is commissioned to engineer and build up an entirely new production system, but before, the system integrator needs to remove the old one. In any case, the system integrator has knowledge about the old production system, since it has engineered and built up it. Expert from case study CS **I** (see Annex D.1) had provided the author this extension to the business ecosystem.

The third party/PS (k) is specialized on the relocation of production systems. When it is commissioned to relocate the production system for a company and this also includes remanufacturing activities, it can happen that used materials and waste is the result. Those are, then, transported to a broker. Expert from case study CS **VII** (see Annex D.7) had provided the author this extension to the business ecosystem.

The third party/Mat (l) is specialized in the removal of and material extraction from production systems and does the disassembly by itself and that as much as it is necessary to get the material sorted. It is commissioned to remove the decommissioned production system for the production system operator. The unmixed scrap is transported to the broker. Case study CS **II** (see Annex D.2) describes this recovery process.

Table 4.2 associates the case studies to the business ecosystem of the EoL phase (from Figure 4.2), in particular to the stakeholders with their recovery process. As mentioned before 'direct' means that there is a recovery process described in Annex D. 'Indirect' means that the author was informed by the interviewed expert about the existence of that stakeholder.

Table 4.2: Case studies associated to the stakeholders of the business ecosystem of the End-of-Life phase

Who recovers? Stakeholder	Case study No.
a) production system operator	direct: CS **IIIa**, CS **VIa**
b) system integrator	direct: CS **I**
c) third party (PS) (with broker)	direct: CS **VII** (CS **V** as broker)
d) production system operator	direct: CS **IV**
e) system integrator	direct: CS **IV**
f) third party (PS) (with broker)	indirect: CS **VII**
g) component manufacturer	indirect: CS **I**
h) third party (Comp.) (with broker)	indirect: CS **V**
i) production system operator with broker	indirect: CS **I**
j) system integrator with broker	indirect: CS **VII**
k) third party (PS) with broker	direct: CS **VII** (CS **V** as broker)
l) third party (Mat.) with broker	direct: CS **II**

As mentioned in Chapter 2 the life cycle phases are interconnected. In Chapter 3 it was shown how the EoL is interconnected. Figure 4.3 provides now an overview of the interconnectedness of the EoL phase in the business ecosystem (based on the previous elaborations of this thesis). In this figure the input and output flow of EoL relevant artifacts for the production system life cycle are shown on the one hand. On the other hand it is mentioned, which stakeholder creates or consumes which artifact. The 'backbone' is the production system with its physical and informational manifestation. This is in line with the Figure 3.8 that had shown the connection of the physical and digital world of a production system based on information technologies.

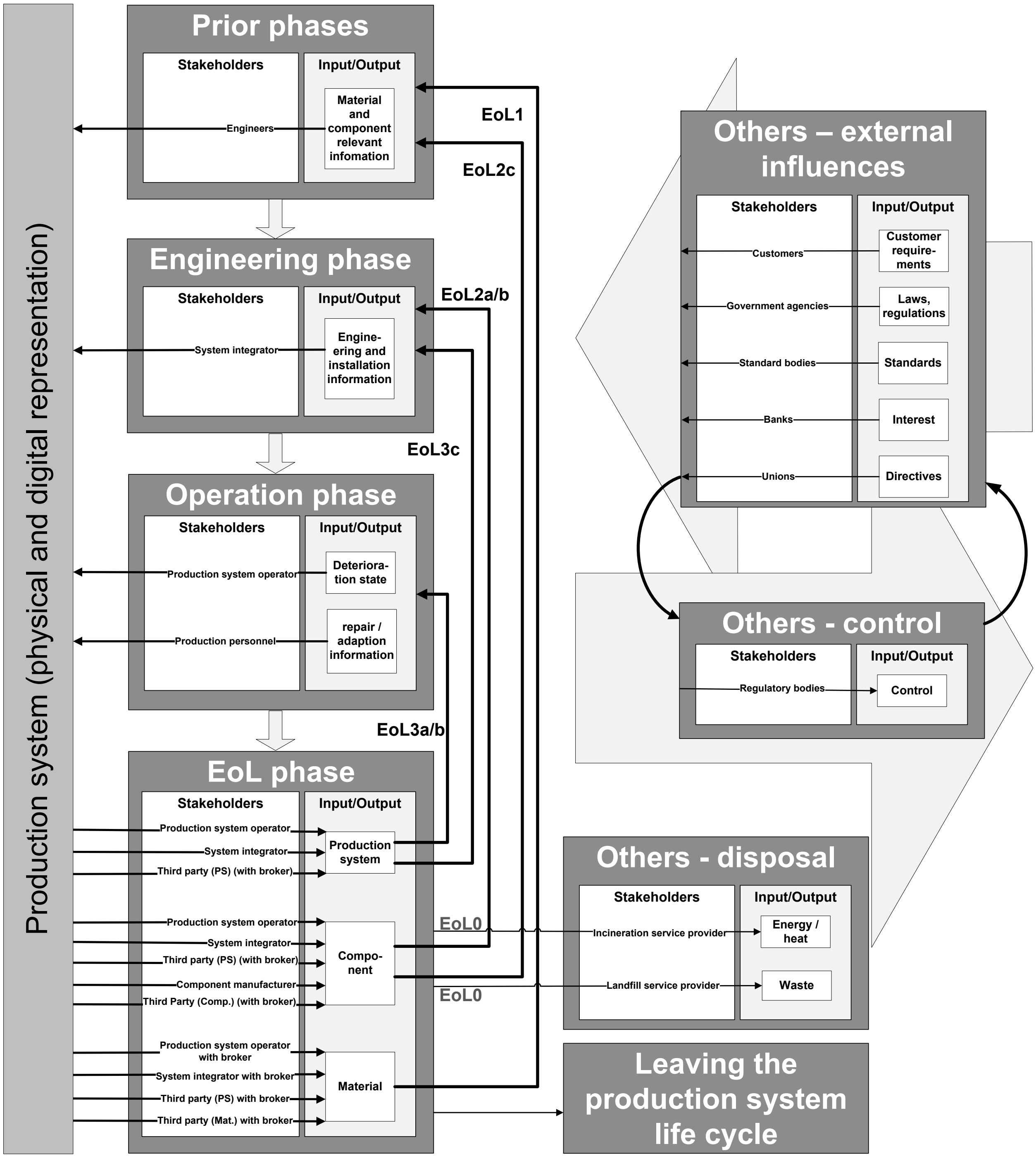

Figure 4.3: Overall business ecosystem with focus on End-of-Life phase

Also other influences on the overall business ecosystem of a production system are shown in Figure 4.3. These are external influences, like directives and laws, and also regulatory bodies that could control the fulfillment of certain laws. The in this thesis not further considered EoL0 - no recovery (introduced in Chapter 3) is depicted here for the sake of completeness. It is also indicated that a production system, a component, or a material can leave the production system life cycle, which is the basis of this thesis. For example, when recovered aluminum is not used to produce components for a production system. Instead it is used to produce cars, then this material has left the production system life cycle.

This section had shown that a lot of different stakeholders are involved in the EoL phase of production systems (which can also cooperate with each other). The author was able to describe some of the stakeholders with their recovery processes very detailed in the case studies (see Annex D). Each of them needs certain documentation of information (dependent on the EoL scenario) and has different access regarding how the production system is built up and what its current/“as-is” state is. If the necessary information would be provided or would be made available or exchangeable, it would have a huge impact on the work of those stakeholders. Also new business models could be developed for the EoL phase or new stakeholders could show up in the EoL phase.

The next section will now analyze the legal basis for the recovery of production systems in Germany. It is focused on EU directives and German laws with a direct linkage to production systems as their life cycle is of main interest in this thesis.

4.5 Legal basis for production system recovery in Germany

This section analyzes if there are documents (EU directives or German laws) available that could assist or restrict the decision maker in the decision about a proper treatment or EoL scenario of the production system. After that, it is analyzed, who has the responsibility to take care of the production system at the end of its life.

4.5.1 Directives and laws

Since the literature review had basically resulted in a number of publications relevant to products (like cars, washing machines, or mobile phones), only laws and regulations on products are mentioned in those publications. These are analyzed in the following.

The EU directive 2012/19/EU on WEEE (Waste Electric and Electronic Equipment) [Off12] is transposed into the German law called ElektroG (Gesetz über das Inverkehrbringen, die Rücknahme und die umweltverträgliche Entsorgung von Elektro- und Elektronikgeräten) [Fed15b]. The ElektroG shall apply e.g. to whiteware or domestic appliances, IT devices, electric and electronic tools, control and monitoring instruments, and photovoltaic modules. However, it shall not apply to (not the complete list) [Off12]:

- “large-scale stationary industrial tools: means a large size assembly of machines, equipment, and/or components, functioning together for a specific application, permanently installed and de-installed by professionals at a given place, and used and maintained by professionals in an industrial manufacturing facility or research and development facility”
- “large-scale fixed installations, except any equipment which is not specifically designed and installed as part of those installations: means a large-size combination of several types of apparatus and, where applicable, other devices, which are assembled, installed and de-installed by professionals, are intended to be used permanently as part of a building or a structure at a pre-defined and dedicated location, and can only be replaced by the same specifically designed equipment”
- “on-road mobile machinery made available exclusively for professional use: means machinery, with on- board power source, the operation of which requires either

mobility or continuous or semi-continuous movement between a succession of fixed working locations while working"

This means that neither the directive nor the law do apply to production systems and their components.
This directive has a more general view on products and their recovery. But there are also more specific directives or laws available, which deal with recovery of products like the EU directive 2006/66/EG on batteries and accumulators [Off06a] that is transposed into the German law called BattG (Gesetz über das Inverkehrbringen, die Rücknahme und die umweltverträgliche Entsorgung von Batterien und Akkumulatoren (Batteriegesetz)) [Fed15a]. Or the EU directive 2000/53/EC on end-of life vehicles [Off00] that is transposed into the German act called AltfahrzeugV (Verordnung über die Überlassung, Rücknahme und umweltverträgliche Entsorgung von Altfahrzeugen (Altfahrzeugverordnung)) [Fed16c].
When looking explicitly at specific production systems (nuclear, wind, or solar power plants), laws or guidelines can be found. For the recovery of nuclear power plants, for example, there is a guideline available for safe decommissioning, disassembly of plants or parts of it [The16]. This guideline is based on the German law called AtG (Gesetz über die friedliche Verwendung der Kernenergie und den Schutz gegen ihre Gefahren (Atomgesetz)) on the Atomic Energy Act [Fed17].
However, those specific production systems are out of the scope of this thesis. The author has not found any directive or law that is applicable to production systems in general. Therefore, it is again looked at the publications relevant to products. Here a directive that is more general than the already described ones is the 2008/98/EC. The EU directive 2008/98/EC on waste [Off08] is transposed into the German law called KrWG (Gesetz zur Förderung der Kreislaufwirtschaft und Sicherung der umweltverträglichen Bewirtschaftung von Abfällen) [Fed16b]. It deals with the support of a circular economy to save natural resources and to ensure the protection of humans and environment when producing and managing waste. However, the scope is the waste pyramid that says avoiding waste before recycling waste before disposing waste. Waste is defined according to [Fed16b] as objects that the owner gets rid of, wants to get rid of, or needs to get rid of. This happens when the original purpose is no longer fulfilled or is no longer required. Out of scope is the reuse of waste.

All in all, there are no assisting or restricting directives or laws available that would influence the decision on a proper EoL scenario in the first place, i.e. considering the production system as the input for the recovery process does not come with regulations. But when the EoL scenarios are considered and decomposed again into recovery goal (future application) and EoL strategy (process), directives or laws may come to light.
When looking at the recovery goals, i.e. the future application of the physical asset, the focus shifts towards the second/third etc. life of production system, component, or material (i.e. the following operation phase or reuse of production system, component, and material). Which directives or German laws are available that provide regulations on the operation of a production system or component, or on the reuse of material? This is an important fact since the reused production system, component, or material might not meet those regulations anymore, because directives or laws may have changed. This is also important when the recovered physical asset will be exported to another country that has different laws. But this is out of scope of this thesis. In this thesis it is assumed that the recovered physical asset is used again in Germany. Otherwise laws

of the country, to which the physical asset is exported, need to be considered as well as customs and import restrictions.
The EU directive 2011/65/EU on RoHS (Restriction of Hazardous Substances) [Off11] restricts the usage of harmful substances and materials in electric and electronic equipment but comes with the same scope like the WEEE. This means that this directive does not apply to production systems and their components.
The EU regulation EC 1907/2006 on Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) [Off06b] says that producer, importers, and the following users of chemicals take the responsibility for them. They need to provide information about their chemicals to ensure a safe handling of them [Eur16]. This directive applies when the production system and component contains harmful chemicals.
The German law called BImSchm (Bundes-Immissionsschutzgesetz) on Federal Immission Control Act [Fed16a] regulates the protection of humans, animals, plants, soils, water, atmosphere, and cultural properties against immissions and emissions. The law applies for the erection and commissioning as well as operation of production systems one the one hand. And on the other hand for assembling, placing on the market, and importing of production systems. With production system BImSchm understands factories and other fixed installations as well as machines, devices and other mobile, technical equipment. This law applied for the operation of production systems and components. And with operation BImSchm includes also the preceding erection and the subsequent decommissioning.
The German act called AVV (Abfallverzeichnisverordnung) on waste declaration [Fed12] is used to declare waste and to classify the waste regarding its harmfulness. For material recovery it is useful to make a distinction between harmful materials and not harmful materials.
This list is not exhaustive. There are more directives and German laws available, also dependent of the type of production system, see [Web08], [AS15], or [OHK10].

When looking at the EoL strategies, i.e. the process needed to recover the physical asset, the focus shifts towards the recovery process itself. Which directives or German laws are available that provide regulations on the recovery process of a production system, component, or material?
The EU directive 2012/19/EU on packaging and packaging waste [Eur94] is transposed into the German act called VerpackV (Verordnung über die Vermeidung und Verwertung von Verpackungsabfällen) [Fed14b] will be replaced by the German law VerpackG in 2017. The VerpackV is applied to all packages independent from if those packages are used or generated in industry, administration, commerce, service industry, or household. When transporting the recovered physical asset and there is a need to pack it, this law is applied.
The German law called ArbSchG (Arbeitsschutzgesetz) is the German Occupational Safety and Health Act [Fed15c]. Its purpose is to ensure or improve the workers safety and health at work. As there are people involved in the recovery process, this is an important law that needs to be considered and complied with.
The German law called KrWG [Fed16b] on waste provides also restrictions on disposing and recycling techniques.
This list is not exhaustive. There are more directives and German laws available, also dependent of the type of production system, see [Web08], [AS15], [OHK10], [VDI14], [VDI16b], [VDI01b], or [Däh17].

After having discussed possible EU directives and German laws, which can have an effect on the decision for a proper treatment or EoL scenario for a production system at the end of its life, and after having mapped those documents correspondingly to the recovery process (see Figure 4.4), the next section will analyze who has the responsibility to take care of the production system at the end of its life.

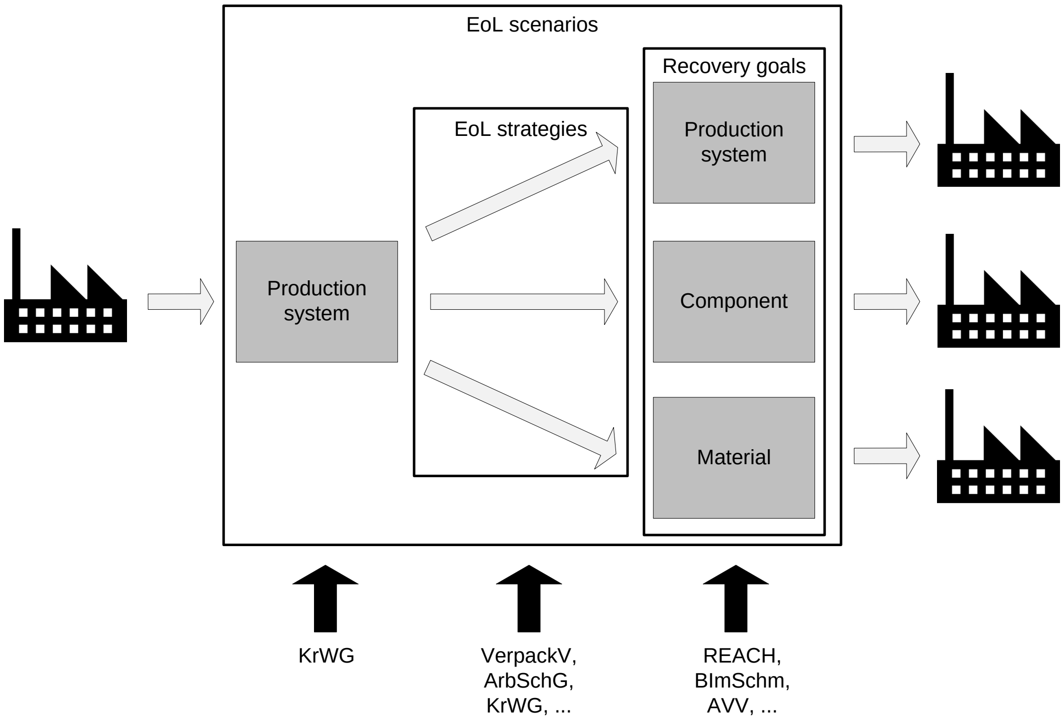

Figure 4.4: Location of the EU directives or German laws in the recovery process

4.5.2 Responsibility

The EU directive 2008/98/EC on waste [Off08] has introduced the Extended Producer Responsibility (EPR). It is “an environmental policy approach in which a producer’s responsibility for a product is extended to the post-consumer stage of a product’s life cycle” [BIO14]. That means that the producer takes back the used products and is responsible to recycle and dispose them in an environmentally acceptable manner.
The German law called KrWG [Fed16b] on waste provides basically a general framework to support a circular economy. It is interlinked with the EU directive 2008/98/EC. Therefore, it also discusses the product responsibility. It says that a product shall be designed in such a way that waste is generated as less as possible during production and usage or operation. And that the product shall be recycled or disposed environmentally acceptable at its end of life. This product responsibility comprises in detail e.g. that recycling shall be preferred over disposing, that harmful substances shall be declared, that take back, recycling, or reuse options shall be named, or that the product shall be taken back when it has reached its end of life which needs to be, then, recycled or disposed in an environmentally acceptable manner.
Based on the general KrWG some, more specific German laws and acts have already integrated this, e.g. ElektroG or AltfahrzeugV. However, this approach regarding responsibility is basically considering a business-to-customer case (B2C), where there

responsibility is not transferred to the customer. In the product area it might be often the case, when thinking about washing machines or cars. But production systems are usually sold between businesses (B2B). Therefore, the responsibility is transferred between the businesses, i.e. when a company buys a production system it is responsible for its recycling and disposal. This was confirmed by the experts the author had interviewed for the case studies (see Annex D).

4.6 Conclusion and current limitations

This chapter has analyzed the EoL phase of production systems from a case study based point of view, due to the lack of publications with this scope. It was possible to get insights by interviewing experts, also about the manifold reasons, why a production system can reach its end of life. Thereby, the literature based elaborations from Chapter 3 could be extended in order to get a holistic picture of the EoL phase of production systems.

The in Section 3.2 identified recovery goals could be confirmed through the case studies (see Annex D). At least two case studies per recovery goal (production system, component, and material) could be developed. In the interviews the author did not explicitly ask for the EoL strategies. The experts could decide, which recovery process or EoL scenario with which EoL scenario they would like to describe. But they have confirmed the other possible treatments or EoL strategies. The decomposition of the corresponding EoL strategy into processes (e.g. decommissioning or disassembly) can be also found in those descriptions in Annex D.
The decision on the EoL scenario or in particular on the EoL strategy, is basically made by the production system operator (or owner) or customer. It comes up with certain requirements on the future application of the physical asset (the recovery goal). But as the recovery process is mainly done by service providers and not by the production system operator (or owner) or customer itself (see the business ecosystem in Section 4.4), the service providers need to identify an appropriate EoL strategy with which these requirements can be met. As mentioned before, the recovery goal is mainly chosen by the production system operator (or owner) or customer. But it could also be the case that the customer leaves the service provider free to decide what will happen with the production system, e.g. when the production system operator (or owner) just wants to get rid of the production system. In this case the service provider needs to decide about the recovery goal as well.

The recovered component from case study CS **IV** CS **V**, and CS **VII** can be associated to the function group level - based on [LSH+17b]. A function group is, here, the technical realization of one value adding or auxiliary function necessary for the smallest non divisible process [SLH+16], like a robot with a gluing gun or an air condition. At least in these two case studies the 'function group' component seems to be best recoverable since most flexible and, therefore, integrable into other production systems. Recovered components could be considered as 'function providers' as proposed by [LSsB06], so the function or a certain process is recovered.
In case of the recovery of production systems (see case study CS **I**, **IIIa**, **VIa**, or **VII**) the product is in focus, which can be produced by the production system. So the production of certain products is recovered.

The recovered material from case study CS **II**, **IIIb**, **VIb**, and **VII** can be used as input for the production of products or to produce new production system components, resources, or equipment.

According to Section 4.5 the production system operator (or owner) has the responsibility to properly treat the 'waste'. Since there are no strict directives or laws for the recovery of production systems (according to the scope of this thesis), the production system operator (or owner) has basically all options or recovery goals available.
Based on the detailed process descriptions, the recovery process can be considered as a project (as it can also be seen in [Däh17] or similar to [VDI16a]). A general recovery process can, therefore, have the following phases:

- **Project preparation:** This phase includes the inspection of the production system, the examination of the available documents, the estimation and determination of the value of the production system and costs for the recovery process, and the collection of requirements on the future application of the physical asset.
- **Project planning:** This phase includes the detailed planning of the recovery process to finally get the recovered physical asset at the right location for the right costs in the right quality in the right time. EoL strategy is chosen. All the requirements on the future application of the physical asset need to be considered and met in this phase.
- **Project execution:** This phase includes the execution of the project plan.
- **Project completion:** This phase completes the project.

Figure 4.5 depicts the general recovery process with its phases and activities to recover a physical asset.

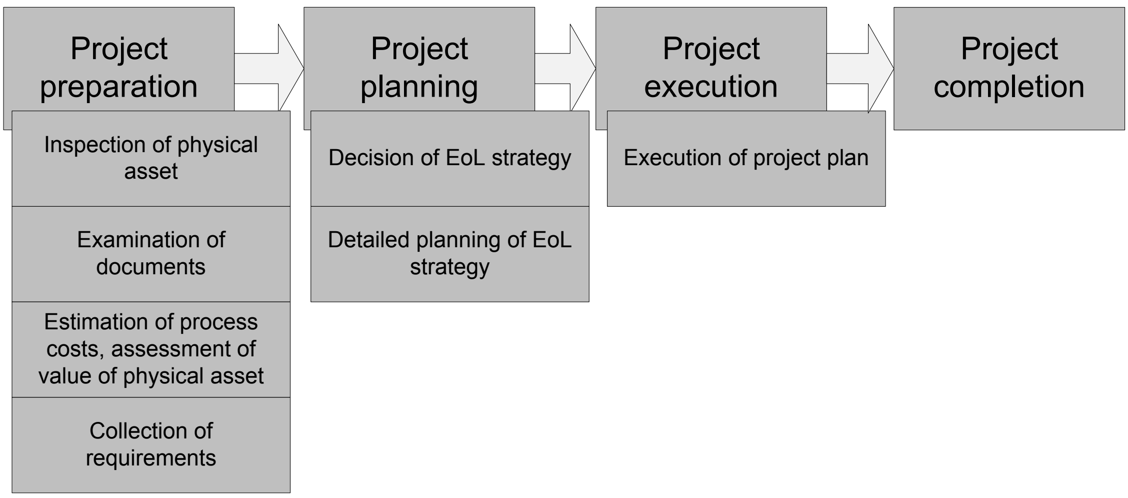

Figure 4.5: General recovery process with phases and activities

In the project preparation phase an offer needs to be created for the customer that contains the estimated value of the production system and the costs of the service provider to the entire recovery process. This offer needs to be accepted before the project planning phase can start. Therefore, a thorough inspection of the physical asset and examination of the documents as well as a holistic consideration of the entire recovery process is crucial.

The necessity of the availability of information from the life cycle phases before the EoL phase as well as EoL relevant information (addressed in Section 3.3) was confirmed through the detailed process descriptions (see Annex D). Since the project planning phase is similar to an engineering process, the readability of the documents about the production system from the previous life cycle phases is crucial. These often come as papers or PDFs. The variety of used software tools can be taken also from the detailed process descriptions in Annex D.

The author thinks that it would be a valuable tool if the decision maker (e.g. one person of a service provider) is supported with a method that allows such a holistic consideration of the recovery process. So that a decision can be made about a proper EoL scenario for the production system. This could shorten the project preparation phase. When consulting the given time information from Annex D, this phase takes usually one to two months. This would be the range for optimization.
Assuming that the digital representation of the production system that comprises information from the engineering and operation phase (as proposed in Section 3.5) is available, the inspection of the physical asset and the examination of the documents could be omitted. When this could be done automatically, time and money are saved on the one hand and on the other hand potentially man-made mistakes are avoided.
The assessment of the value of the physical asset, if it is able to meet the requirements on the future application of it, could also be partially automated, when the requirements or more precisely the technical characterization can be made comparable with the technical characterization of the physical asset to be recovered.
Assuming that the decision maker would have access to the digital representation of the production system and has all the EoL relevant information that were used in Section 3.3 to create the EoL data model, the decision maker would be able to assess and characterize the entire recovery process including its costs and would be, therefore, able to prepare and propose a precise offer to the customer.
This is the scope of this thesis and is intended to be supported by the mentioned method that should enable the decision maker to decide about an appropriate EoL scenario for the production system. Figure 4.6 visualizes again the recovery process phases with the corresponding activities that are subjects to be supported by the results of this thesis (see black boxes). It is also highlighted, where one life ends and where the new life of a production system begins (vertical alignment). The EoL business ecosystem is additionally mapped onto this figure to have the results of this chapter at hand.

This method - called recovery planning method (RPM) - will be now developed in the next chapter. With the EoL data model developed in Section 3.3, companies or stakeholders involved in the EoL phase of production systems could develop new business models and software tools. The business ecosystem from Section 4.4 shows the variety of different stakeholders which could potentially benefit from the results of this thesis.

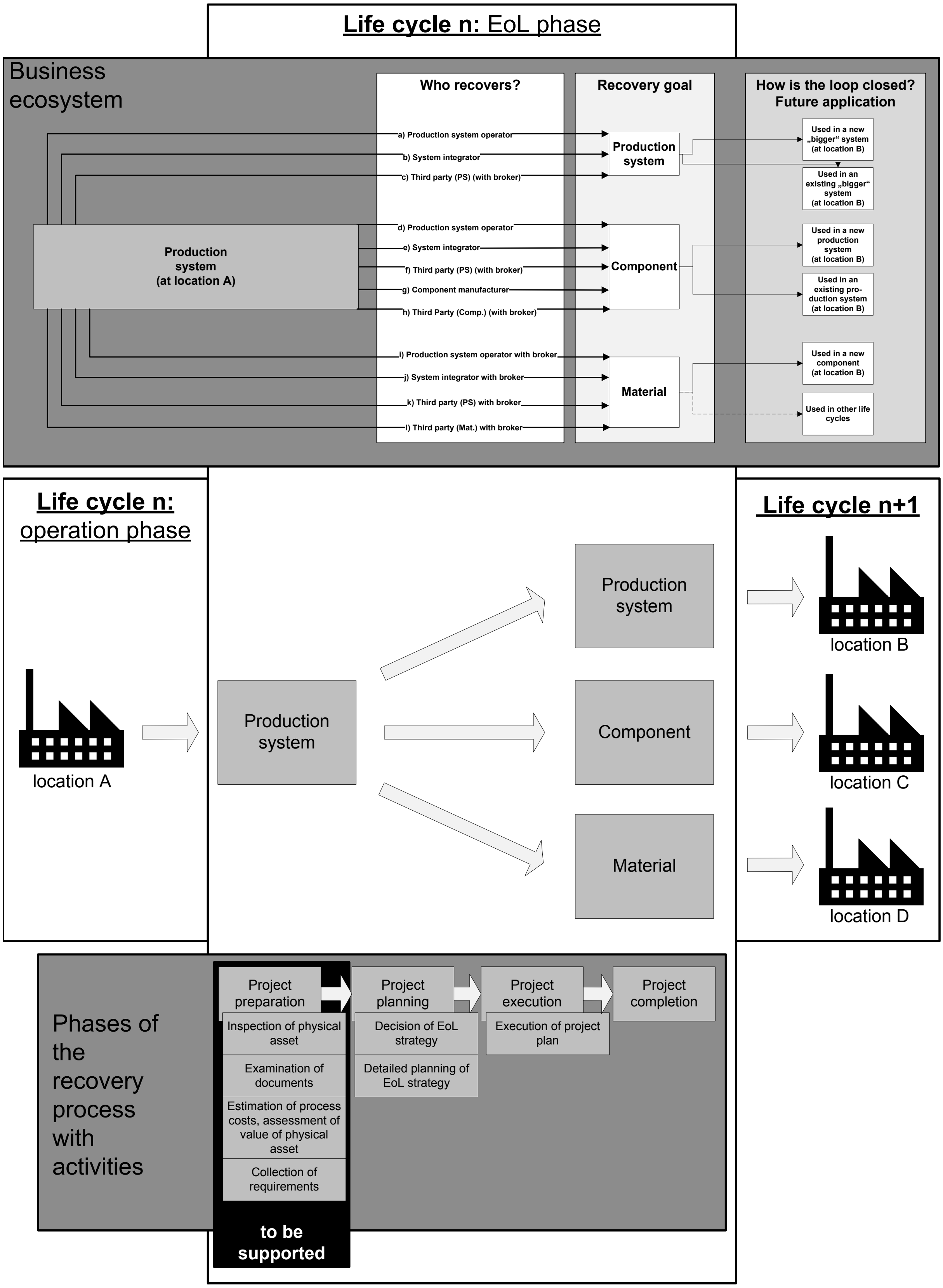

Figure 4.6: To be supported: project preparation

5 Recovery planning method for recovery support

The inductive scientific approach of this thesis shall result in a holistic consideration of the production system at the end of its life and, therefore, in a generalization of it by having aggregated different use case specific publications and case studies. This also applies for the Recovery Planning Method (RPM) to be developed in this chapter. By developing a general method for the recovery process of production systems, it is the author's intention to provide experts a means to face the EoL phase in a systematic manner.
Needless to say, that when applying a general method it needs to be adapted (reduced and/or extended somehow) according to the use case. As shown in Chapter 3 some product related publications might already fit certain use cases for the production system recovery. But in case of new use cases, this thesis and this RPM can be a good starting point.

The main goal of the RPM is to support the project preparation phase of the recovery process and to make EoL scenarios comparable so that a decision maker is able to make an informed decision about the best EoL treatment or scenario for a certain production system. The following activities of this phase should be, therefore, supported:

1. Inspection of physical asset
2. Examination of documents
3. Estimation of process costs, assessment of value of physical asset
4. Collection of requirements

Therefore, the RPM provides a holistic view on the EoL phase or recovery process of production systems. So, that a decision maker is able to think through the recovery in its entirety.
For applying the RPM, however, production system data from its life cycle is necessary. This data stored in the EoL data model will be addressed first in the next section, before the RPM will be explained .

5.1 End-of-Life data model

This section analyzes the EoL data model that was developed in Chapter 3. The goal is to develop a generalized, unified, and domain independent EoL data model, which is in line with the inductive scientific approach of this thesis.
This section is structured as follows: At first, the base EoL data model is generalized. This data model comprises all the information that needs to be provided by the life cycle of the production system and serves as input for the EoL phase. It describes the physical asset - the production system with all its components and materials.
Secondly, the EoL process data model is analyzed. This data model describes the recovery process executed in the EoL phase. Here, product, process, and resource

related information is needed. Usually this information is gathered or generated right at the time when the recovery process is executed. However, some information from the base EoL data model is necessary as input, since the recovery process recovers the physical asset. In the end, both data models are combined to the EoL data model to show, which information the data model finally needs to store.
Third, the indices are given on which basis the system as well as the EoL strategy can be characterized. This is required to make EoL scenarios comparable.

5.1.1 Feed forward: Base End-of-Life data model

This subsection analyzes the base EoL data model shown in Figure 3.3, which contains necessary information from the prior phases of the production system life cycle. It gives some indication of the physical asset and, therefore, of the recovery goal, i.e. 'what' can be possibly recovered from the production system. This is a characterization of the system.

In a first step, the attributes of the base EoL data model are grouped. For this, the proposals for an appropriate grouping from [Foe13] and [HSF+17] were combined resulting in the groups: type, technical characteristics, instance information, and measured data. Type indicates, which capabilities a system, process, or resource has. Technical characteristics give details on the physical realization of those capabilities. Instance information contains here mainly organizational and identifying instance related information. Measured data are technical characteristics that are measured during operation.

Figure 5.1 shows the grouping of the classes of the base EoL data model (see Figure 3.3). The base EoL data model contains this information, which needs to come from the engineering phase and operation phase in order to enable a proper EoL phase. When looking at the EoL process data model (see Figure 3.4), which contains information that usually needs to come from the EoL phase itself, it is obvious that some of this information needs to be also provided by the engineering and/or operation phase, so that a proper EoL phase can be enabled. In the case of the classes *production system* and *component* this is the 'life expectation', 'weight', and 'size'. For *material* it is 'life expectation', 'weight', 'size', and 'technical characteristics', for the class *connection* 'technical characteristics' and 'working surface', and for the class *memory* nothing else is added.

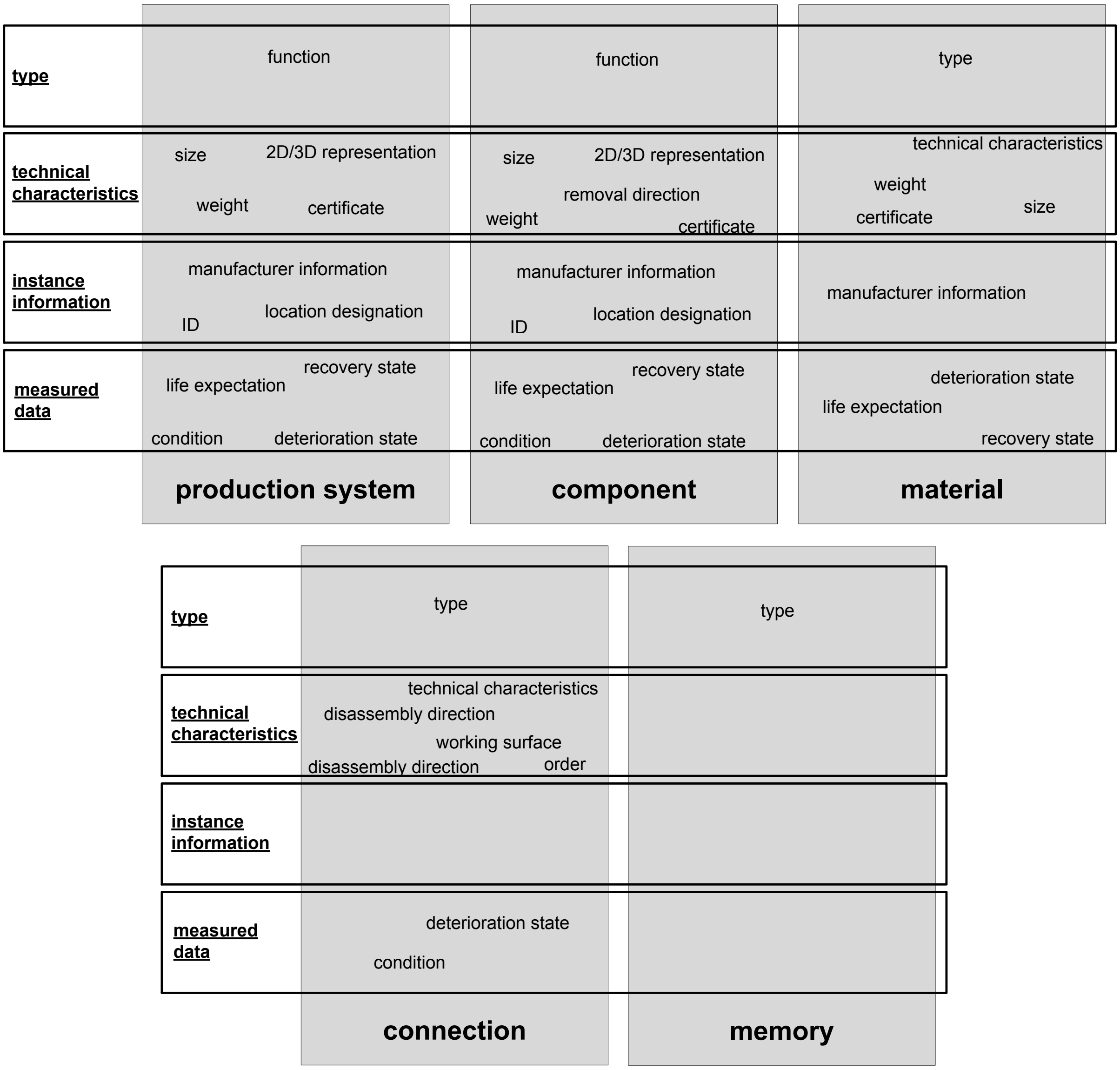

Figure 5.1: Addition and grouping of the attributes of the classes of the base End-of-Life data model

A detailed description of the attributes of these classes of the base EoL data model can be found in Annex F.1. The shown attributes in this subsection are based on the literature review addressed with Chapter 3. It is, therefore, not an exhaustive list.

The relations between the classes remain unaffected. Therefore, the generalized base EoL data model is as given in Figure 5.2.

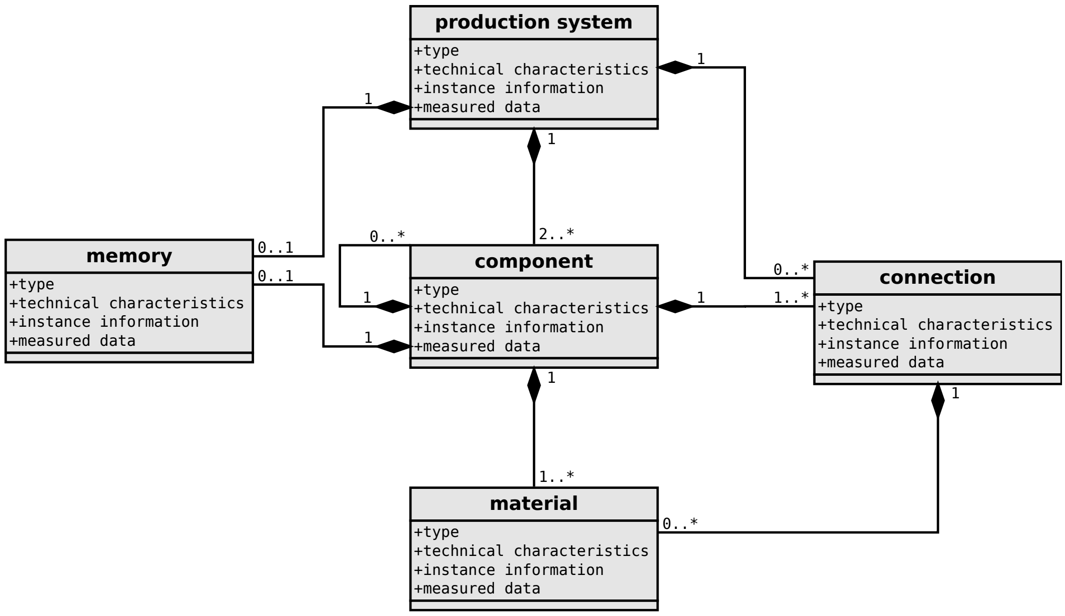

Figure 5.2: Base End-of-Life data model (generalized)

After having found a proper grouping for the attributes of the classes of the base EoL data model as well as having identified additional information as input for the EoL phase, the next subsection will deal with the EoL process model.

5.1.2 Feed within: End-of-Life process data model

This subsection analyzes the EoL process data model shown in Figure 3.4, which contains the needed information from the EoL phase itself. It gives some indication of the recovery process and, therefore, of the EoL strategy, i.e. 'how' a physical asset can be possibly recovered from the production system. This is a characterization of the EoL strategy.
Based on the EoL process data model and the base EoL data model from the previous subsection, the EoL data model can be finally developed.

The EoL process data model is structured according to the PPR concept. In this subsection this is used and appropriately adapted regarding the needs of EoL phase. Here, the product represents the production system, component, or material, since this is the system that is treated somehow by a process. A process converts input into output, which generally is the recovery process in this context. Resources are necessary to physically realize a process, e.g. a screwdriver is needed to undo a screw connection to get a component disassembled. It is obvious that the recovery process is dependent on the EoL scenario.

Product part: the first P of the PPR concept According to the literature review done in Chapter 3 not all attributes of the classes of the base EoL data model can be found in the EoL process data model. However, additional EoL process specific attributes are needed. The following Figure 5.3 shows this overlap of information that both data models share. In addition, the attribute *harmfulness* was added to the classes *production system* and *component* to have the possibility to express the harmfulness of

a component, like a spring under tension or an airbag.
Since the EoL data model (see Figure 3.5) is the sum of the base EoL data model and the EoL process data model, Figure 5.3 shows all the attributes of the product related classes of the EoL data model.

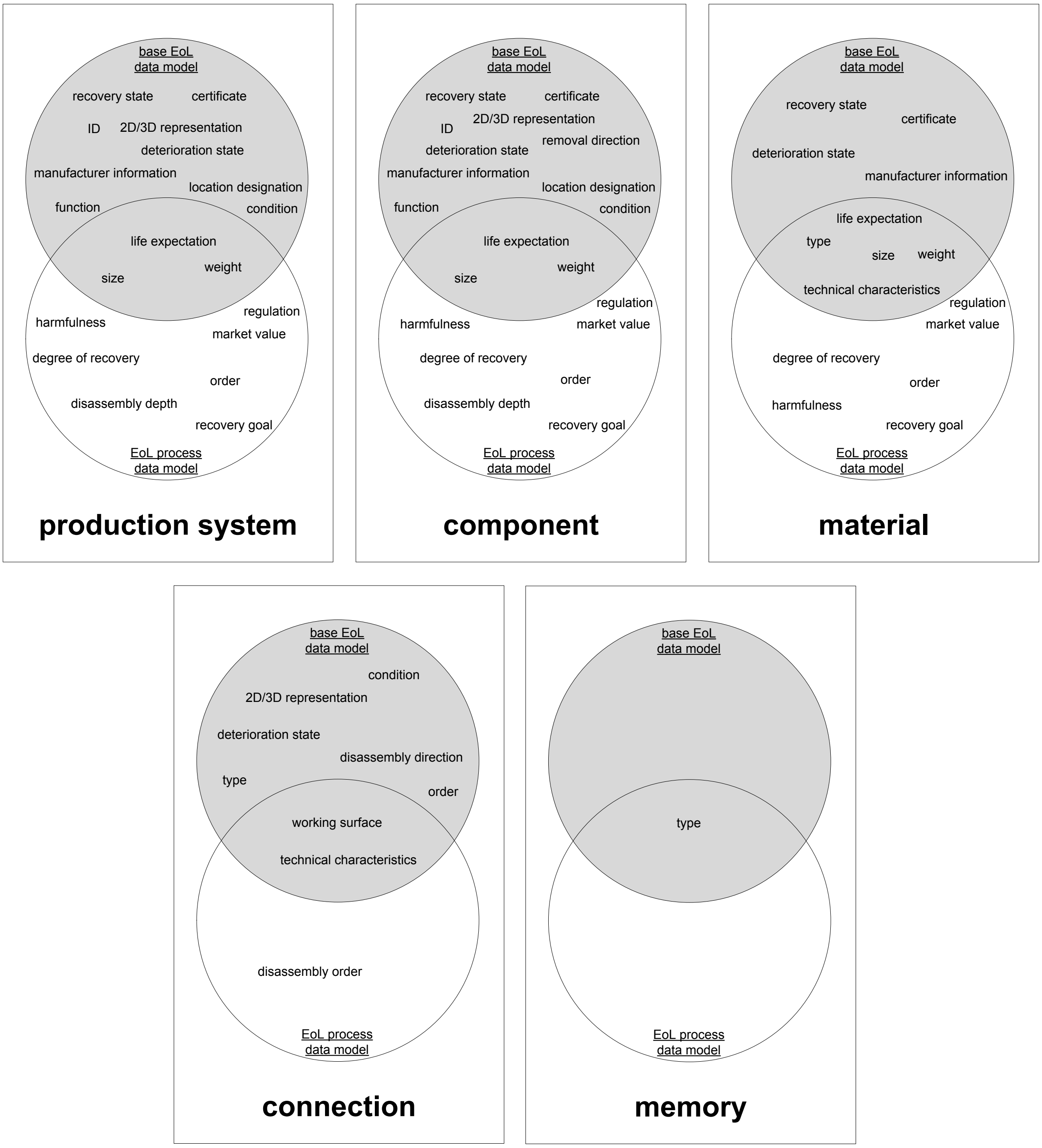

Figure 5.3: Overlap of information of the base End-of-Life data model and the End-of-Life process data model in the five classes of the base End-of-Life data model

When looking at these attributes of the EoL process data model, they can be grouped into product related information and process related information.
The first group contains attributes from the base EoL data model. But at the same time, it adds product related attributes necessary for the recovery process - see Figure 5.4.
The second group means that the product provides information that is necessary for the

recovery process of itself. But those cannot be provided by the life cycle of the product. They need to be gathered or created at the point in time when the decision about the EoL scenario is needed.

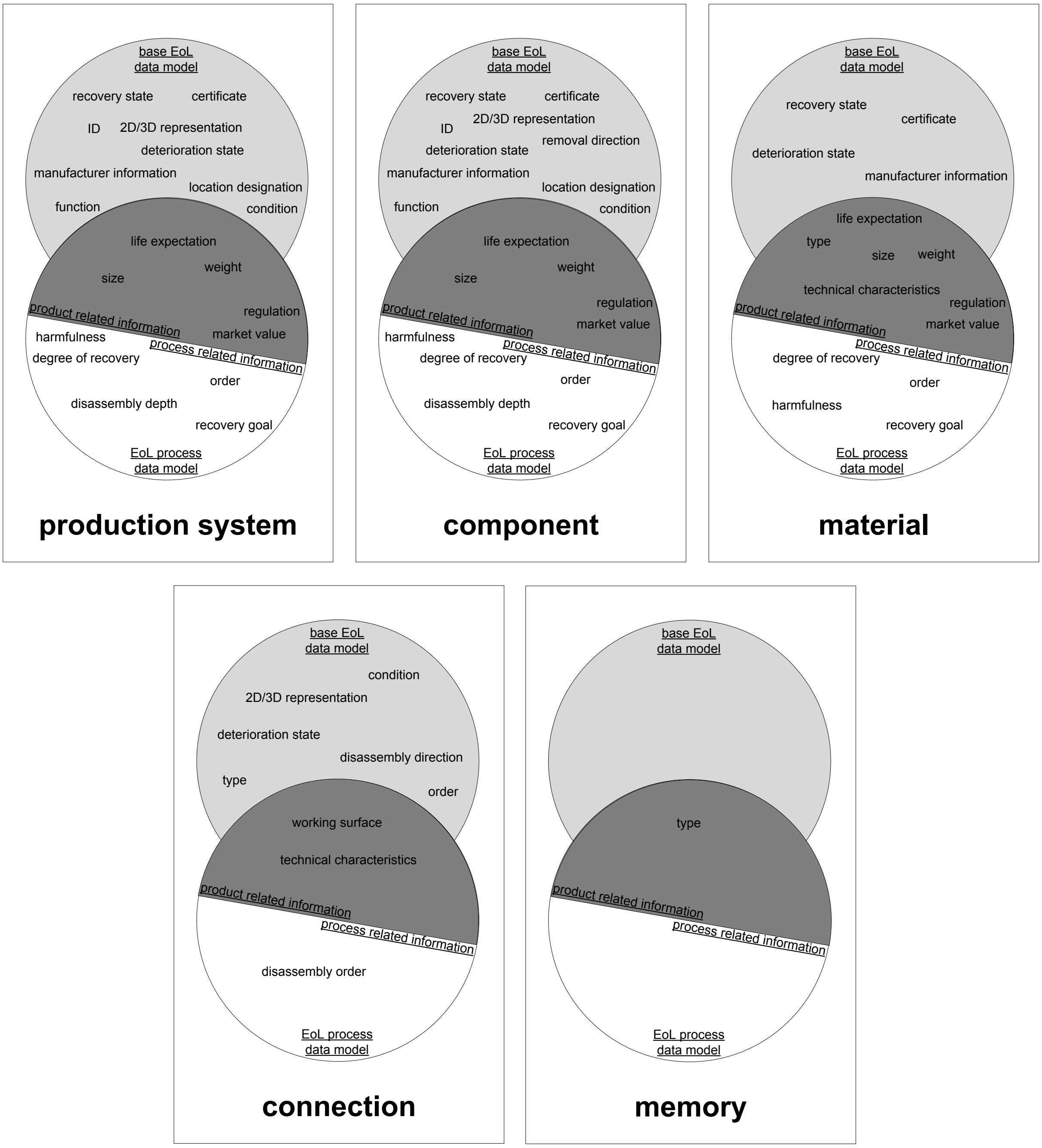

Figure 5.4: Highlighting product and process related information

A detailed description of the attributes of these classes of the EoL process data model can be found in Annex F.2. The shown attributes in this paragraph are based on the literature review addressed with Chapter 3. It is, therefore, not an exhaustive list.

After having found a proper grouping for the attributes of the product related classes of the EoL process data model, the next paragraph will deal with the process related classes of the EoL process data model.

Process part: the second P of the PPR concept The literature review done in Chapter 3 led to twelve process classes - all with a different number in their attributes. This is due to scope of the publications the author has read. The only attribute that all the classes have in common is *type*. But when looking at the attributes, more common attributes can be found - see Figure 5.5. The author thinks that all attributes gathered through the literature review can be considered as common attributes and, therefore, applied for each process class. A grouping into product, process, and resource related information seems reasonable.

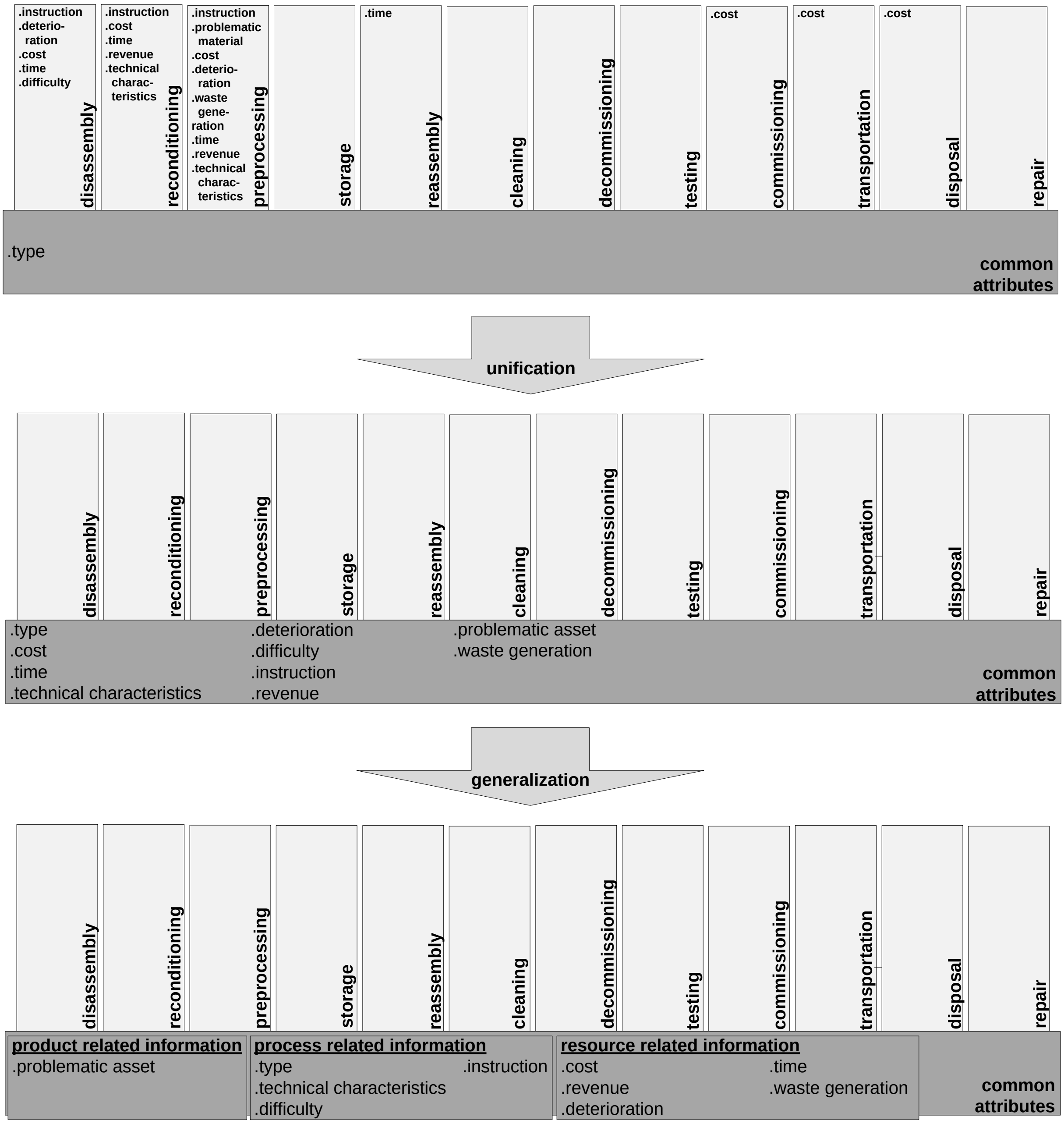

Figure 5.5: Attributes of the process classes of the End-of-Life process data model (unified and generalized)

A detailed description of the common attributes of these classes of the EoL process data model can be found in Annex F.2. The shown attributes in this paragraph are based on the literature review addressed with Chapter 3. It is, therefore, no exhaustive list.

After having the attributes of the process related classes of the EoL process data model unified and after having found a proper grouping from them, the next paragraph will deal with the resource related class of the EoL process data model.

Resource part: the R of the PPR concept The process has a strong relationship to the resource. A process is a realization independent and logic description of an action or treatment for a product. With the resource, in contrast, the process is hardware specifically and physically realized. The literature review from Chapter 3 resulted in only one resource class - called *equipment*. Equipment ranges from workers, tools, vehicles up to facilities and factories.
The attributes correspond with the process attributes in *cost* and *revenue*, since only the physical realization of a process can generate costs and revenues. Therefore, the attributes are grouped into process related information and resource related information - see Figure 5.6.

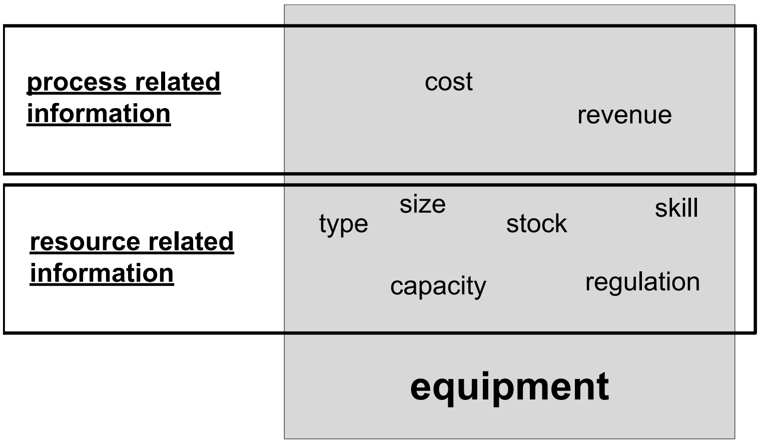

Figure 5.6: Grouping of the attributes of the class equipment (of the End-of-Life process data model)

Additionally, the attributes *skill* and *size* are added as resource related information to the class *equipment*. Since also humans are considered as equipment in this thesis, certain skills may be required to execute a certain process or to operate certain equipment. With size the dimensions of the equipment are understood, e.g. a vehicle for transporting a component must fit into the factory or must fit through the door.

A detailed description of the attributes of this class of the EoL process data model can be found in Annex F.2. The shown attributes in this paragraph are based on the literature review addressed with Chapter 3. It is, therefore, not an exhaustive list.

The relations between the process and resource related classes are also unified and generalized. The following figure shows this unified and generalized EoL process data model in a generic way (see Figure 5.7). That means that the process related classes are subsumed in one super class. The multiplicities of the association relations of the EoL process data model could be considered as 0..* but are not explicitly, quantitatively given in this figure.

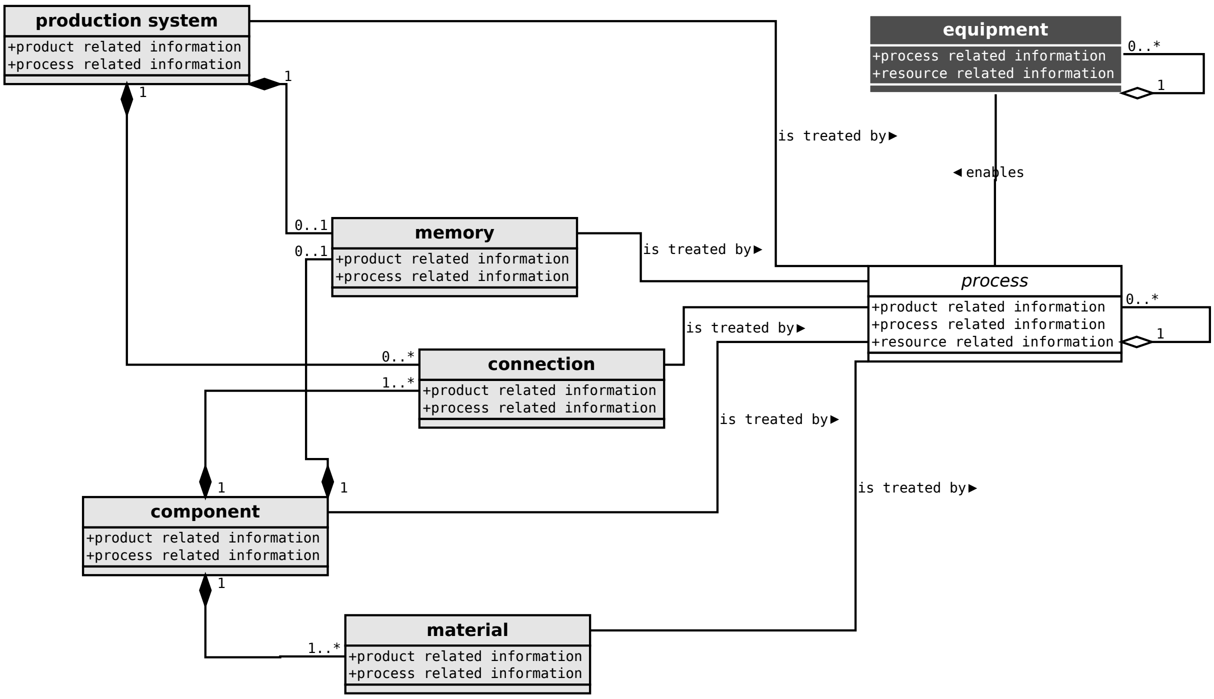

Figure 5.7: Generic End-of-Life process data model (unified and generalized)

The processes - subsumed in one super class - can treat the product related classes (production system, component, material, connection, and memory). A process can be divided into subprocesses. Each process is enabled or realized by equipment. Equipment, again, can be composed of other equipment.
It is obvious that not every process makes sense at a certain product related class, e.g. disassembly. For this information, Figure F.10 in Annex F.2 can be consulted, which represents the unified and generalized EoL process data model (without having it made generic).

When both data models are combined to the EoL data model it becomes clear, which information the data model finally needs to store. The attributes of the base EoL data model and the product related attributes of the EoL process data model are named as system related information - see Figure 5.8.

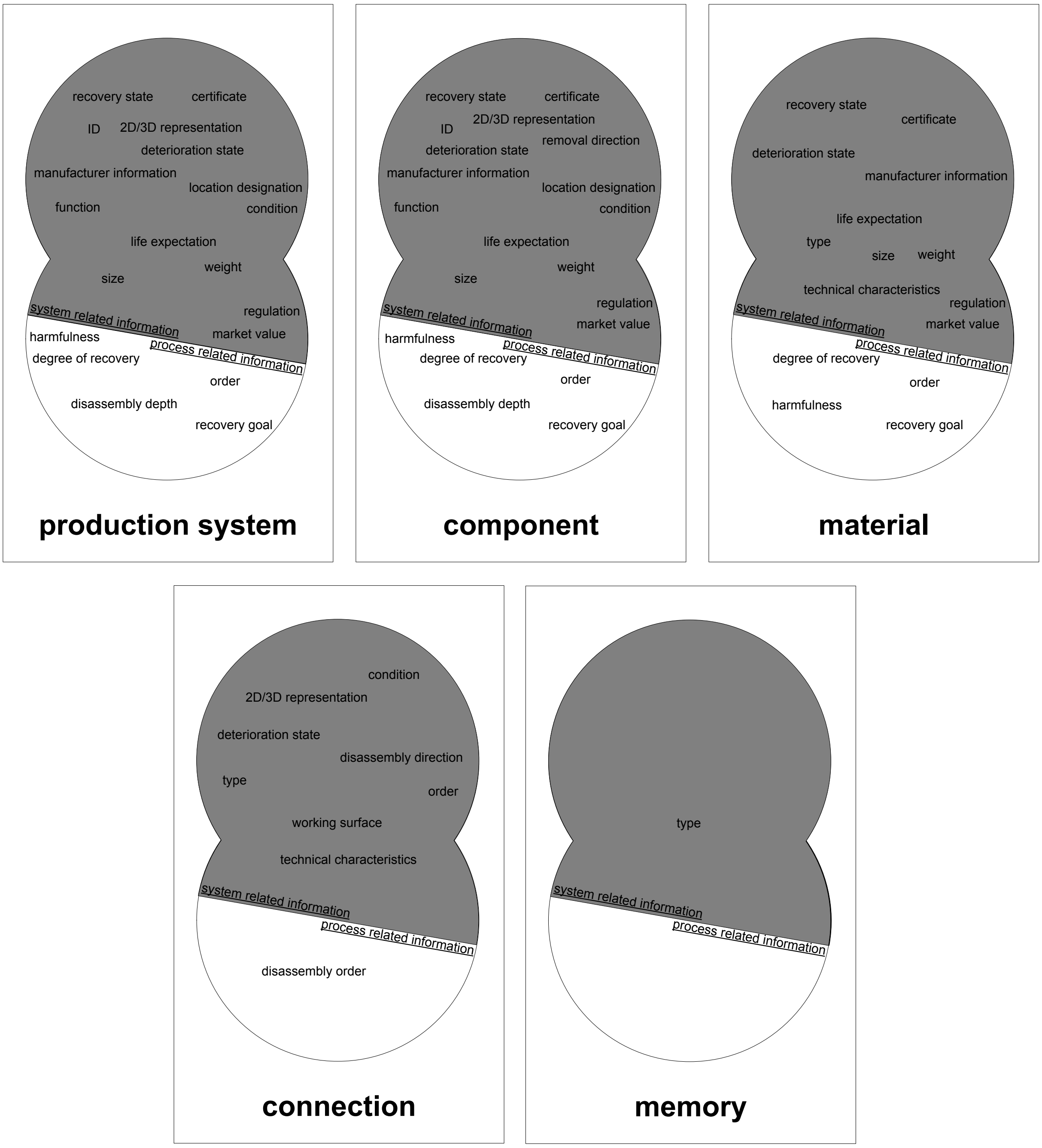

Figure 5.8: System related information of the End-of-Life data model

Given this definition, the unified and generalized EoL data model can be developed in a generic way (integrating the base EoL data model from Figure 5.2). Also here the multiplicities of the association relations of the EoL process data model could be considered as 0..* but are not explicitly, quantitatively given in this figure.

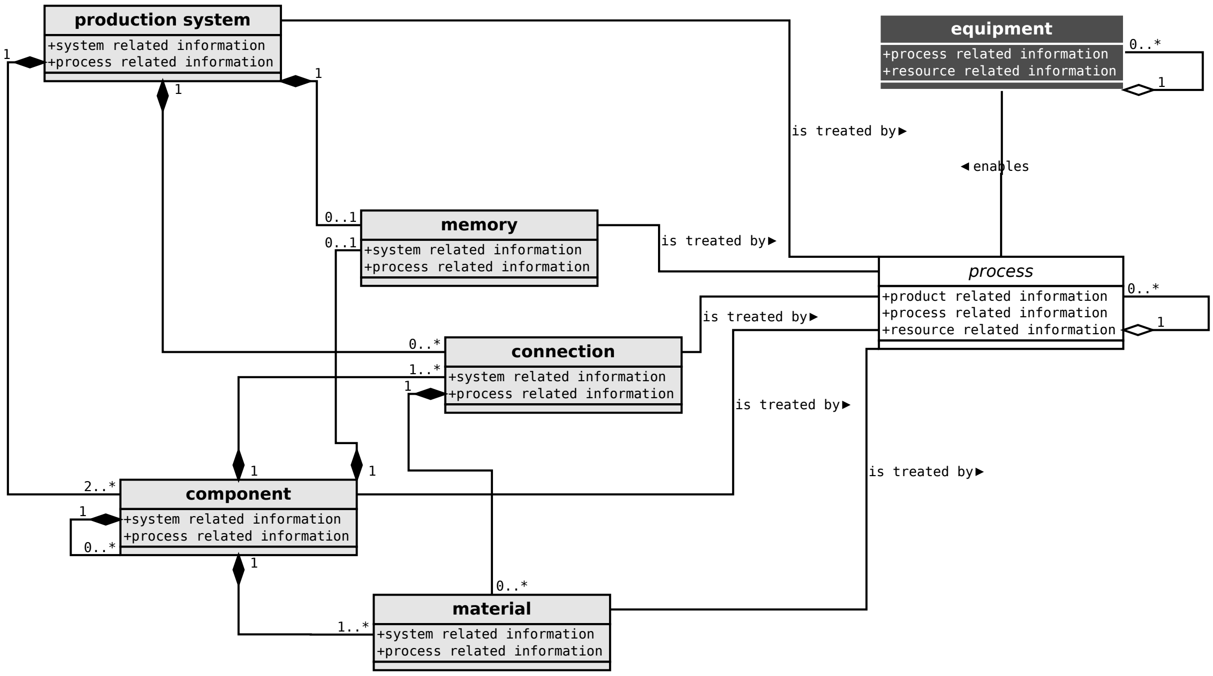

Figure 5.9: Generic End-of-Life data model (unified and generalized)

For detailed relationship information, Figure F.11 in Annex F.3 can be consulted, which represents the unified and generalized EoL data model (without having it made generic).

After having found a proper grouping for the attributes of the classes of the EoL process data model as well as unification for the processes and their associated relationships, the next subsection will deal with the comparability of the different EoL scenarios.

5.1.3 Feed back: End-of-Life data model

In Section 3.3.3 and 3.6 it has been determined that a characterization of the system and of the EoL strategy is necessary to compare the different EoL scenario with each other. Several indices or impact factors have been proposed and considered as useful namely: economic, ecological, societal, time, and quality.
Regarding the system characterization: Here, the recovery goal is characterized, which gets comparable with a future application of it. Is the production system, component, or material capable to meet the new requirements (of the future application)?
Regarding the characterization of EoL strategies: Here, the recovery process is characterized. Is the process as efficient and/or effective as needed? Does the process have an effect on the system characterization?
Figure 5.10 gives an overview of the two categories each with their five indices. The 'system.current.time' is not reasonable. This will be explained in the next section.

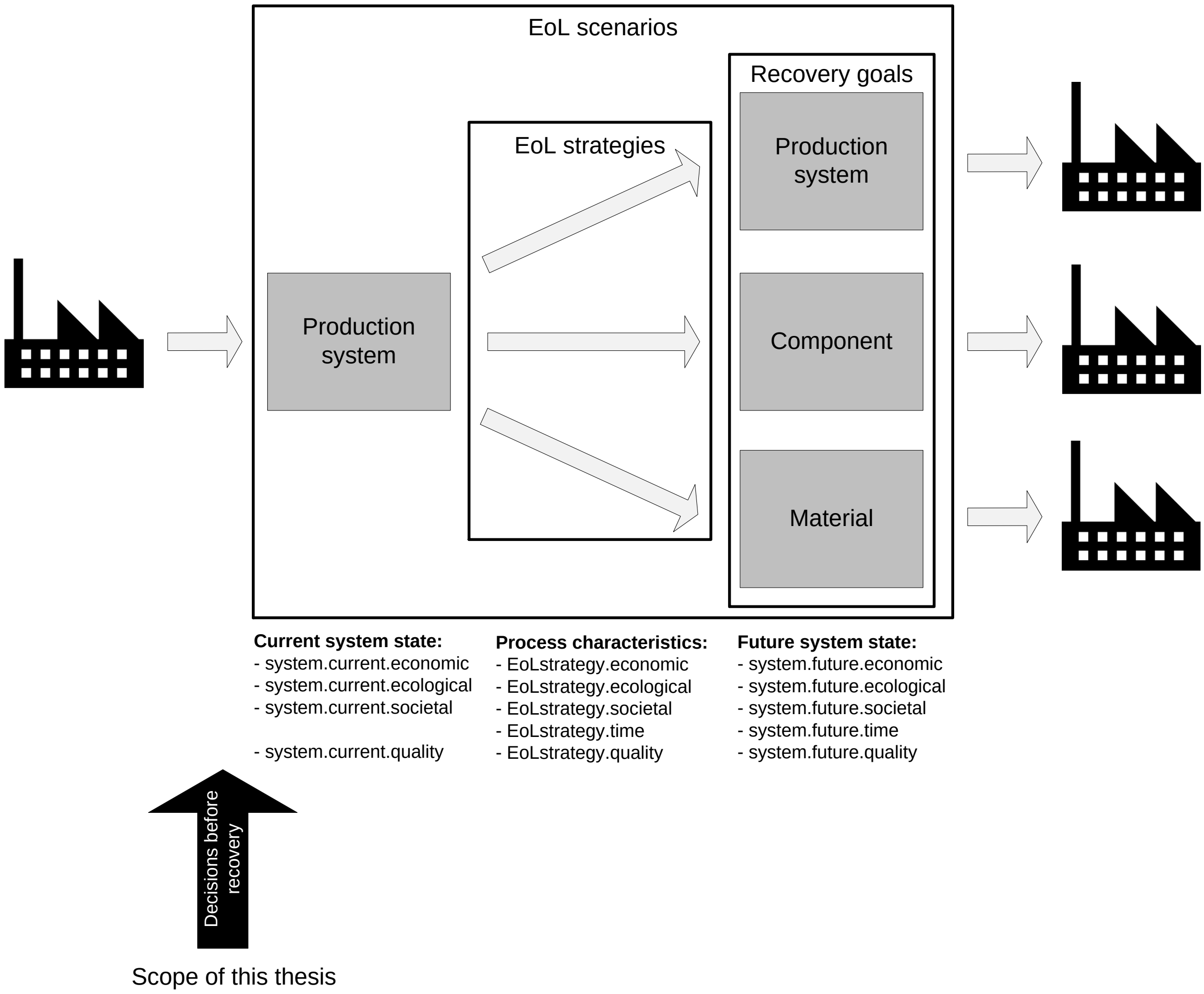

Figure 5.10: Characterization of an End-of-Life scenario using indices

To finally get to this characterization and, therefore, to the basis for a comparison, a procedure or method is required. It should systematically assist the decision maker to come to an informed decision about the best EoL treatment of scenario for a certain production system. This method is called Recovery Planning Method (RPM) and will be developed in the next section.

5.1.4 Conclusion

Even though, the base EoL data model is crucial and essential for the EoL phase, it is not state of the art having it, yet. Today, the production system needs to be inspected by sight from certain personnel, which compares the available documents or documentation of the production system with the real production system to derive the as-is state. This takes some time as shown by the case studies described in detail in Annex D.
However, the acquisition and provision of the data from the engineering and operation phase - namely the capturing of the as-is state of the production system - is not easy. Different issues come up, like 'There is no sensor available that can measure this data.', 'The company cannot assure that this is really the as-is state of the production system.', 'It is not profitable enough to measure this data.', 'This data has no value for the own company, it has only value for another company (service provider).', or 'This value is

too sensitive in terms of competitive advantage, it cannot be made available'. All those reasons that can be against such a base EoL data model are understandable from the viewpoint of the companies, but might disappear in the future. When measuring techniques evolve, when IT infrastructure and storage capabilities enhances, when business cases change, when external pressure from government, laws etc. is put onto companies, when the life cycle thinking is focused in a more connected world, in which data and services become more and more important rather than the physical assets themselves [Pla16a], this could become the time of the EoL phase of production systems. Ideas, trends, and developments emerging from the Industrie 4.0 paradigm are pointing in that direction.

This thesis assumes that such a base EoL data model is available. The thesis intends to highlight the possibilities that can emerge from this, once such a base EoL data model is available. However, in Section 3.5 it was discussed how a digital representation of a production system, which includes the base EoL data model (see Figure 5.11), could be possibly enabled and facilitated. A further analysis of this topic is not provided by this thesis.

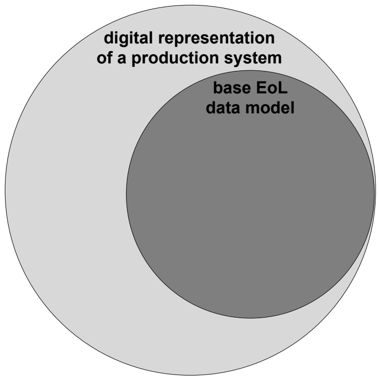

Figure 5.11: Relation of digital representation of a production system to base EoL data model

The base EoL data model is the basis for the RPM, which will be developed in the next section. It also supports or shortens the activities **inspection of physical asset** and **examination of documents** from the project preparation phase of the recovery process by making both labor intensive activities easier or even obsolete.

5.2 Recovery planning method

The purpose of the Recovery Planning Method (RPM) is to characterize EoL scenarios. By having them characterized, they become comparable and a decision maker can use this to make an informed decision about a suitable EoL treatment or scenario for a production system that is in accordance with certain goals, like regulations, laws, corporate strategy etc.
A brief introduction is given in Figure 5.12. These steps, which lead to this characterization, are explained in the following subsections (also in line with [SL17a] and [SL17b]).

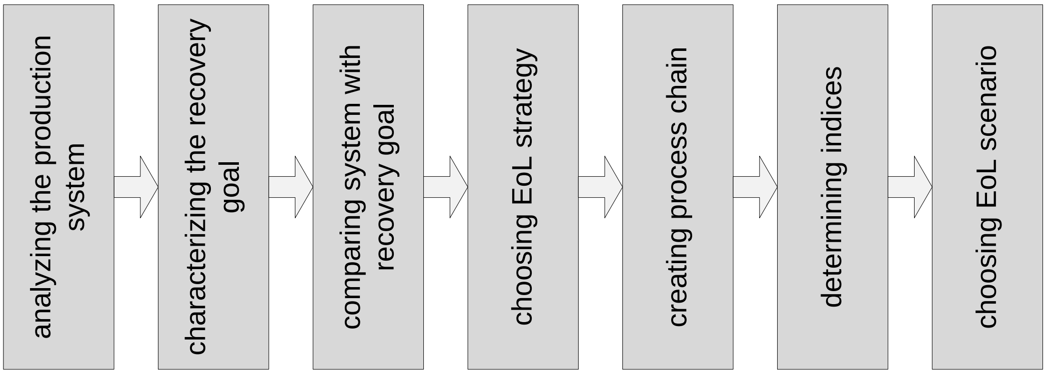

Figure 5.12: Recovery planning method briefly introduced

At it, only one physical asset is characterized at a time. If several physical assets need to be characterized, the RPM needs to be applied several times accordingly.

5.2.1 Step 1: analyzing production system data and identifying recovery goal

In the first step the production system data is analyzed. The decision maker analyzes the hierarchy or structure of the production system and identifies physical assets of interest/to be recovered in this hierarchy (see Figure 5.13) - namely the recovery goals. The components consist of materials, which are not explicitly shown in this figure. It answers the question: Which physical asset is recovered?

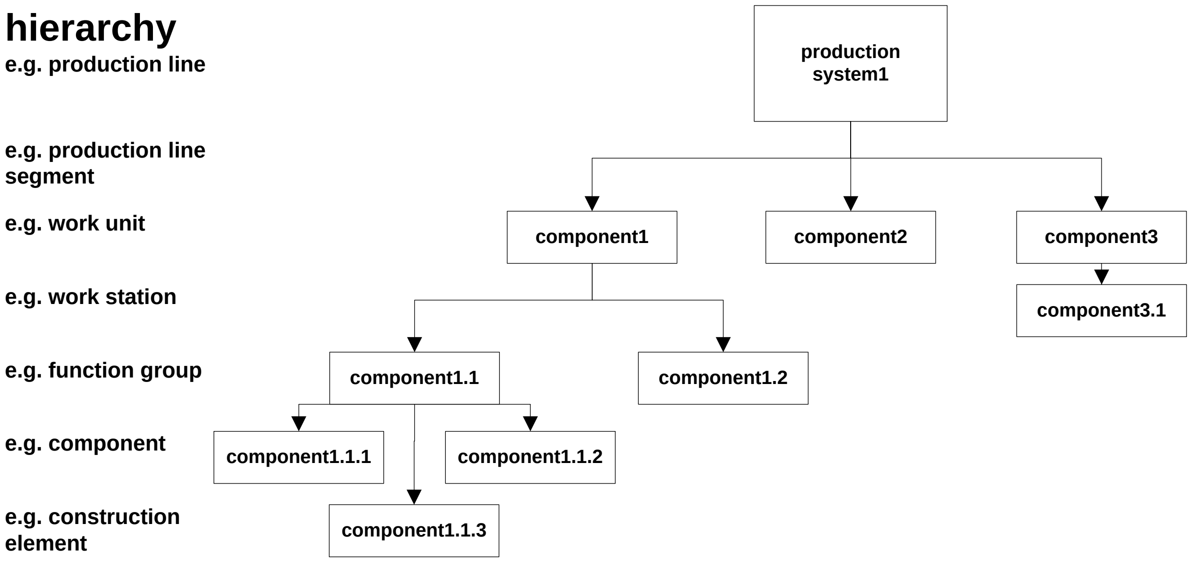

Figure 5.13: Step 1: analyzing production system data (hierarchy according to [LSH+17b])

From this hierarchy, individual assets, types or classes of assets, or assets on a certain hierarchy level can be chosen. An 'individual asset' when it is clear that only this asset needs to be recovered and e.g. the costs or time for the recovery process are needed. A 'type of assets' when, e.g. all logistics assets should be recovered. A 'class of assets' when e.g. all robots should be recovered or all the copper. 'Assets on a certain hierarchy level' when, e.g. all function group assets should be recovered.

For a holistic consideration all possible, recoverable physical assets would need to be considered.

The production system data is provided by the base EoL data model.

5.2.2 Step 2: characterizing recovery goal

In the second step the identified and chosen recovery goals need to be characterized, i.e. the future application of the physical asset. As shown in Figure 5.10 in Subsection 5.1.3, the author proposes the five indices to characterize the future application of or the requirements on the physical asset in its new setting, e.g. a reused robot in another work station - see Figure 5.14. It answers the question: How does the future application of the physical asset to be recovered look like?

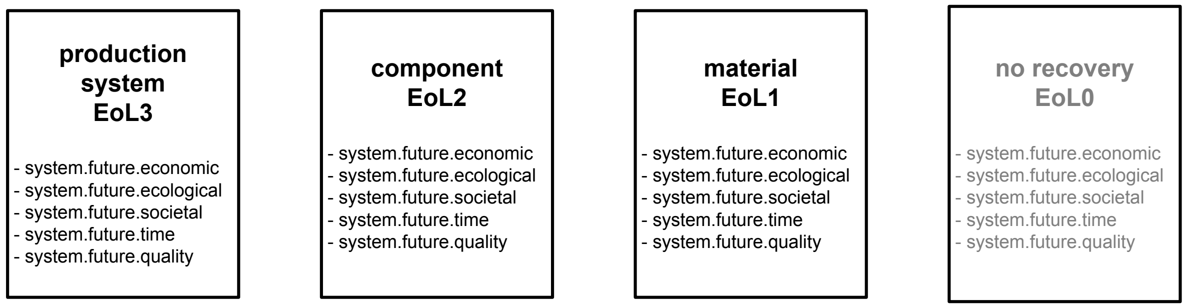

Figure 5.14: Step 2: characterizing recovery goal

With 'system.future.economic' the allowed costs for having the physical asset recovered are considered. But also the allowed costs for the physical asset itself are taking into account. With 'system.future.ecological' the allowed environmental impact for having the physical asset recovered is considered as well as ecological footprint of it, e.g. energy efficiency. With 'system.future.societal' the allowed societal impact for having the physical asset recovered is considered as well as working conditions of the physical asset when it is operating. With 'system.future.time' the allowed time for having the physical asset recovered is considered. With 'system.future.quality' the required quality of the physical asset is considered, e.g. the repeat accuracy of a robot. According to [DIN05] quality is understood as the "degree to which a set of inherent characteristics fulfills requirements". In this thesis it is applied to the technical characteristics of a physical asset. The recovery process can have an effect on the quality of the physical asset.

Those requirements on the future application of the physical asset need to be provided by the decision maker.
In the end, this information is somehow stored as *recovery goal* attributes at the classes *production system*, *component*, or *material* into the EoL data model.

5.2.3 Step 3: comparing system with recovery goal

In the third step the physical asset is compared to its intended future application with its requirements. At it, the same indices from step 2 are applied to enable a comparison.

The comparison is divided into 'technical assessment' and 'value assessment' - see Figure 5.15). It answers the question: Is the actual physical asset capable to fulfill the future application with its requirements?

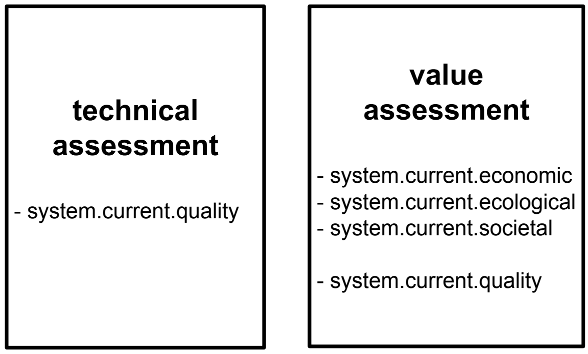

Figure 5.15: Step 3: comparing system with recovery goal

The 'value assessment' considers the current system state of the physical asset. With 'system.current.economic' the market value of the physical asset is considered. With 'system.current.ecological' the ecological footprint of the actual physical asset is considered, e.g. energy efficiency. With 'system.current.societal' the working conditions of the physical asset are considered when it is operating. The 'system.current.time' is not existing because this is a process variable only and the 'value assessment' only evaluates the physical asset itself and not the recovery process necessary to get the physical asset recovered. With 'system.current.quality' the actual quality of the physical asset is considered, e.g. the repeat accuracy of a robot.
The 'value assessment' is later used again in step 6.

All information for the current system state needs to be provided by the system related information of the EoL data model (see Figure 5.16).

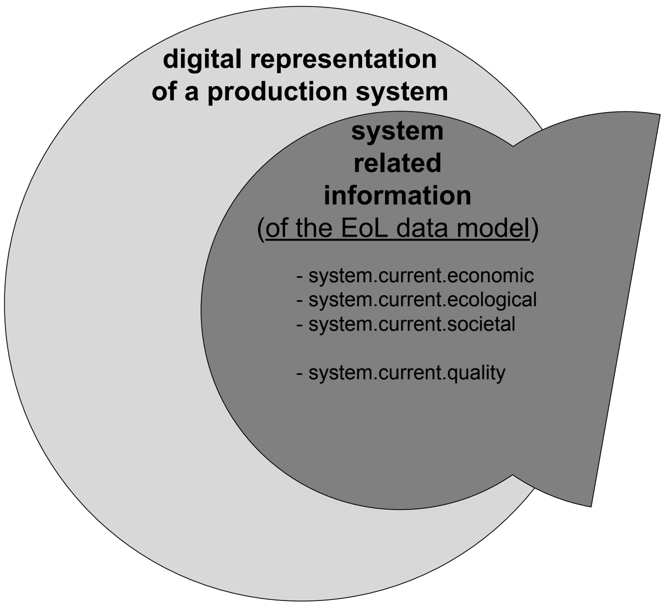

Figure 5.16: Relation of the current system state to the digital representation of a production system and the EoL data model

This figure uses the symbols from Figure 5.8. It visualizes the necessary information for the classes *production system*, *component*, *material*, *connection*, and *memory*. It shows that more information is needed then provided by the digital representation of a production system.

How those indices correlate in particular with the system related information of the EoL data model (including type, technical characteristics, instance information, and measured data of the base EoL data model and the product related information of the EoL process data model), presented in Figure 5.8, will not be analyzed in this thesis. Because they are use case dependent. For example: For determining the 'ecological footprint' of a welding or painting robot cell of a production system different and more information is necessary than for a handling robot cell.

The 'technical assessment' evaluates with which recovery process or EoL strategy the physical asset can achieve the fulfillment of the requirements on the future application. Figure 5.17 shows a possible decision tree to find an EoL strategy. This is equivalent to the decision tree from Subsection 3.2.2.

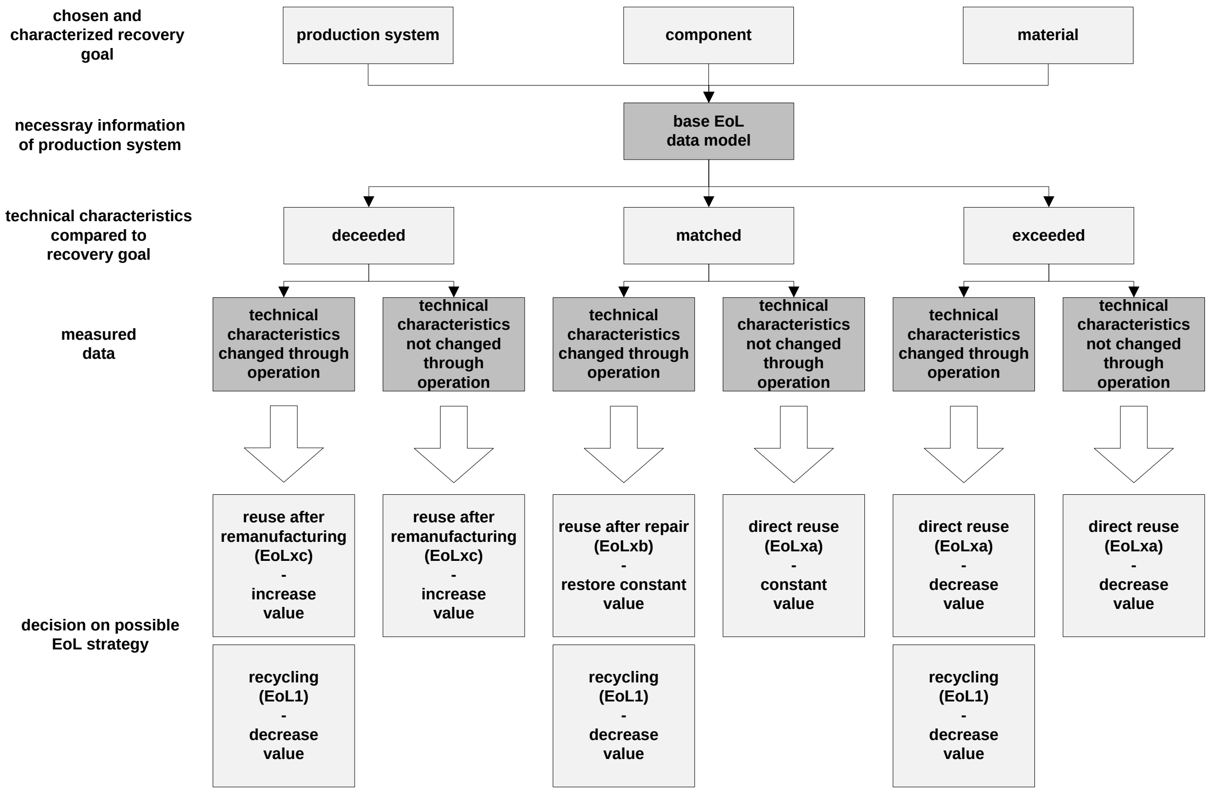

Figure 5.17: Decision tree for EoL strategies

The decision is basically made by using the technical characteristics of the physical asset. At first, it is compared whether the technical characteristics in its current state of the physical asset deceed, match, or exceed the technical characteristics of the future application. Secondly, it is compared whether the technical characteristics in its current state of the physical asset have changed though operation. For it, the initial technical characteristics (before operation) are compared with the deterioration or change of the technical characteristics (after operation) captured by the measured data.

Depending on whether the physical asset deceeds, matches, or exceeds its future application certain EoL strategies are necessary that increase, restore, or decrease the current value of the physical asset correspondingly. In three out of six cases two EoL strategies are possible. According to the KrWG law [Fed16b] recycling is always considered as decreasing the value of a physical asset that says reusing before recycling before disposing. For example, when a component should be recovered that matches the future application, but due to operation the component is partially worn out. Then, it needs to be repaired first, before it can start its second life in the future application. EoL scenario EoL2b is applied. But when this component is too worn out (according to the measured data), the only option is to recycle it. EoL scenario EoL1 is applied. In this case, disposing would also be an option (EoL0), but is in general not considered in this thesis.
The combinations shown in Figure 5.17 are not exhaustive. Cases for this example like 'just a little bit worn out, no repair needed, directly reused, EoL2a' are not considered.

The necessary information is provided by the base EoL data model.

In addition, the author wants to point out that also non-technical reasons can lead to a certain EoL strategy. Some of those were collected during the expert interviews in the case studies and are listed in the following, but are not further considered:

- Know-how, which was put into the manufacturing processes of the production system, shall be protected.
  This causes an elimination of EoL3 and may also cause an elimination of EoL2.
- Avoid direct competition for the same product.
  This can cause an elimination of EoL3 and may also cause an elimination of EoL2.
- Avoid a saturation of the market for a specific production system or components of it.
  This causes an elimination of EoL3 or even EoL2.
- The production system is not allowed being relocated to a specific country, e.g. when it is producing cigarettes or it can be somehow used to build weapons.
  This causes an elimination of EoL3 and may also cause an elimination of EoL2.

5.2.4 Step 4: choosing EoL strategy

In the fourth step all EoL strategies that were assessed as possible in step 3, are presented and one EoL strategy of them is chosen by the decision maker - see Figure 5.18. It answers the question: By which EoL strategy is the actual physical asset capable to fulfill the future application with its requirements?

Figure 5.18: Step 4: choosing EoL strategy

The definitions for the four EoL strategies are given in Section 3.2. The fifth one 'disposal/new (EoL0)' is only presented for the sake of completeness. It is not considered in this thesis but should not be neglected, either.
As mentioned, the choice needs to be made by the decision maker.

5.2.5 Step 5: creating process chain and characterizing processes

In step 5 the EoL strategy is composed to a process chain by the single processes on the one hand. On the other hand each process is characterized according to the indices used in step 2 and 3. The list of possible processes shown in Figure 5.19 is in line with the processes identified by the literature review. It answers the question: What is the process chain for the EoL strategy?

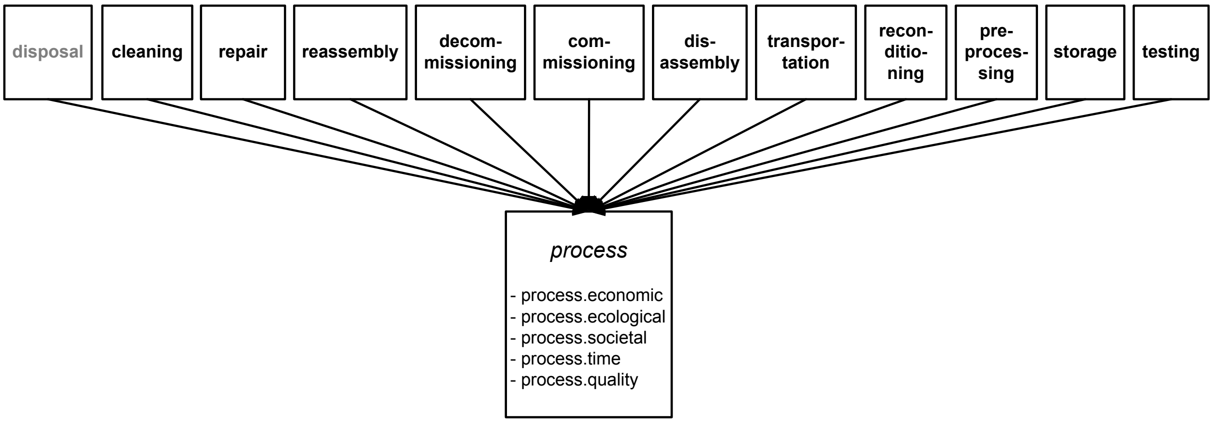

Figure 5.19: Step 5: creating process chain and characterizing processes

This characterization of the processes is necessary since the recovery process itself has an effect on the recovery of a physical asset. For a holistic consideration it should be included.
For comparability again the five proposed indices are applied. With 'process.economic' the estimated costs for a single process are considered, e.g. expected costs for transportation. With 'process.ecological' the estimated environmental impact of a single process is considered, e.g. expected implications of cleaning a production system. With 'process.societal' the estimated working conditions when executing a single process are considered, e.g. toxic gases during preprocessing. With 'process.time' the estimated time a single process takes is considered, e.g. time for disassembling a component. With 'process.quality' the estimated impact of a single process on the quality of the physical asset, e.g. possible damage through transportation.

This process characterization needs to be done by the decision maker.
In the end, this information is stored as attributes of the *process* classes as well as of the *equipment* class into the EoL data model (see Figure 5.9). As mentioned before, only when executing a process with equipment costs are generated.
But how those indices correlate in particular with the product, process, and resource related information of the *process* classes and with the process and resource related information of the *equipment* class of the EoL data model will not be analyzed in this thesis. These attributes and their categorization are highly use case dependent because

of the diversity and range of the processes considered here in this thesis and, hence, also of the equipment.

How each EoL strategy is composed by the processes is use case dependent. Based on the author's gained knowledge through the case studies (from Section 4.2) and on publication [BE95] and [LSsB06], possible process chains could be these proposed ones in Figure 5.20.

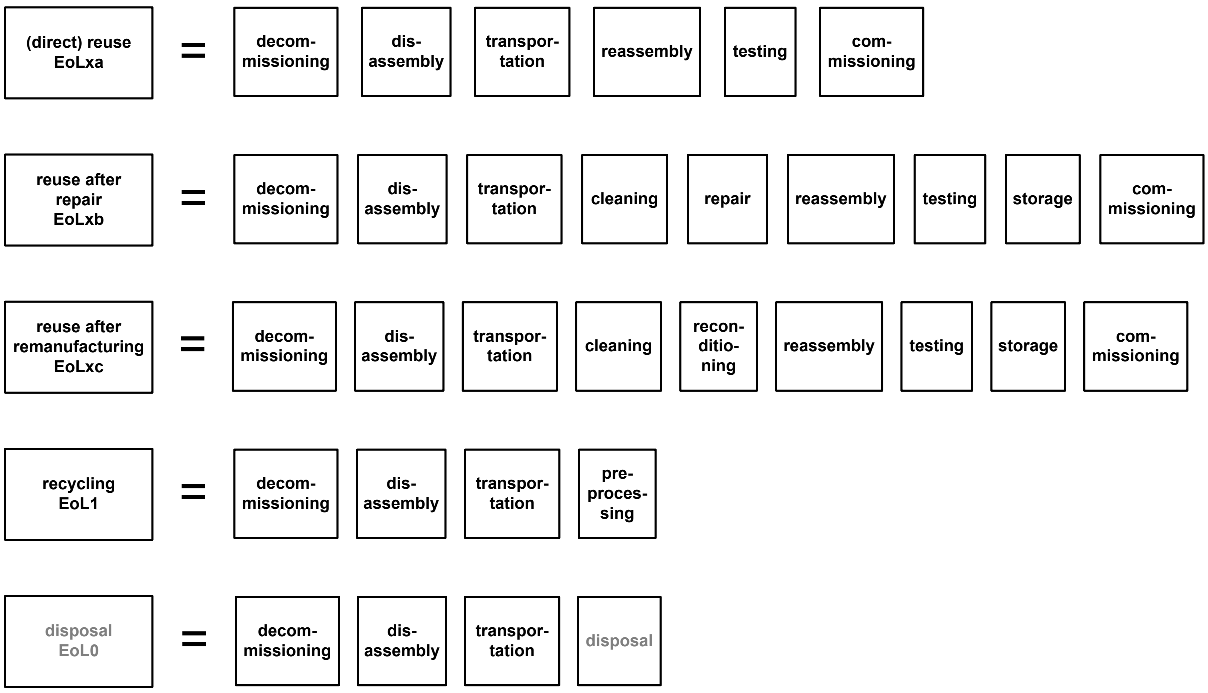

Figure 5.20: EoL strategies with possible process chains

Creating the process chain is also done by the decision maker.
In the end, this information is stored as *order* attributes at the classes *production system*, *component*, or *material* into the EoL data model on the one hand. And on the other hand as *process* classes. As guidance it can help to use the generic EoL data model from Figure 5.9.

It should be considered that a process is physically realized by resources, i.e. each process needs to have a corresponding resource assigned to it.

5.2.6 Step 6: determining and weighting indices

In step 6 all characterizations from the previous steps using the five indices (see Figure 5.21) are summed up. These indices are called performance indices in this thesis. 'Performance' should indicate how well an EoL scenario is doing regarding economics, ecology etc. It answers the question: How is the physical asset with its EoL strategy characterized? How is the EoL scenario characterized?

Figure 5.21: Step 6: determining indices

According to the process chain of EoL strategy from step 5 each index is summed up, which results in the characterization of the EoL strategy. This EoL strategy characterization is summed up with the characterization of the current state of the physical asset. This result is, then, compared to the future application or future state of the physical asset - see Figure 5.22.

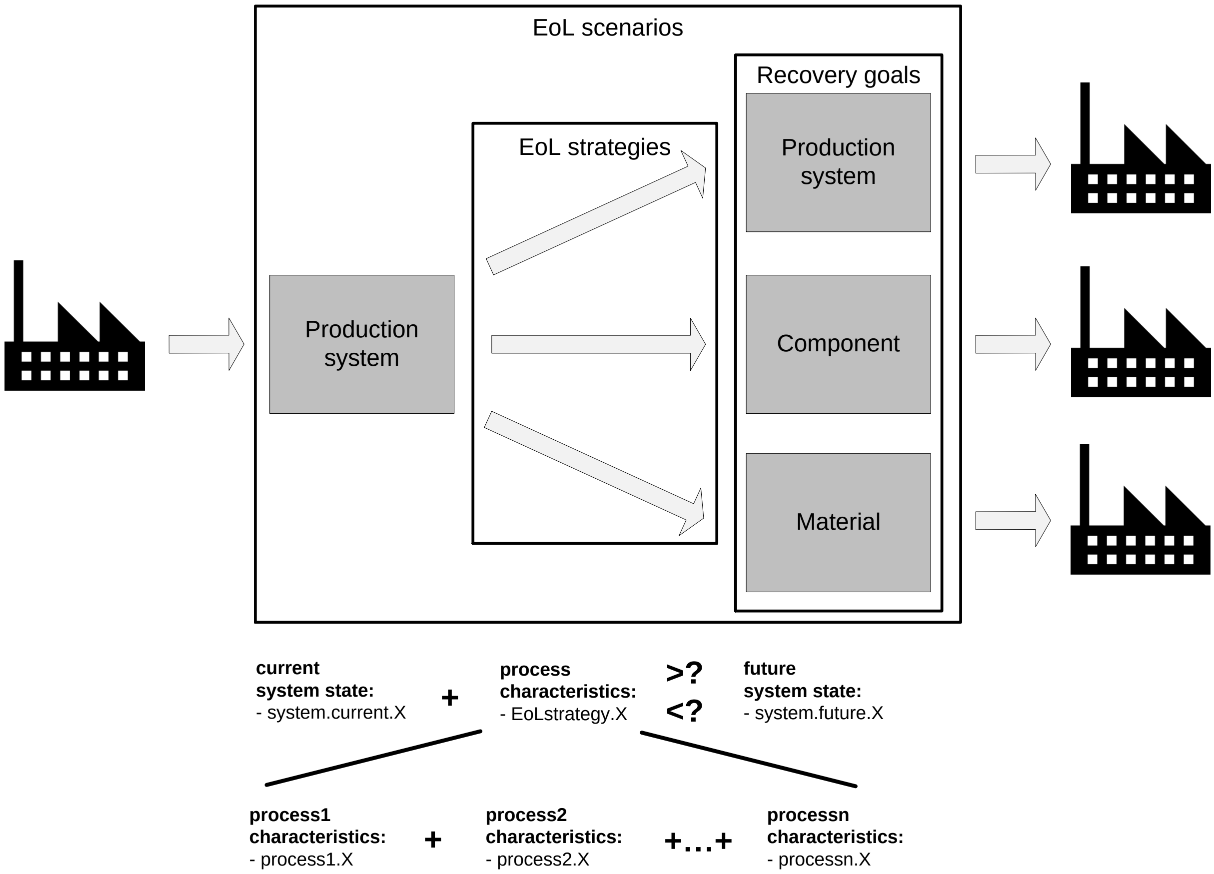

Figure 5.22: Calculation of performance indices

Depending on the corporate strategy, regulations etc. each index could also be weighted, in case that one index is more important than others.

5.2.7 Step 7: presenting characterized End-of-Life scenarios and choosing one

In the seventh and last step of the RPM the characterized EoL scenario is presented to the decision maker - see Figure 5.23. In case that several EoL scenarios were characterized and, therefore, the RPM was applied several times, all EoL scenarios can be presented at a glance, e.g. as a spider chart. Methods for presenting the characterized EoL scenarios can be applied, e.g. portfolio [Lin09].

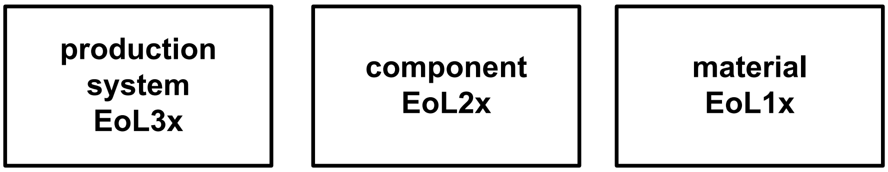

Figure 5.23: Step 7: presenting characterized End-of-Life scenarios

The decision maker can now make an informed decision about a suitable treatment of a production system at the end of its life. But also KPI based strategic planning methods could be used further based on the characterization and these performance indices, e.g. balanced scorecard, benchmarking, decision tables, (weighted) points rating, weighting, cost-benefit analysis, (pairwise) comparison, prognosis, scenario technique, advantage-disadvantage-comparison, preference matrix [Lin09], or sensitivity analysis [Lin09] [Ves02].
How the performances indices or KPIs are further assessed by the decision maker, this is essential for the overall result [Hub14] of the recovery process.

In the end, this choice is stored as *recovery goal* attributes at the classes *production system*, *component*, or *material* into the EoL data model.

5.2.8 Summarizing steps 1-7

Figure 5.24 shows step 1-7 at a glance. It summarizes the previously explained steps in that way that the procedure becomes clear.

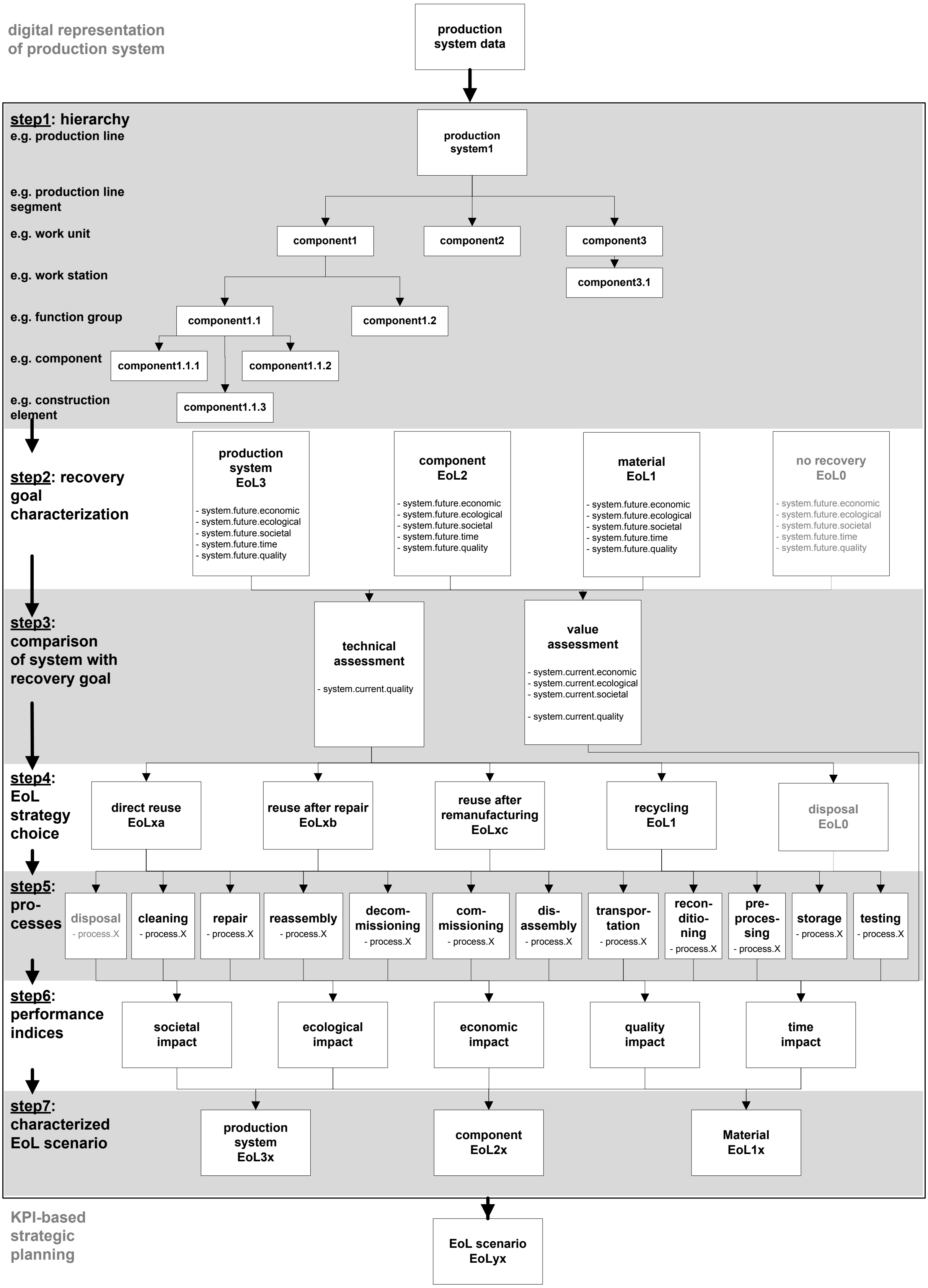

Figure 5.24: Steps of the recovery planning method

The RPM is data intensive and when it is applied for the first time, time needs to be spent in acquiring and aggregating all the information and data, especially for step 2, step 3, and step 5. But once done, e.g. the characterization of a certain process could be stored in a library and used again as a template. In case, this template has a high probability to be used again, a library concept can be applied. This is shown in Figure G.5 in Annex G. Thereby, the RPM is extensible and adaptable for further applications. According to Figure 5.24, some libraries are dependent on each other, e.g. performance indices are used for the recovery goal characterization. Those interrelations are addressed by Annex G, besides the workflow for using and creating the corresponding library entries.

5.2.9 Conclusion

The RPM supports the activities **estimation of process costs, assessment of value of physical asset** and **collection of requirements** of the recovery process of production systems by design - but on a very general basis due to the inductive scientific approach of this thesis. It intends to provide guidance for decision makers but needs to be adapted to the specific use case.
The RPM is, therefore, not applicable 'out of the box'. It is rather a framework that needs to be filled with specific information to make it work. The RPM only provides types of information - not the specific information needed to make a decision on a specific production system. Production systems are per definition individual solutions and it is barely possible to find a specific set of information, which answers the question on a proper EoL scenario and are general enough to fit the different manifestations of production systems.
Based on information (namely base EoL data model and EoL process data model) valid EoL scenarios need to be created. This comes with finding valid recovery goals and EoL strategies with a valid process chain. A challenge, out of the scope of this thesis, is how to determine the market value of the physical asset. Through the case studies the author knows when a physical asset is sold within one company, the value is assessed by the time it was operating regarding write-off. However, if the physical asset is still functional it has a value. When the physical asset is sold company crossing, this functional value needs to be assessed monetarily somehow. Case study CS **V** 'marketplace provider' addresses this.
Regarding the characterization of EoL scenarios, a challenge is to find a common set of information (for the physical asset and for the processes) on which the performance indices could be calculated. Also to make the indices for the different processes comparable is a challenge in particular. For example [SSZ+14] has proposed and applied different normalization methods, like expert assessment or benchmark normalization, to make the metrics for the different indices comparable among each other. Also the already in Chapter 3 mentioned Design-for-X guidelines can be consulted, which usually provide metrics for assessing the X (e.g. recyclability, remanufacturability, transportability etc.). However, this is out of the scope of this thesis. It would be anyhow conceivable when system/component manufacturers and their customers would develop or agree on a standard for describing the systems and components regarding the four indices (system.current.economic, -ecological, -societal, and -quality), it would be easier for the customers to create the system.future.X of them. Comparable to the energy label (ranging from D to A+++) what indicates the energy consumption of electronic devices.

Also out of the scope of this thesis is finding appropriate metrics or characteristics to calculate the different performance indices. As partially mentioned in Chapter 3, e.g. for ecological impact the DIN EN ISO 14031 can be consulted [DIN13] or other publications with reference to LCA (life cycle assessment) [HJA05b], for the societal impact the sLCA (social life cycle assessment) [Lif09], for the economic impact the VDMA 34160 [VDM06], for time impact REFA [Bet97], or for quality [MT02], [Gar87], [Fed16a], [COR16], or [RSI00] could be consulted.

In general, the RPM is a method that consumes data from the EoL data model but also generates data into this EoL data model - as indicated in the different steps.
As mentioned before, the RPM shall support the project preparation phase of the recovery process (see Figure 4.6), so that the project planning phase can start with making a decision on the EoL strategy or EoL scenario and start to plan it in detail.
The RPM with its steps is integrated in an activity workflow for the project preparation phase (see Figure 5.25).

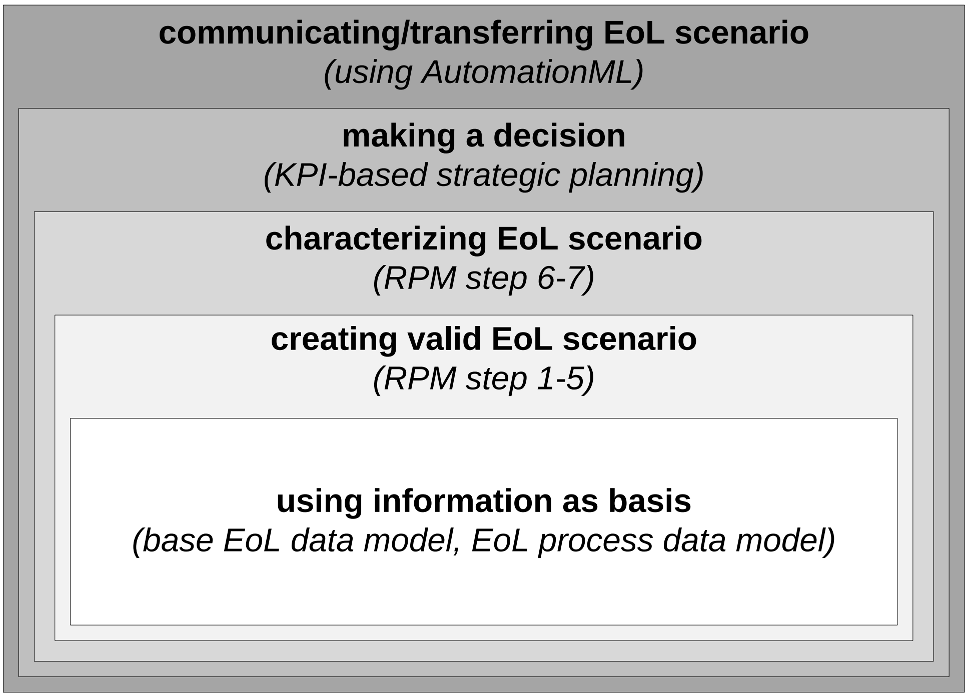

Figure 5.25: Activity workflow for the project preparation phase

The project preparation phase is the first phase of the recovery process. When looking at the engineering process that is described in Subsection 2.2.1 it also has different phases, which are more specific, but similar to the recovery process. When comparing the recovery process with the engineering process, it can be stated that the project preparation phase corresponds to the analysis phase of the engineering phase. Here, the requirements on the production system are collected and a first possible draft solution for the to be planned production system is developed [LSR+13]. This first possible draft solution corresponds with the characterized EoL scenario, i.e. the result of the RPM. This first draft solution for the to be recovered production system, of course, needs to be communicated and transferred, since other experts need to detail this draft solution later on. So, that in the end the production system can be recovered.

Having this in mind, it becomes essential to have the results of the RPM machine processable and exchangeable stored. It should be also stored in a neutral way as there are a lot of experts and stakeholders involved in the recovery process or EoL phase of production systems (as described in Chapter 4).

AutomationML has exactly addressed this issue on making the production system data consistently exchangeable in the engineering process, from the first draft until the very detailed specification of the production system [Dra10]. However, the engineering process is characterized by a heterogeneous software tool landscape, i.e. the experts using highly specialized tools to design or engineer one or more aspects of the production system. Usually they have incompatible interfaces, so that the data from tool A cannot be imported into tool B. To still enable a data exchange between tool A and tool B the results from tool A are usually printed out. The other experts needs, then, to insert this data manually, which is time consuming, error-prone and can result in an information loss over time as shown by [BKKB15]. AutomationML wants to get rid of this paper interface.
But according to the very detailed descriptions of the recovery processes of the case studies (see Annex D) the software landscape does not appear such heterogeneous and highly specialized as the one of the engineering phase. Mainly MS office products are used. The author thinks that it is likely that this is because of the already mentioned absence of the digital representation of the production system and, therefore, the absence of the base EoL data model. There is no use case for software tool vendors to develop and provide tools highly specialized for the EoL phase on the one hand. On the other hand, potential software tool users cannot put pressure on the software tool vendors (by giving them a use case) to develop those, since such tools could not generate any benefits in saving time and money, because the required data is not available in a machine readable manner. The author is convinced that when the digital representation of the production system is once available the EoL phase with its stakeholders will highly benefit from this.

Due to the reasons mentioned in Subsection 3.5.3, it is proposed to use AutomationML in the *future* heterogeneous software tool landscape of the recovery process and EoL phase of production systems. The next chapter will, therefore, address the mapping of the EoL data model onto AutomationML.

6 Realization of the data representation of the recovery planning method

This chapter addresses the systematically and stepwise development of the mapping of the EoL data model onto the data exchange format AutomationML and it elicits how the Recovery Planning Method (RPM) is applied to fill the data in the EoL data model. At first, the chapter starts with an introduction of AutomationML itself. Subsection 3.5.3 had mentioned the reasons for this choice.

6.1 Introduction of AutomationML

This section briefly introduces the data exchange format AutomationML with its characteristics and capabilities, even though AutomationML was already addressed by previous chapters.
AutomationML (AML) is an open, neutral, free, XML based, scalable, and extensible data exchange format that is standardized in IEC 62714 [IEC14] in TC 65/SC 65E/WG 9 [Int17b]. It enables a consistent and lossless data exchange of the production system data within the entire engineering process (see Subsection 2.2.1). This means AutomationML is capable to store data from rough descriptions until very detailed descriptions of a production system. It can be used to connect engineering software tools (discipline and company crossing) ranging from mechanical construction tools, over electrical construction tools, PLC programming tools to virtual commissioning tools.
The basic architecture of AutomationML consists of three different data formats, which in combination can enable that mentioned consistent and lossless data exchange (see Figure 6.1). It was not the intention to develop an own format. Instead already existing and established formats should be used.

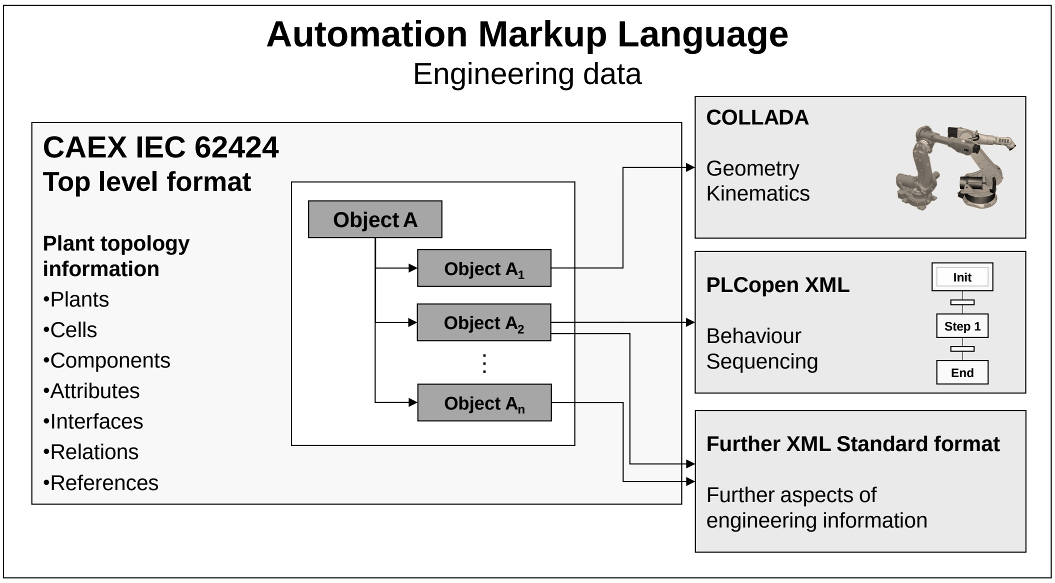

Figure 6.1: Basic architecture of AutomationML [IEC14]

As seen in Figure 6.1, the AML data format or the AML model can store the production system structure what is expressed by a hierarchy of AML objects with their relations among each other. At each AML object this information is stored that is relevant for the object and describes it. This description can range from rough to detailed. In case that this object description shall contain a 2D/3D representation, a reference to an external COLLADA document [ISO12] is created (*.dae). In case that this object description shall also contain a logic behavior, another reference but to an external PLCopen XML document [PLC] is created (*.xml). PLCopen XML [PLC] itself is currently under development as IEC 61131-10 in TC 65/SC 65B/WG 7 [Int17a].
Each AML object can integrate descriptions with different semantics. Thereby, the different aspects of a production system related to its disciplines are expressible. [Dra10] [LS17a]
To make an AML object identifiable within the entire engineering process a consistent object identification concept is required. [IEC14] defines that objects are identified by their ID and classes by their name. This ID shall be unique. It is proposed to use UUID (universally unique identifier) that could be realized as a GUID (globally unique identifier), e.g. 788eb291-f103-4fdc-aba0-4893b599f556. By using a GUID it is likely that all AML objects in an engineering process have a unique ID and are, therefore, unambiguously identifiable. The object name has, then, only display purposes.
All in all, AML is technically capable to store the virtual representation of the production system that is the result of the engineering phase.

Current developments [Aut16], [DIN16b], [HS14] have enabled AML to be exchanged during operation time - by mapping AML to the OPC UA information model. That means that AML is technically also capable to store the digital representation of the production system that is the result of the operation phase - see Figure 6.2.
AML objects were used to create the physical components and these current developments can now enable these components to communicate their current (possibly changed) description in AML via OPC UA to the (superordinated) digital representation of the production system. Since the virtual representation has become the digital representation, the AML objects can be used beyond the engineering phase. With the digital representation they are now used in the operation phase. And they have kept their ID, i.e. it is likely that these IDs are unique in the operation phase and, therefore, the components are unambiguously identifiable. Hence, an up to date digital representation or as-is-state of the production system is possible. [SL17b] This is the assumption that is made in this thesis.

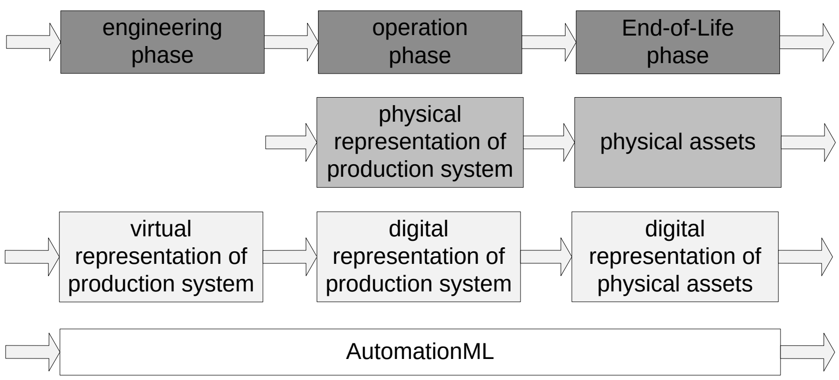

Figure 6.2: AutomationML in the production system life cycle

Based on this assumption and, therefore, with the input, the RPM (developed in Chapter 5) is applicable. Based on this assumption and based on the RPM, it is possible to divide the physical representation of the production system (coming from the operation phase) into single physical assets. Besides the “physical division” the division of the digital representation of the production system is made. When AML is capable to store the production system’s digital representation, it is also capable to store parts of it, i.e. the digital representation of the physical assets. This is also shown in Figure 6.2.
That means that AML is technically capable to store the digital representation of the physical assets that is the result of the EoL phase.
When using the IDs from the operation phase and the digital representation, a physical asset can be unambiguously identified and treated in accordance with its intended EoL scenario. And when the physical asset intends to start a “new life” the ID shall be kept as well. In doing so, this ID becomes a life cycle ID.
Hence, physical assets (with the description in AML) can be easily used again in other life cycles.

Not only the life cycle phase crossing usage of AML makes sense. As mentioned in Subsection 5.2.9, also the usage within the EoL phase makes sense, since the recovery process of the EoL phase is comparable to the engineering process of the engineering phase. Therefore, the next section will address the stepwise development of the mapping of the EoL data model to AML and show how RPM is applied to fill the data in the EoL data model.

6.2 Mapping onto AutomationML

In AML the semantics and syntax are separated. While the syntax is completely set through the AML standard IEC 62714 (especially in IEC 62714-1 [IEC14]), the semantics is flexible by design. Even though, IEC 62714-1 [IEC14] specifies a basic semantics and IEC 62714-2 [IEC15a] a more specific semantics, there is still no semantics defined that would fit the EoL specific needs. Therefore, an EoL specific semantics is developed in this thesis.

The semantics is stored in the CAEX part (*.aml) of AML’s distributed multi-document structure that provides four concepts: RoleClassLibraries, InterfaceClassLibraries, SystemUnitClassLibraries, and InstanceHierarchy. The first three are libraries. With role classes (RC) the neutral meaning of an object (namely InternalElement (IE)) is expressed. With interface classes (IFC) the neutral meaning of an interface (namely ExternalInterface) is expressed. With system unit classes (SUC) the specific meaning or type of an object is expressed. SUC can be used as templates that are instantiated in the InstanceHierarchy (IH). The IH represents the real project, the virtual or digital representation of the production system. [LS17a] All the four concepts are again visualized in Figure 6.3.
With the new edition of CAEX (IEC 62424, Edition 2.0 [DIN16a]) a fifth concept will be available: attribute libraries. Like role class libraries it gets possible to store attribute semantics in a neutral manner. AML will integrate it into its upcoming second editions of the standards and it is, therefore, not considered in this thesis.

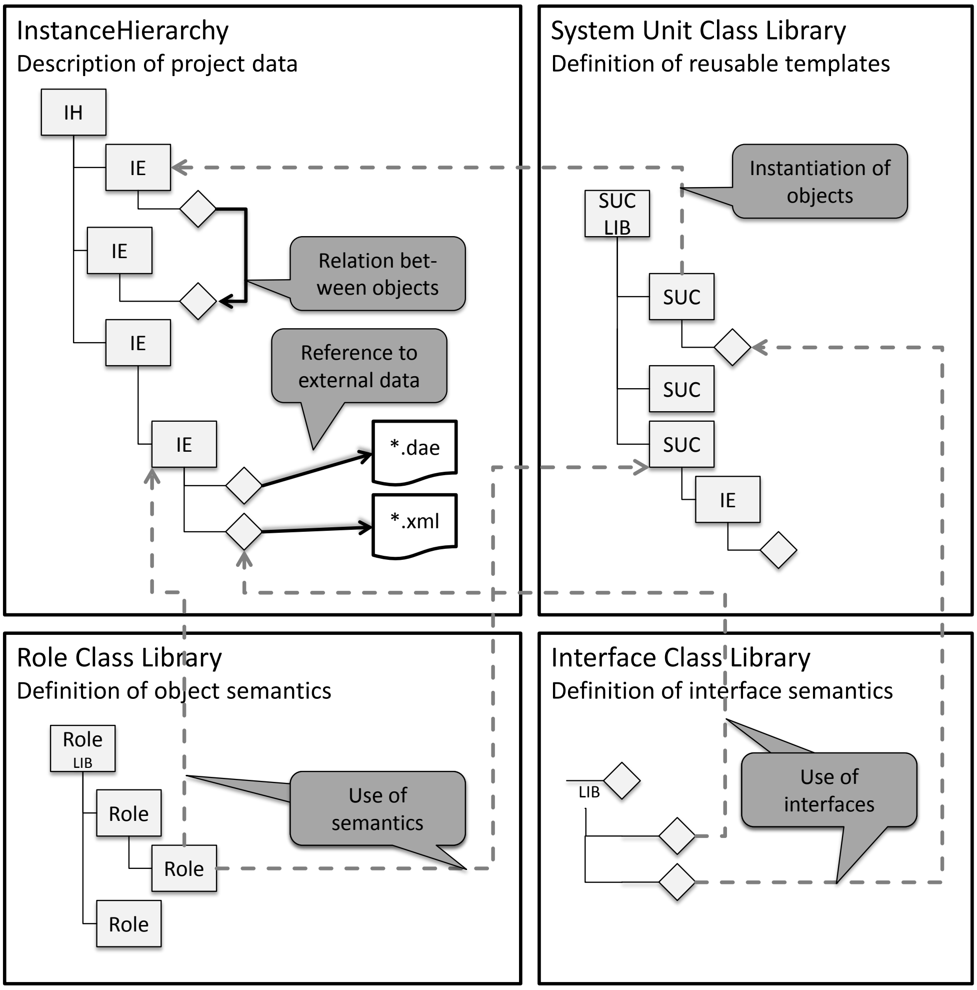

Figure 6.3: AutomationML's concepts in CAEX (based on [SL15])

References are realized as ExternalInterfaces that point to externally stored data. Relations are links between objects within the *.aml document. They are realized by creating ExternalInterface at each object or InternalElement to be linked. Those ExternalInterface are, then, connected by an InternalLink.
At each object, class, and interface attributes are allowed, which are used to model the corresponding properties of them.

Based on [LS17a] and [Aut14a] the AML application process comprises the following steps:

1. Analysis of domain and identification of relevant objects
2. Development of use case specific role classes
3. Development of use case specific interface classes
4. Development of use case specific system unit classes
5. Development of detailed information for the system unit classes
6. Creation of the (example) model by using defined libraries

The next subsections are structured according to these steps to finally store the EoL data model onto AML and, by doing this, create the basis for modeling and exchanging the (rough) description of the recovery process.
For this, the generalized base EoL data model, EoL process data model, and EoL data model from Section 5.1 are used to map on them - step by step - the necessary objects and classes. This will be visualized by several figures, in which the changes will be highlighted.

6.2.1 Analysis of End-of-Life domain and identification of relevant objects

Before the mapping onto AML can be executed, it needs to be decided about the concepts and modeling guidelines that are applicable in this context, i.e. the objects, syntax, and structuring of the EoL specific elements.
As mentioned in Subsection 5.2.9 the output of the application of the RPM can be seen as a first draft solution or rough description of the recovery process for the particular production system. As the author proposed, this should be further used and, therefore, made exchangeable, so that this rough description can be the basis for the detailed recovery process planning. Figure 6.4 shows the inputs and output.

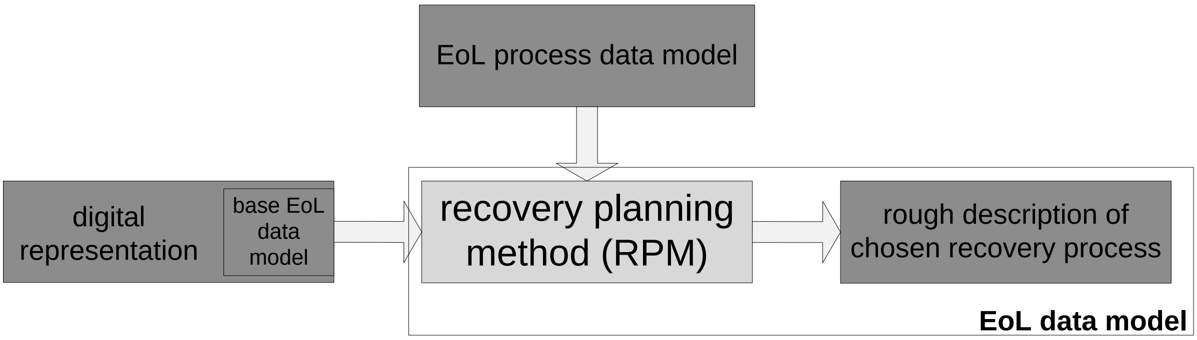

Figure 6.4: Recovery planning method with its inputs and output

The EoL data model (consisting of base EoL data model and EoL process data model) is necessary for the application of the RPM. But also the output - the rough description of the, then, chosen recovery process - should be expressible by this model.

Base End-of-Life data model

At first, the syntax of the base EoL data model needs to be modeled. Figure 6.5 shows how the UML classes and relations are realized syntactically in AML.

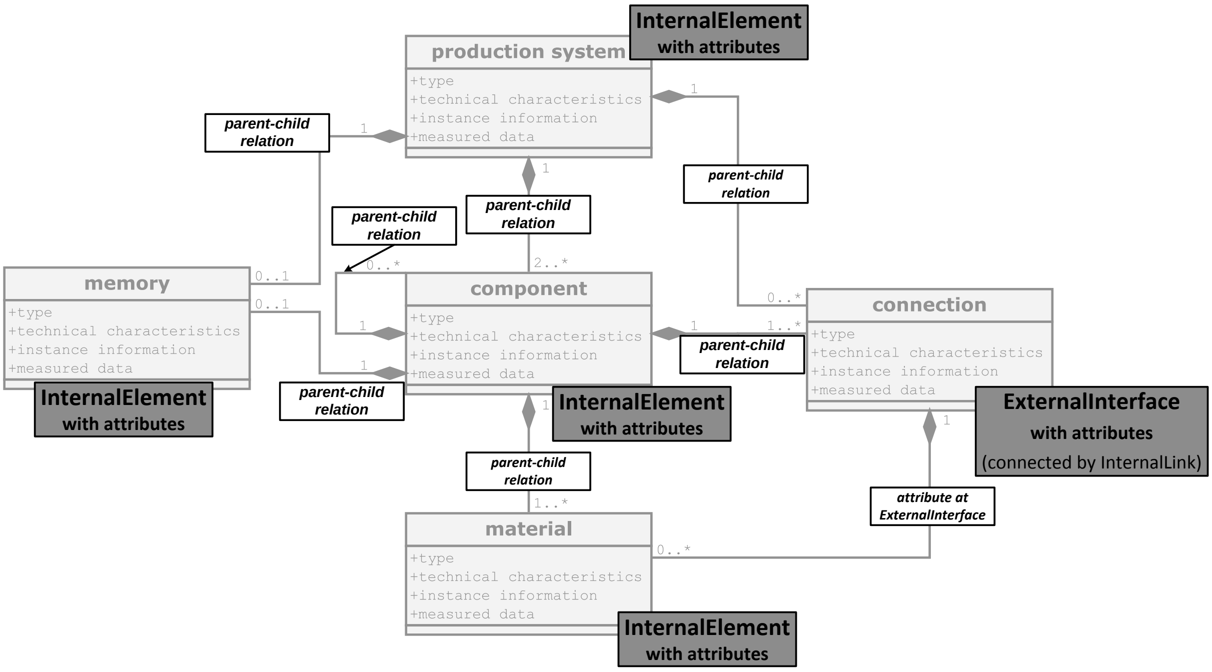

Figure 6.5: Base End-of-Life data model (generalized) syntactically mapped onto AutomationML

Regarding the UML class *material* there are two realization approaches possible shown in Figure 6.6. The first one is to model the UML class as an own InternalElement (depicted on the left side). The second approach is to model the UML class *material* as an attribute at the InternalElements (right side in Figure 6.6). The properties of material would be, then, modeled as subattributes.

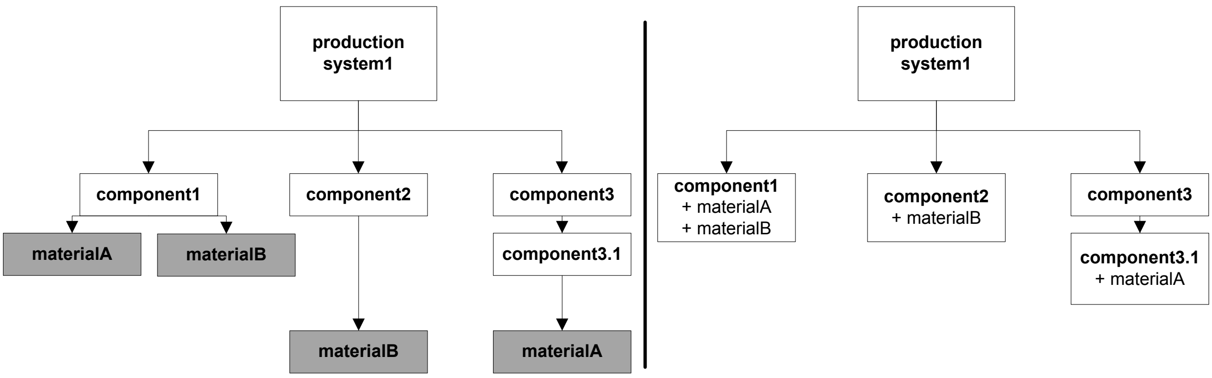

Figure 6.6: Two realization approaches for UML class material

The author recommends the first approach. For the recovery of material it is essential to be able to treat it in the same way like production system and component, since material recovery is one of the recovery goals. Therefore, the UML class *material* will be also modeled as an own object or InternalElement. Only this approach is discussed and applied in this thesis.
The approach - storing the material directly within the geometry and kinematics data format COLLADA - is not recommended due to the not avoidable distribution of the material properties. It is possible to create a <material> in <library_materials>

and give it a name and a visual representation. By creating <physics_material> in <library_physics_materials> it is possible to store properties, like friction or coefficient of restitution. But the property mass for example is defined in <rigid_body>. [ISO12]

The UML class *connection* is not considered as an own object or InternalElement. It will be considered as an interface modeled as ExternalInterface.

How the base EoL data model is structured (parent-child relation) or how many connections, components etc. it has, this is different from production system to production system. The base EoL data model is the input for the EoL phase but also the result of the engineering and operation phase. Therefore, the structure is considered as arbitrary for the EoL phase, since the structure was defined in the prior phases of the production system life cycle.
In the end, only the syntax can be considered as known.

End-of-Life process data model

Since the recovery process is a process, analyzing process models is reasonable. It can help to find a proper concept or modeling guideline in AML.
[Däh17] and [Mat15] have analyzed common process descriptions or process models, like SFC (Sequential Function Charts), GANTT charts, Petri nets, or VDI/VDE 3682 (formalized process descriptions). All descriptions have in common that they can be considered as graphs.
A graph is a model consisting of a set of vertices and of a set of edges. Vertices can represent, e.g. objects, states, situations, or positions. Edges can represent, e.g. objects or processes. Vertices are linked by edges. [Aut14a] [Nit09] How graph based structures can be modeled with AML is given in [Aut14a].

Figure 6.7 gives an overview of a graph model example that is mapped onto AML. Whereas the edge objects are pooled below an extra, own InternalElement, nothing like this is necessary for the vertex objects.

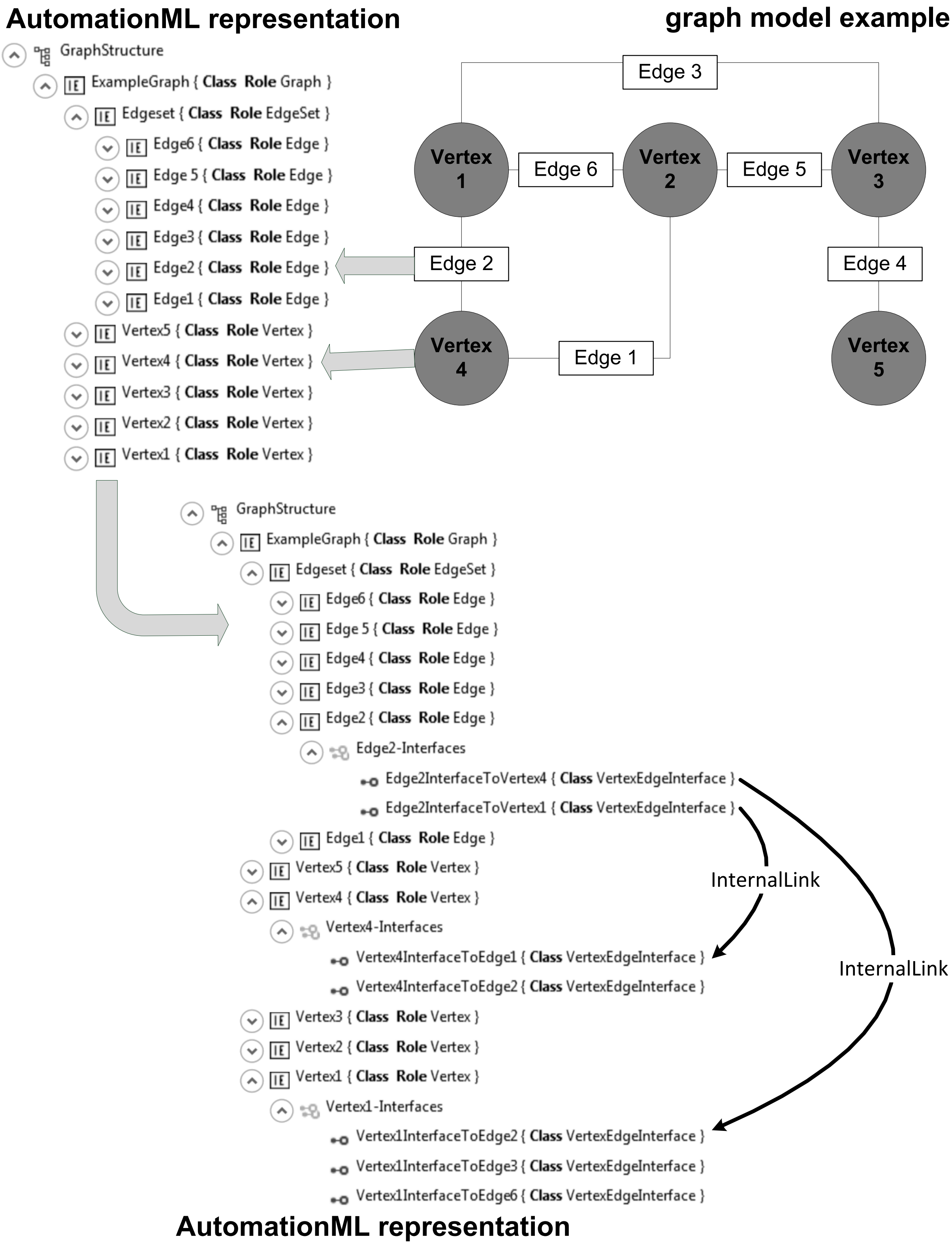

Figure 6.7: Graph model example mapped onto AutomationML (based on [Aut14a])

Edges are explicitly modeled as objects in order to assign properties to it. The edge objects themselves have ExternalInterface that are linked by InternalLinks to the ExternalInterfaces of the corresponding vertex objects. This is made by design. Otherwise the edge would only be modeled as a relation, i.e. as an InternalLink that connects the

ExternalInterfaces of the corresponding vertices. InternalLinks cannot have attributes. Additionally, the 'VertexEdgeInterface' interface class has an attribute 'direction' that indicates the direction of an edge. The syntax that results from the graph concept (including its structuring) is shown in Figure 6.8.
The relation between the UML classes *process* and *connection* is not modeled by AML means, since it is not modeled as an own InternalElement. The connection belongs, as indicated by the parent-child relation, to the production system and/or the component. In case of a disassembly process this connection would be considered as a property of the production system or component used to derive an appropriate disassembly technique.

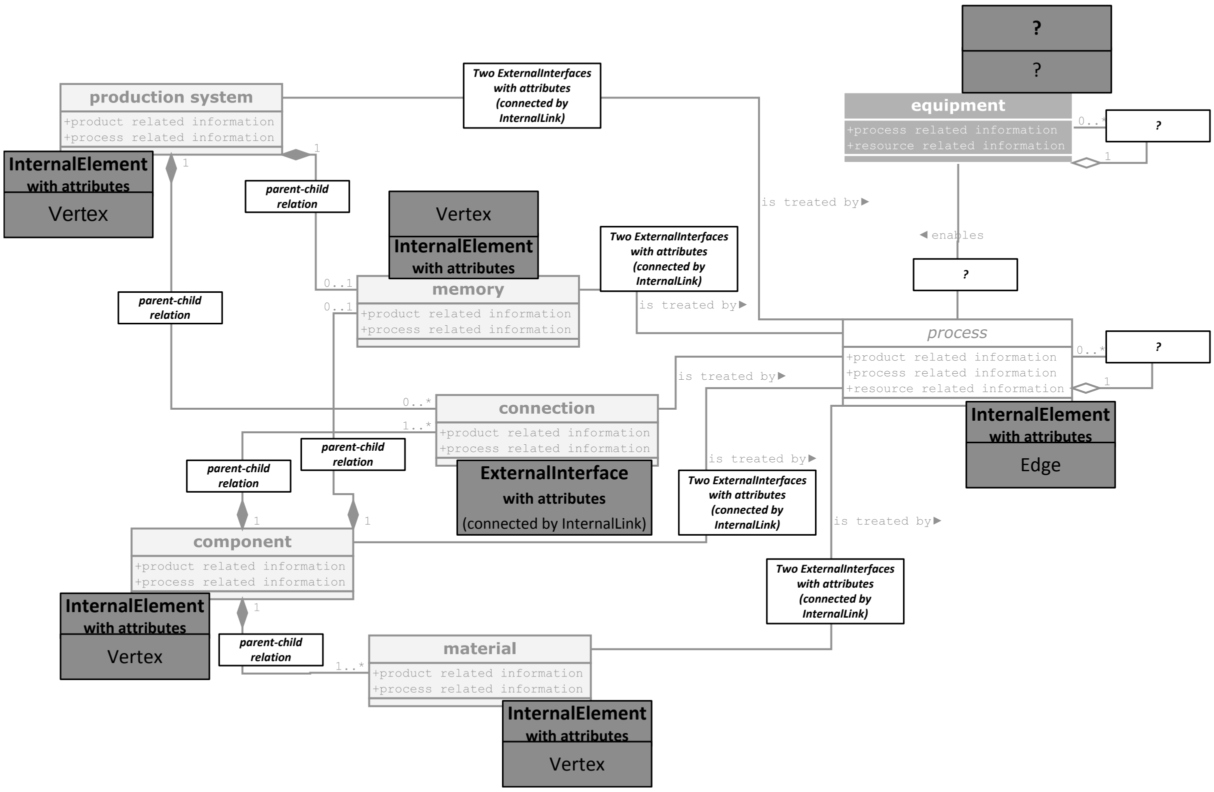

Figure 6.8: End-of-Life process data model syntactically mapped onto AutomationML according to the graph concept

Some question marks or open issues are still existent - also visualized in this figure. That means that the concept does not cover all the aspects in mapping it onto AML. So, other concepts need to be applied additionally, which is a viable approach [LSC+16].
The EoL process data model has a product, process, and resource part. One open issue is the modeling of the resources that are assigned to the processes. For this, the author proposes to apply the AML's PPR modeling concept - see [IEC14]) or [Dra10]. Figure 6.10 gives an overview of an PPR example that is mapped onto AML. The product, process, and resource objects are pooled below an extra, own InternalElement. Products, processes, and resources are modeled as objects. All of them are equipped with an ExternalInterface, which is connected by InternalLinks. This 'PPRConnector' interface class has no attributes. [IEC14] mentions that in case a direction is needed, the objects should be equipped each with an additional ExternalInterface that uses the interface class 'order'. This standard interface class has an attribute 'direction'. But it is not modeled in the figures.

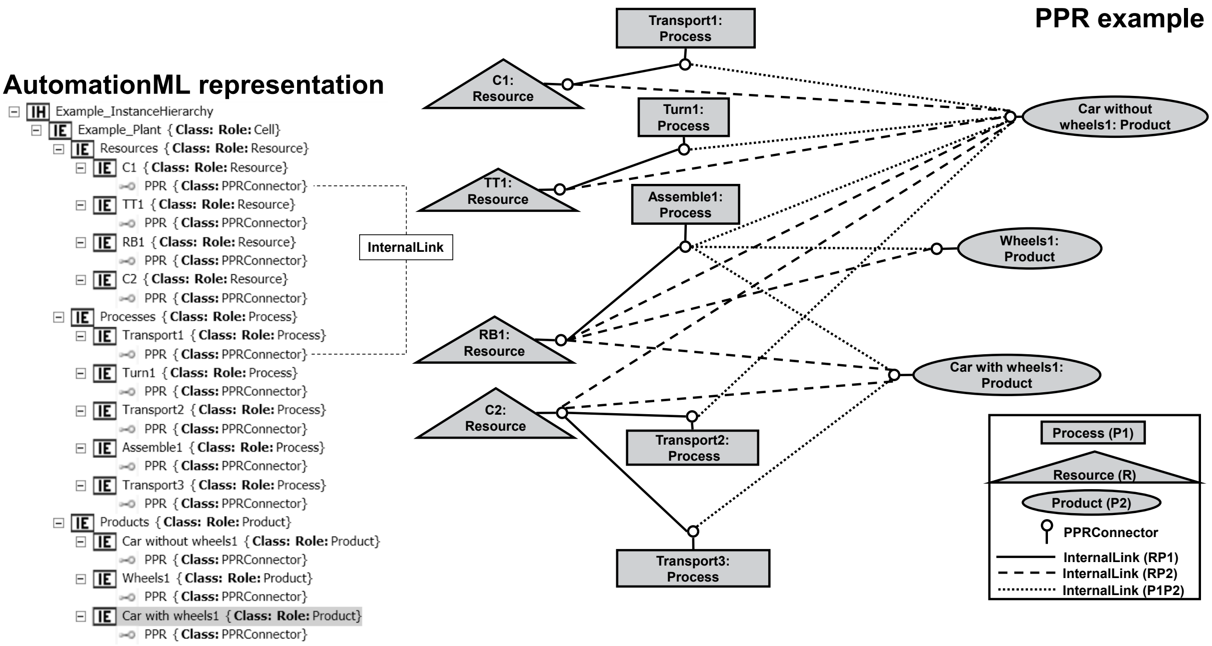

Figure 6.9: PPR example mapped onto AutomationML (based on [IEC14])

In case, one of the parts of the PPR concept is the leading one, InternalLinks can be saved. The following Figure 6.10 shows a resource centric representation, where the InternalLinks between the products and the resources are neglected.

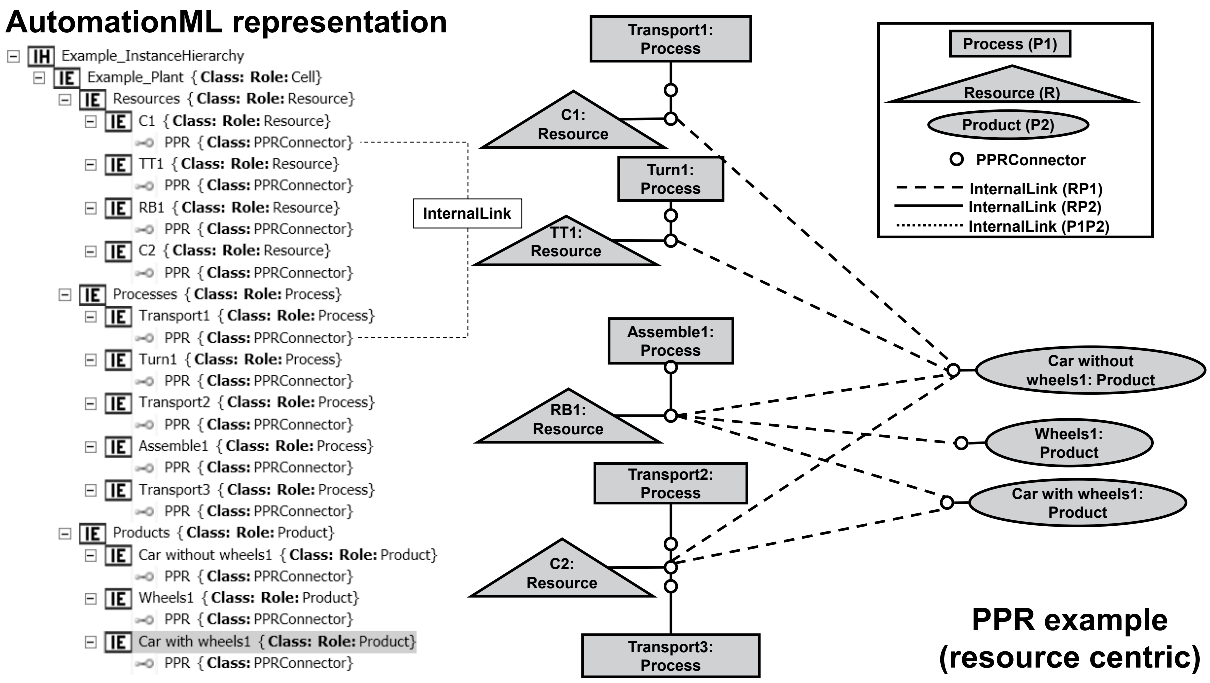

Figure 6.10: PPR example (resource centric) mapped onto AutomationML (based on [IEC14])

In this thesis, the product centric PPR concept is used, because the production system to be recovered in the focus of the recovery process.
In contrast to the graph concept, the PPR concept can also model the resource part of the EoL process data model. The PPR concept uses one ExternalInterface at the objects

that, then, links other ExternalInterfaces by InternalLinks. In contrast, the graph concept uses as many ExternalInterfaces as edges exist. This is schematically visualized in Figure 6.11.

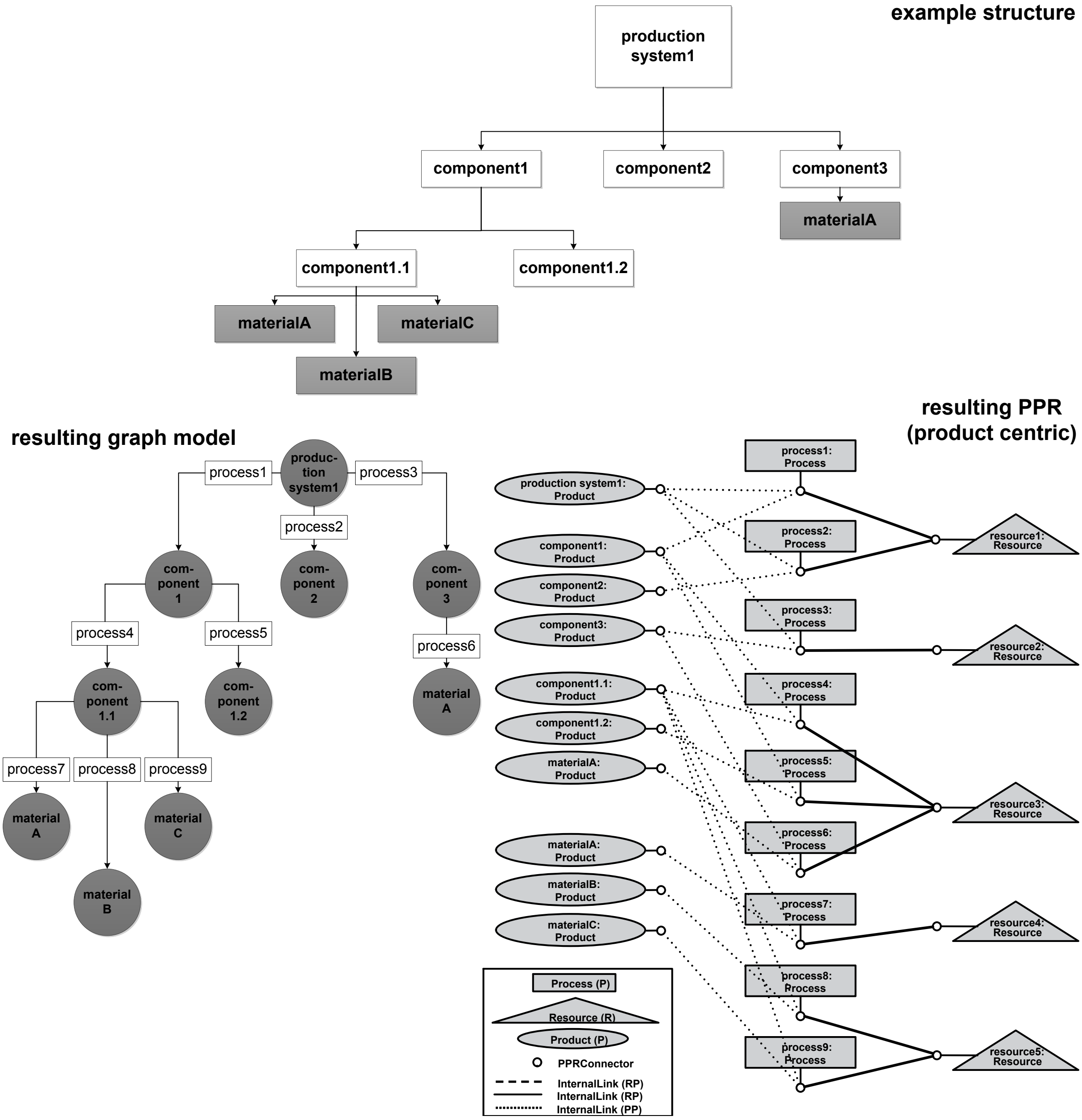

Figure 6.11: Example structure modeled with graph concept and PPR concept

To avoid modeling redundant information the graph concept and the PPR concept are combined. Since the graph concept already covers the relationship between product and process, this will be neglected in the PPR concept (see Figure 6.12). Thus, the PPR concept becomes an PR concept (Process-Resource) - similar to the approach given in [JCSF12]. In the following this concept is still called PPR concept.

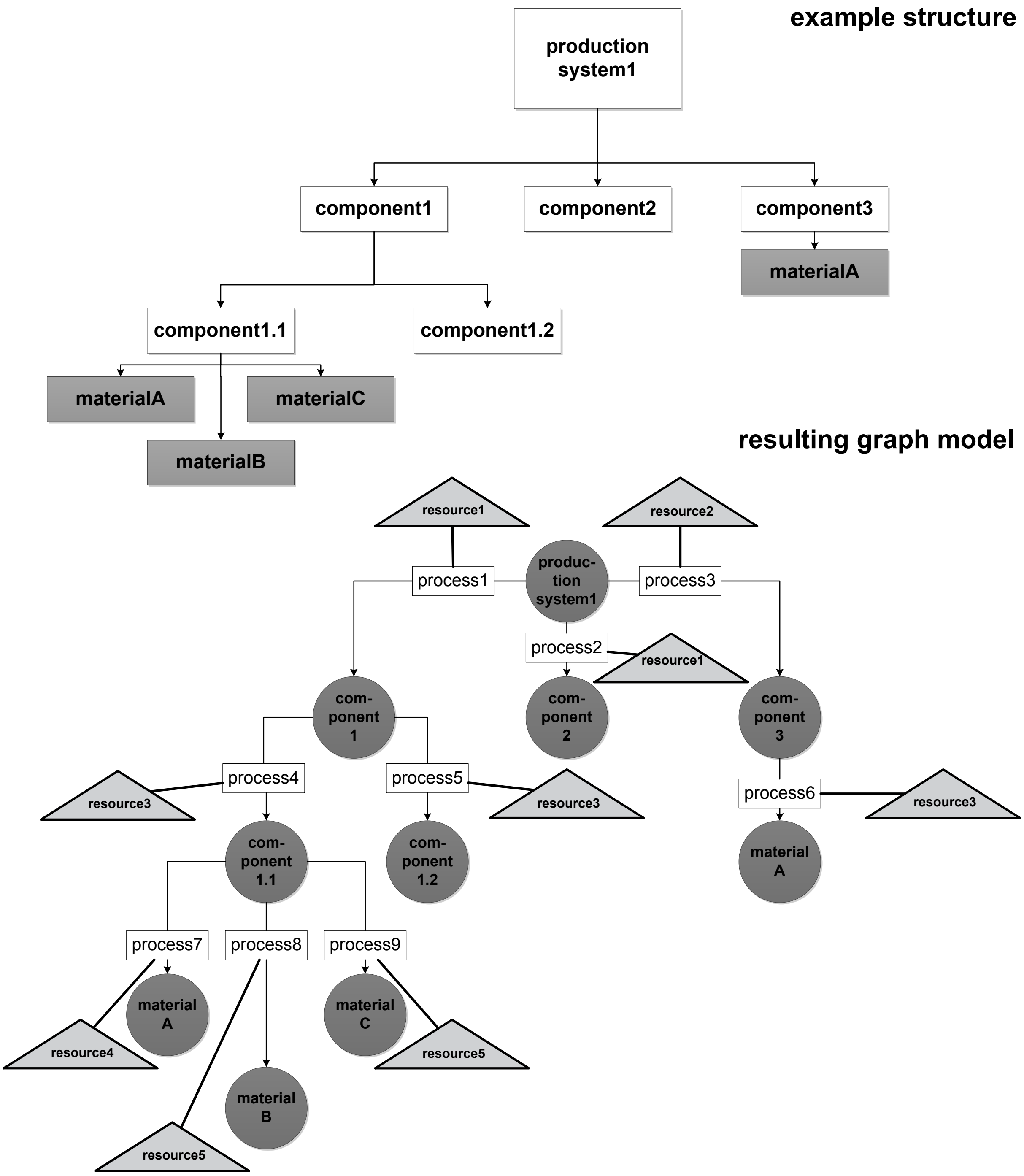

Figure 6.12: Example structure modeled with the combination of graph concept and PPR concept

The syntax that results from the graph concept and product centric PPR concept (including its structuring) is shown in Figure 6.13.

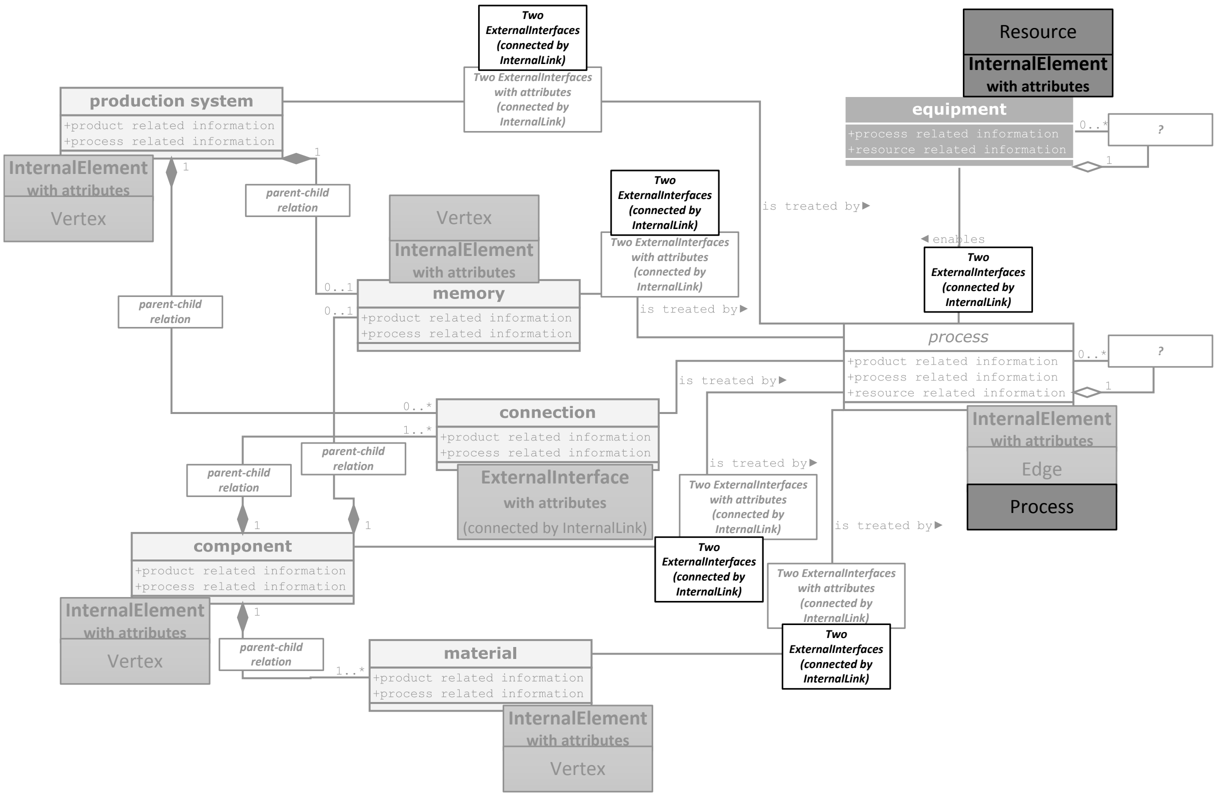

Figure 6.13: End-of-Life process data model (unified and generalized) syntactically mapped onto AutomationML according to the graph concept and PPR concept

To further describe the processes, it would be also possible to equip the corresponding InternalElement with an ExternalInterface that references a PLCopen XML document that contains further information about the process, see [LHK10] and [Aut17b]. But this is not considered in this thesis. Further information is modeled as attributes instead.

Some question marks or open issues are still existent - also visualized in this figure. These address the decomposition of processes and resources.
In [JCSF12] a similar modeling approach is chosen. Here, the formalized process description (standardized in [VDI15a] and [VDI15b]) is mapped onto AML.
Product, process, and resources are grouped below extra, own InternalElements. With ExternalInterfaces connected by InternalLinks the process is described how input products are converted through a process into output products. To each process a resource is assigned.
For the decomposition of processes [JCSF12] proposes an approach as shown in Figure 6.14.

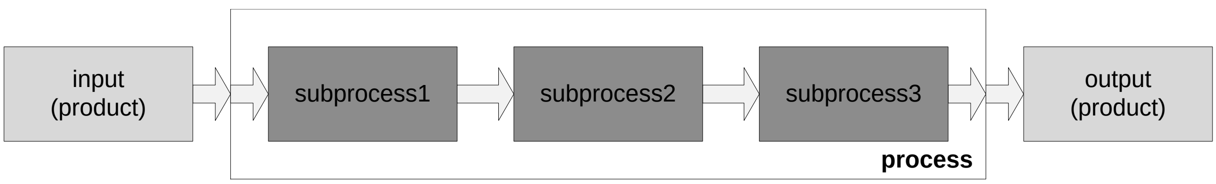

Figure 6.14: Decomposition of processes according to [JCSF12]

However, in [JCSF12] it is not mentioned how the subprocesses can get brought into an order. From the graph concept point of view, a process followed by a process is not allowed since each process has a product as input and a product as output. Regarding Figure 6.14 this would result in an additional intermediate product, one after subprocess 1 and 2.
On the other hand, AutomationML shall allow to store non-complete and rough descriptions. In order to follow this idea, the author applies the approach from Figure 6.14, but extends it by the opportunity to express the sequence of those subprocesses.
The same approach is applied for the subresources.
The syntax that results from the graph concept with the product centric PPR concept and the formalized process description approach (including its structuring) is shown in Figure 6.15.

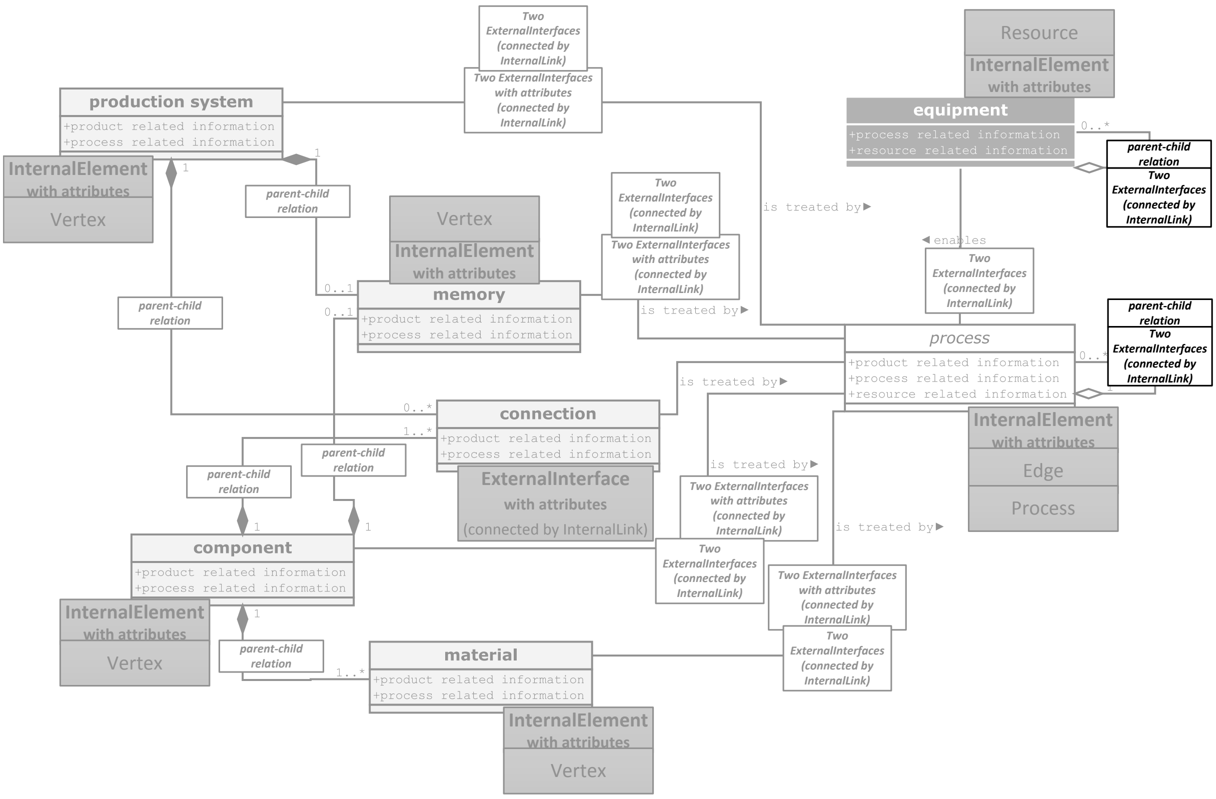

Figure 6.15: End-of-Life process data model syntactically mapped onto AutomationML according to the graph concept and PPR concept (additionally with formalized process description approach)

End-of-Life data model

The product objects are not pooled below an extra, own InternalElement 'Products' as it is proposed by the PPR concept and the formalized process description approach. When having the already existing product structure - or in the context of this thesis production system structure - from the base EoL data model, it is not easily possible to restructure it entirely. When considering the mentioned object identification concept of AML, direct copies of objects are not recommended since one object should only exist one time to keep it consistent (no branching, no object with two different IDs).

Restructuring of a structure is possible only by applying the mirror concept (see [IEC14]). With this concept objects can occur in different hierarchies. The mirror concept has master objects and mirror objects. Mirror objects are only pointers that are pointing to the master object. Changes and modifications are only made at the master object. Below a mirror object no structure, content, or containment is allowed. Thus, only the processes (in line with the graph concept) and the resources are pooled below an extra, own InternalElement.

The syntax that results from the previous subsubsections and this subsubsection is shown in Figure 6.16.

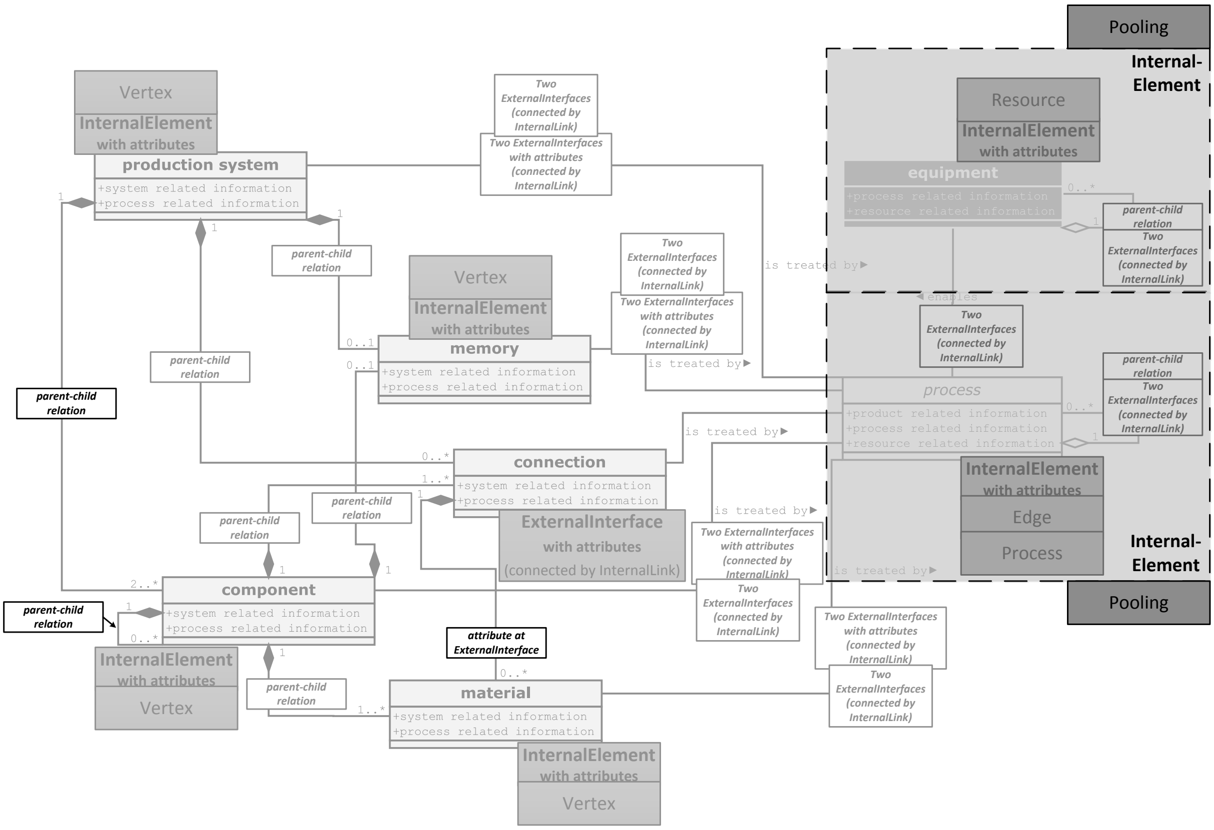

Figure 6.16: End-of-Life data model (unified and generalized) syntactically mapped onto AutomationML

If the production system structure needs to be extended in order to add intermediate products to it, i.e. components or materials, the EoL data model shall allow this.

In the next subsections the corresponding libraries are developed.

6.2.2 Development of End-of-Life specific role classes

Besides the objects with their syntax and structuring, the semantics is the third important step to map the EoL data model onto AML. This subsection deals with the object semantics modeled as role classes in AML.

Base End-of-Life data model

According to Section 5.1, each UML class has the property *type*. This is representing the functionality, the meaning, the semantics of this UML class.
As mentioned in Subsubsection 6.2.1, the structure of the base EoL data model is arbitrary, since it is the result of the engineering and operation phase. For the same reason also the semantics of the base EoL data model is arbitrary and is, therefore, not further considered in this thesis - Figure 6.17.

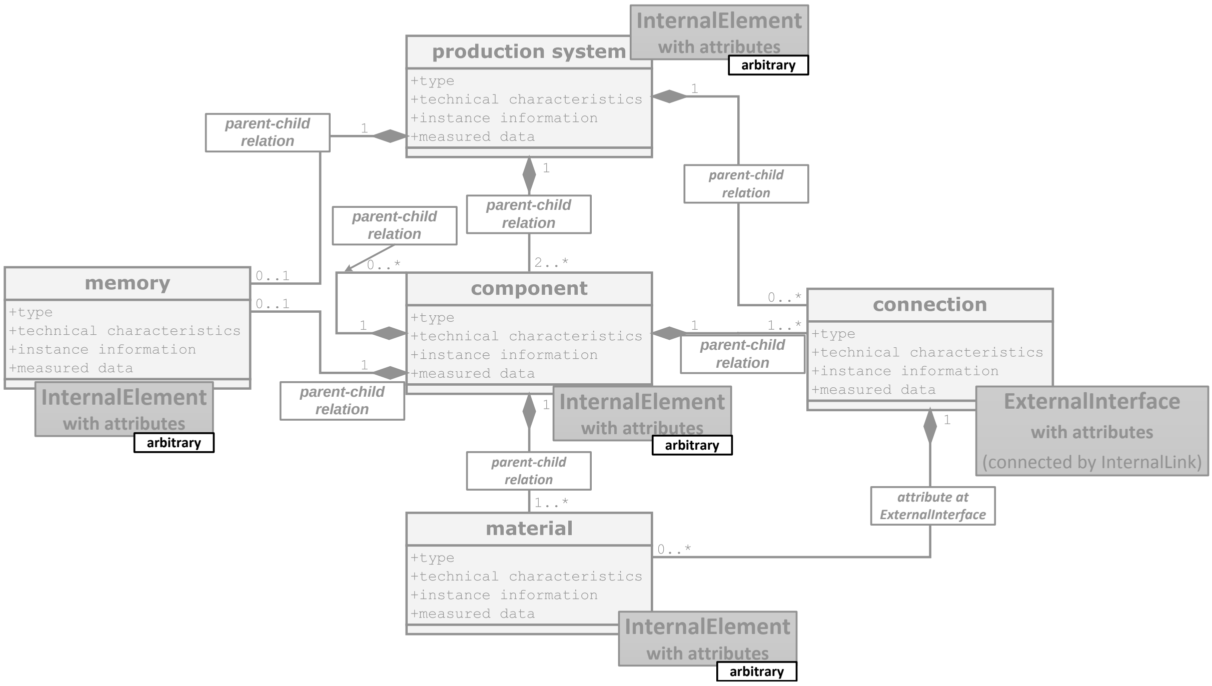

Figure 6.17: Base End-of-Life data model (generalized) mapped onto AutomationML regarding role classes

It might be imaginable that in the future a basic object semantics is specified or standardized, which is generally accepted and applied in industry. Several authors have developed a hierarchy to logically group similar components as presented in [LSH$^+$17b]. The EoL phase would benefit from that. For the material classification there are standards already existing, e.g. VDA 260 [VDA07] (applied in VDI 2074 [VDI14]), or VDA 231-106 [VDA97].
However, the AutomationML e.V. specifies stepwise certain aspects of the engineering process regarding structuring and object semantics, e.g. for communication networks [Aut14b] or for automation project configuration [Aut17a]. But the entire engineering phase (or operation phase) is not specified in every aspect, yet.

End-of-Life process data model

As the base EoL data model is arbitrary, i.e. the experts from the EoL phase have no influence on it, the experts from the EoL phase do have influence on the EoL process data model. Because this model includes information that is gathered or collected at the EoL point in time.

The previous subsection has shown that several concepts are needed to meet the EoL specific needs of the EoL data model. The different concepts also come with different semantics. For this, the capability of AML is used to model multiple role classes at one InternalElement. An InternalElement can only have one 'RoleRequirements' that represents the main role or meaning of an object on the one hand. But on the other hand an InternalElement can have zero or more 'SupportedRoleClass'es with which all the other meanings of an objects can be expressed. [IEC14] It is common that objects has several meanings or functions, e.g. when thinking of smart phones, computers, or washing machines.

For the graph concept, the following role classes are proposed (see Table 6.1). To group them, an EoL specific role class library is created (called AutomationMLEOLRoleClassLib). The two role classes for the graph concept inherit from the 'Vertex' and 'Edge' role classes from the AutomatioMLGraphRoleClassLib, which is specified in [Aut14a].

Table 6.1: RoleClassLibrary AutomationMLEOLRoleClassLib - graph concept part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLVertexRoleClass	AutomationMLEOLRoleClassLib/EOLVertexRoleClass	AutomatioMLGraphRoleClassLib/ Vertex
EOLEdgeRoleClass	AutomationMLEOLRoleClassLib/EOLEdgeRoleClass	AutomatioMLGraphRoleClassLib/ Edge

Besides the name, also the structural path to the role class is provided in the table. According to [IEC14], classes are identified by their names. The referencing is technically done by using this path to the role class that needs to be unambiguous, i.e. the names on the same level in the path need to be unique.
Another requirement is that all role classes shall inherit from the role class AutomationMLBaseRole to be compliant with AML. Therefore, also the inheritance path of each role class is provided.

The next part is the semantics for the PPR concept.
When looking at the content of the RoleClassLibraries defined in IEC 62714-2 [IEC15a], they are basically structured according to the three different types of industries: discrete manufacturing industry (DMI), continuous manufacturing industry (CMI), and batch manufacturing industry (BMI). Since the recovery process is a discrete process as well, role classes from DMI library might be usable when thinking of undoing connections with equipment of the manufacturing process. However, the RoleClassLibraries defined in IEC 62714-2 [IEC15a] are not applied due to the different scope. Instead the more general role classes from the base library (specified in IEC 62714-1 [IEC14]) are used. In particular the role classes 'Product', 'Process', and 'Resource' are used to express the PPR concept. Those general role classes are specialized and applied for EoL purposes, i.e. EoL specific derivations of them are created.

The EoL specific role class library - AutomationMLEOLRoleClassLib - is extended according to the PPR concept (except for the product part): EOLProcessRoleClass and EOLResourceRoleClass (see Table 6.2 and 6.3). This is in line with the structuring in IEC 62714-2 [IEC15a].

Table 6.2: RoleClassLibrary AutomationMLEOLRoleClassLib - PPR, process part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLProcessRoleClass	AutomationMLEOLRoleClassLib/EOLProcessRoleClass	AutomationMLBaseRoleClassLib/ AutomationMLBaseRole/Process

Table 6.3: RoleClassLibrary AutomationMLEOLRoleClassLib - PPR, resource part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLResourceRoleClass	AutomationMLEOLRoleClassLib/EOLResourceRoleClass	AutomationMLBaseRoleClassLib/ AutomationMLBaseRole/Resource

Regarding step 5 ’creating process chain and characterizing processes’ of the RPM attributes (in the five performance indices) are needed at the processes - see Table 6.4.

Table 6.4: Attributes of AutomationMLEOLRoleClassLib/EOLProcessRoleClass

role class	role class attributes
EOLProcessRoleClass	ProcessEconomic, ProcessEcological, ProcessSocietal, ProcessTime, ProcessQuality

For the formalized process description approach no additional role classes are needed.

For the RPM another set of role classes is needed, which are describing the recovery goal, i.e. the future application of the production system, component, and/or material. This is not explicitly modeled as a structural element in the EoL process data model but as a property called *recovery goal* (see Figure 3.4). This corresponds with step 2 ’characterizing recovery goal’ of the RPM (see Subsection 5.2.2). When modeling the future requirements, these can be compared with the current state of production system, component, and material. Therefore, the role class called EOLRecoveryGoalRoleClass is created - see Table 6.5.

Table 6.5: RoleClassLibrary AutomationMLEOLRoleClassLib - recovery goal part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLRecoveryGoalRoleClass	AutomationMLEOLRoleClassLib/ EOLRecoveryGoalRoleClass	AutomationMLBaseRoleClassLib/ AutomationMLBaseRole

Additionally, at this role class there are attributes created in accordance with the RPM (see Figure 5.24). Here, in step 2 ’characterizing recovery goal’ the future application of the production system, component, and/or material (in the five performance indices) is needed. Therefore, these attributes are modeled at the EOLRecoveryGoalRoleClass - see the following Table 6.6.

Table 6.6: Attributes of AutomationMLEOLRoleClassLib/EOLRecoveryGoalRoleClass

role class	role class attributes
EOLRecoveryGoalRoleClass	SystemFutureEconomic, SystemFutureEcological, SystemFutureSocietal, SystemFuture-Time, SystemFutureQuality

Each role class can have properties realized as attributes. The purpose of those at the role classes is to indicate, which properties usually come with a certain semantics. Those attributes have no value, because role classes are vendor independent and generic. In the first place it is just about the existence of them.

Figure 6.18 visualizes again, at which InternalElements the role classes of the graph and PPR concept are applied (white boxes at the InternalElements). Additionally, the mentioned recovery goal semantics is added.

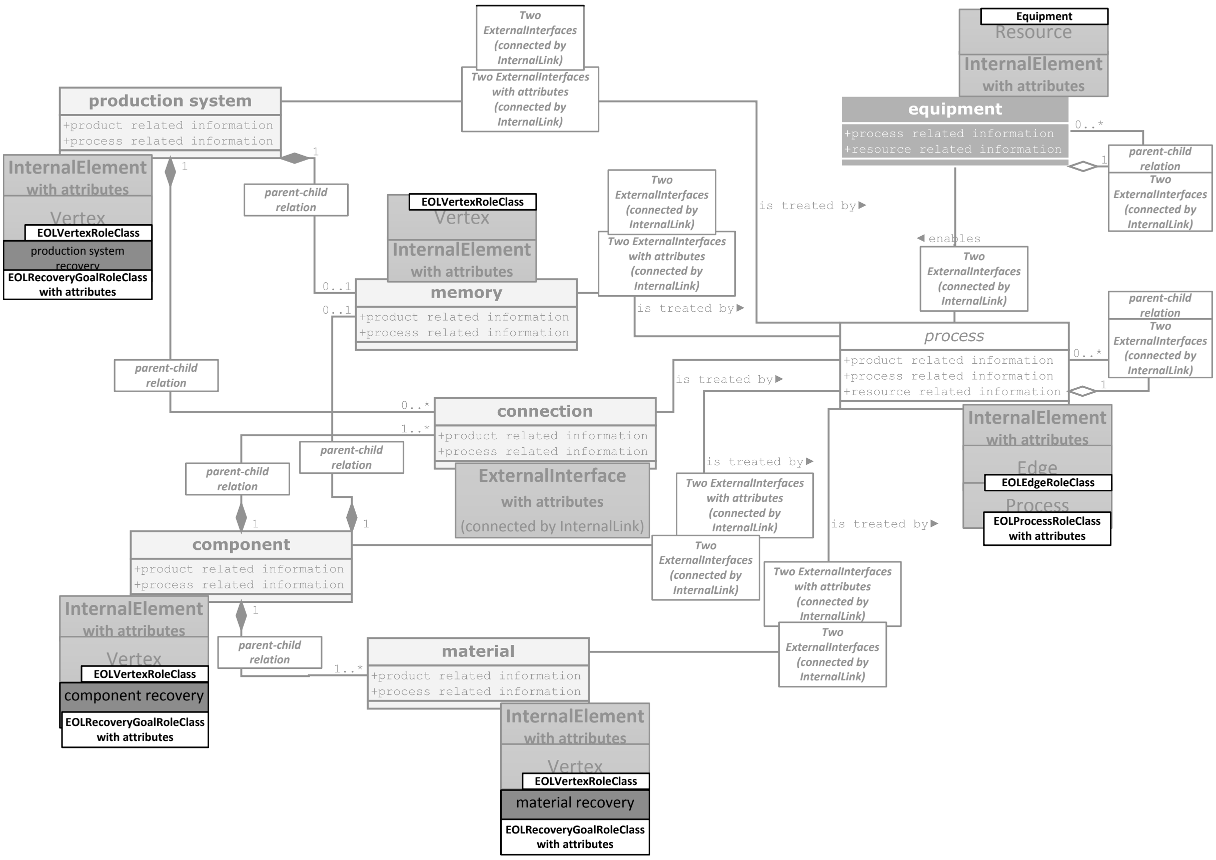

Figure 6.18: End-of-Life process data model (unified and generalized) mapped onto AutomationML regarding role classes

In this figure it can be seen that some InternalElements have more than one role class assigned to it. As already indicated, AML supports multiple role classes.

A standardization or specification of process semantics seems reasonable. For example, the German standard family DIN 8580 [DIN03a] provides a classification and terminology for manufacturing processes. The separating processes are described in DIN 8588, DIN 8589, DIN 8200, DIN 8590, DIN 8591, and DIN 8592. Disassembly processes are

specified in DIN 8591 [DIN03b]. These could be used as standard semantics for the process objects.
The author has translated this disassembly classification into English and has modeled it as AML role classes - see Figure 6.19.

Figure 6.19: DIN 8591 (disassembly processes) as AutomationML role classes

An XML representation of the RoleClassLib of DIN 8591 for disassembly processes is given in Annex H.

Another example could be the standardization of the preprocessing processes necessary for recycling. The author has created an AML role class library with common preprocessing processes based on [Mar11], [Nic96], and [Ott16] - see Figure 6.20.

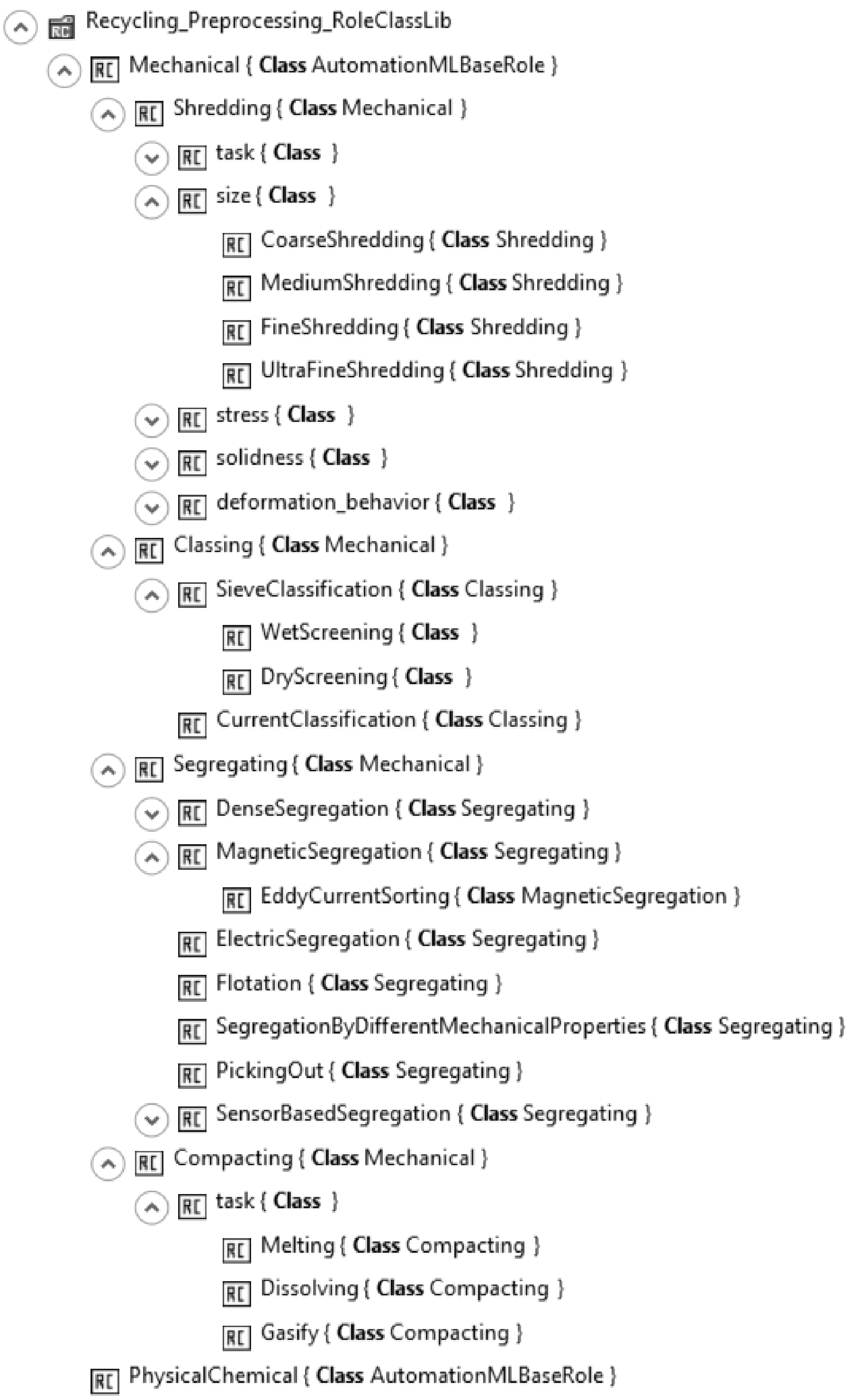

Figure 6.20: Preprocessing processes as AutomationML role classes

An XML representation of the RoleClassLib for preprocessing processes is given in Annex I.

In another step the semantics for the equipment objects could be standardized. IEC 62714-2 provides a simple role class library 'AutomationMLDMIRoleClassLib' that contains the role classes, e.g. tool, robot, or machine [IEC15a]. But also more advanced equipment semantics can be found, e.g. provided by eCl@ss [eCl].

End-of-Life data model

The two InternalElements that intend to group the resources and processes do not necessarily need an extra role class assigned to them. But to fully semantically describe the EoL data model, the author proposes the two role classes listed in Table 6.7.

Table 6.7: RoleClassLibrary AutomationMLEOLRoleClassLib - structuring part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLProcessSetRoleClass	AutomationMLEOLRoleClassLib/ EOLProcessSetRoleClass	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/ Structure/ProcessStructure
EOLResourceSetRoleClass	AutomationMLEOLRoleClassLib/ EOLResourceSetRoleClass	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/ Structure/ResourceStructure

As mentioned before, processes can have subprocesses and resources subresources. For those hierarchy levels no additional semantics is created here, except for one level and only in the process hierarchy: for the EoL strategy. The EoL strategy represents a process chain. This is not explicitly modeled neither as a structural element in the EoL data model nor as a property (see Figure 3.4). However, this correlates with step 4 ’choosing EoL strategy’ of the RPM (see Subsection 5.2.4). It is also needed when applying the library concept for the RPM as shown in Annex G. The additional role class needed for the RPM is given in Table 6.8.

Table 6.8: RoleClassLibrary AutomationMLEOLRoleClassLib - EoL strategy part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLStrategyRoleClass	AutomationMLEOLRoleClassLib/EOLStrategyRoleClass	AutomationMLBaseRoleClassLib/ AutomationMLBaseRole

This role class has the following attributes - see Table 6.9.

Table 6.9: Attributes of AutomationMLEOLRoleClassLib/EOLStrategyRoleClass

role class	role class attributes
EOLStrategyRoleClass	EOLStrategyEconomic, EOLStrategyEcological, EOLStrategySocietal, EOLStrategy-Time, EOLStrategyQuality

For the RPM another role class is needed (see Table 6.10), which is describing a performance index. This is not explicitly modeled neither as a structural element in the EoL data model nor as a property (see Figure 3.4). However, this correlates with step 6 ’determining and weighting indices’ of the RPM (see Subsection 5.2.6) and the characterization of the recovery goal and its comparison with the corresponding current state. It is also needed when applying the library concept for the RPM as shown in Annex G.

Table 6.10: RoleClassLibrary AutomationMLEOLRoleClassLib - performance index part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLPerformanceIndexRoleClass	AutomationMLEOLRoleClassLib/ EOLPerformanceIndexRoleClass	AutomationMLBaseRoleClassLib/AutomationMLBaseRole

Figure 6.21 visualizes, at which InternalElements the role classes are applied (white boxes at the InternalElements). Additionally, the EoL strategy and performance index semantics is added.

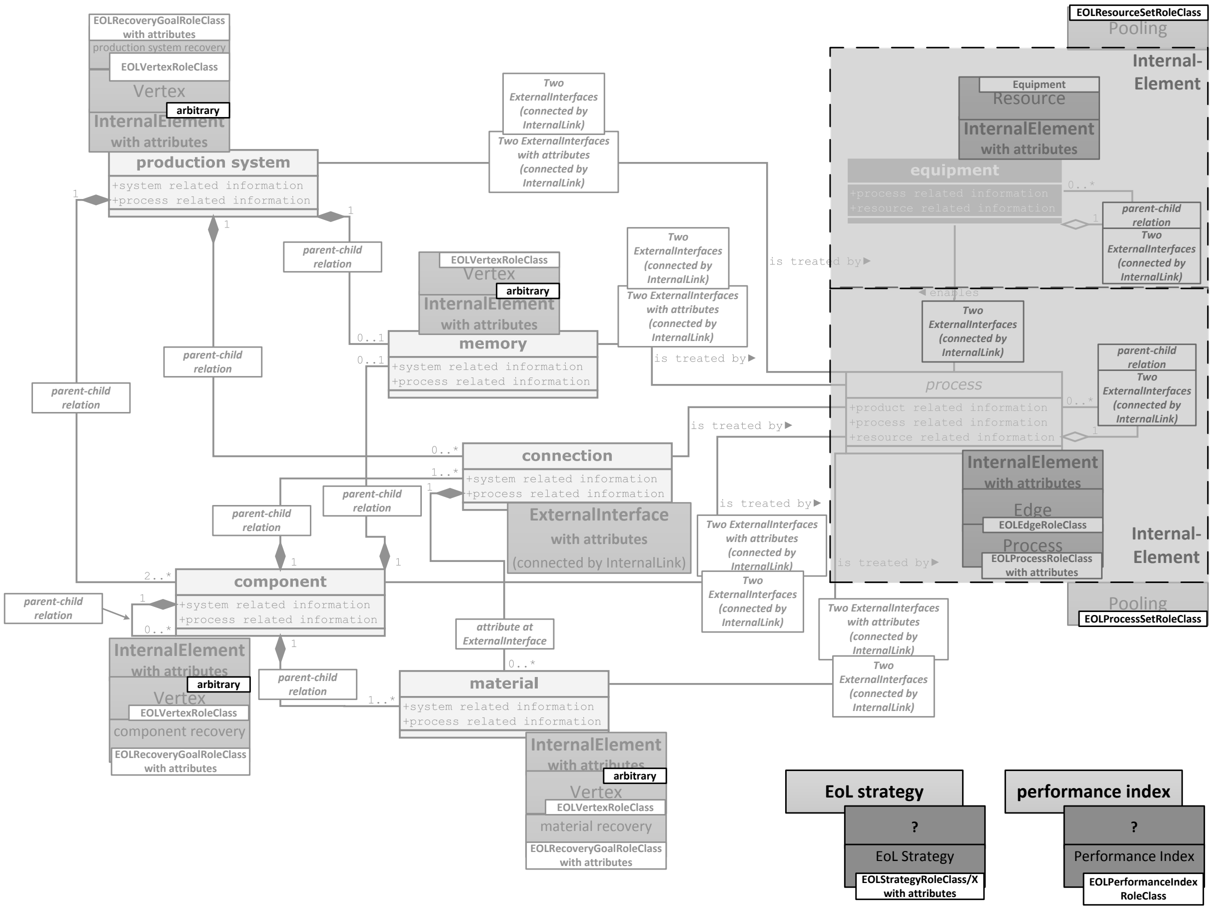

Figure 6.21: End-of-Life data model (unified and generalized) mapped onto AutomationML regarding role classes

The EoL data model is the result of the base EoL data model and the EoL process data model. As mentioned before, the object semantics and the structuring of the base EoL data model is arbitrary. This arbitrary object semantics and structuring is kept and on top of this, the object semantics and structuring of the EoL process data model is modeled.

The next subsection considers the necessary interface classes.

6.2.3 Development of End-of-Life specific interface classes

This subsection deals with the interface semantics modeled as interface classes in AML.

Base End-of-Life data model

One requirement on the base EoL data model is that connections or interfaces need to be modeled explicitly. It is not sufficient that the hierarchy of the production system is expressed by parent-child relations, i.e. a containment relation. This would only express that there is a relationship but the semantics of the relationship cannot be modeled by this parent-child relation, e.g. a production system consists of four components. In this scope of the thesis the interface semantics is needed, especially the mechanical connections are of interest. These need to be explicitly modeled since they are the basis for the disassembly.
The base EoL data model is the result of the engineering and operation phase. So, these phases shall pay attention that the mechanical interfaces are explicitly modeled. But also because of this reason the interface semantics is considered as arbitrary for the EoL phase, since the interfaces or connections were defined in the prior phases of the production system life cycle.
As mentioned, the parent-child relation cannot have any semantics. Therefore, no interface classes are created at all. This is visualized by the gray, italic boxes at the relations in Figure 6.22.

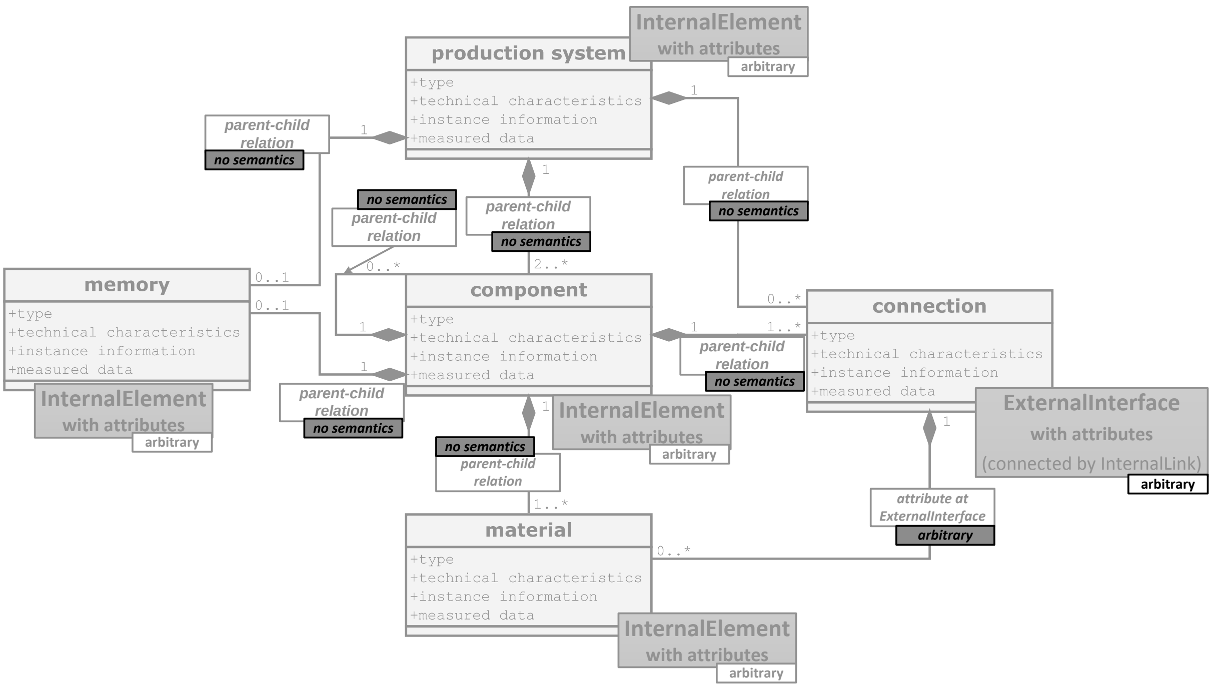

Figure 6.22: Base End-of-Life data model (generalized) mapped onto AutomationML regarding role and interface classes

However, a standardization or specification of interface semantics, to some extent, would support the EoL phase. For example, the German standard family DIN 8580 [DIN03a] provides a classification and terminology for manufacturing processes. The eight parts

of the DIN 8593 [DIN03c] are especially focusing on joining processes (like welding or assembling). This joining classification can be used as interface semantics for the mechanical connections. The EoL phase could, then, use this to derive appropriate destructive, semi-, or non-destructive disassembly processes or techniques. A possible disassembly processes role class library (based on DIN 8591), which could be used for this, was introduced in the previous subsection.
The author has translated this joining classification into English and has modeled it as AML interface classes - see Figure 6.23.

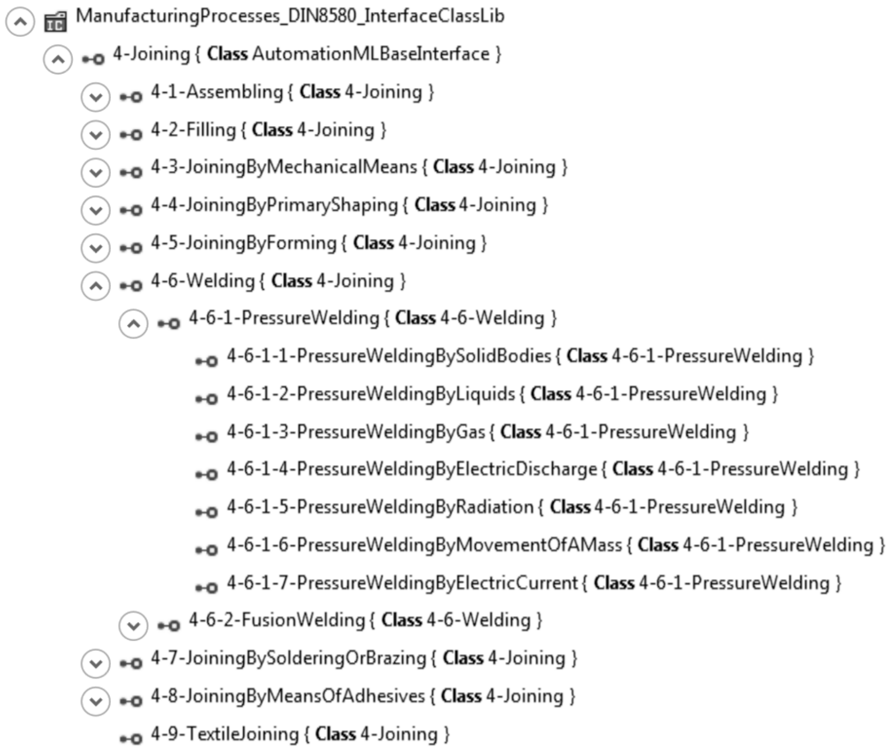

Figure 6.23: DIN 8593 (joining processes) as AutomationML interface classes

An XML representation of the InterfaceClassLib of DIN 8593 for joining processes is given in Annex J.

End-of-Life process data model

The graph concept needs at each vertex and edge an own ExternalInterface that is connected by an InternalLink. For each edge that is connected to a vertex one ExternalInterface is created at the vertex. All these interfaces have the same semantics. The created interface class is given in the following Table 6.11. This is stored in an EoL specific interface class library - called AutomationMLEOLInterfaceClassLib.

Table 6.11: InterfaceClassLibrary AutomationMLEOLInterfaceClassLib - graph concept part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLVertexEdgeInterfaceClass	AutomationMLEOLInterfaceClassLib/ EOLVertexEdgeInterfaceClass	AutomationMLGraphInterfaceClassLib/ VertexEdgeInterface

EOLVertexEdgeInterfaceClass is derived from the VertexEdgeInterface of the AutomationMLGraphInterfaceClassLib specified in [Aut14a]. EOLVertexEdgeInterfaceClass inherits one attribute from VertexEdgeInterface - see Table 6.12.

Table 6.12: Attributes of AutomationMLEOLInterfaceClassLib/ EOLVertexEdgeInterfaceClass

role class	role class attributes
EOLVertexEdgeInterfaceClass	Direction (inherited)

This attribute can have the value: In, Out, or InOut [Aut14a]. To express an edge that connects vertex A with vertex B, this attribute would be used as follows: Vertex A has an ExternalInterface with the direction 'Out' that is connected to one ExternalInterface of the edge with the direction 'In'. The second ExternalInterface of the edge has the direction 'Out' that is connected to the ExternalInterface of vertex B with the direction 'In'. In case the direction would not matter, i.e. this would be an undirected edge, the attribute value 'InOut' can be used.

For the PPR concept an interface semantics is needed that associates the process and resource objects with each other. Here, each process and resource is equipped with an ExternalInterface and correspondingly connected with each by an InternalLink. The resulting interface class is given in Table 6.13.

Table 6.13: InterfaceClassLibrary AutomationMLEOLInterfaceClassLib - PPR concept part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLPPRConnectorInterfaceClass	AutomationMLEOLInterfaceClassLib/ EOLPPRConnectorInterfaceClass	AutomationMLInterfaceClassLib/ AutomationMLBaseInterface/ PPRConnector

EOLPPRConnectorInterfaceClass is derived from the standard interface class PPRConnector [IEC14]. This has no attributes and no attributes are additionally created.

For the formalized process description approach the hierarchization of processes and resources is enabled. Additionally, within one hierarchy level an order can be expressed. While the hierarchy itself is modeled by the parent-child relation, the order is expressed by equipping each process with an ExternalInterface that is connected by an InternalLink. The created interface class is given in Table 6.14.

Table 6.14: InterfaceClassLibrary AutomationMLEOLInterfaceClassLib - structuring part

role class name	path of role class (structuring)	path of role class (inheritance)
EOLOrderInterfaceClass	AutomationMLEOLInterfaceClassLib/ EOLOrderInterfaceClass	AutomationMLInterfaceClassLib/ AutomationMLBaseInterface/ Order

EOLOrderInterfaceClass is derived from the standard interface class Order [IEC14]. EOLOrderInterfaceClass inherits one attribute from Order - see Table 6.15.

Table 6.15: Attributes of AutomationMLEOLInterfaceClassLib/ EOLOrderInterfaceClass

role class	role class attributes
EOLOrderInterfaceClass	Direction (inherited)

This attribute can have the value: In, Out, or InOut [IEC14].

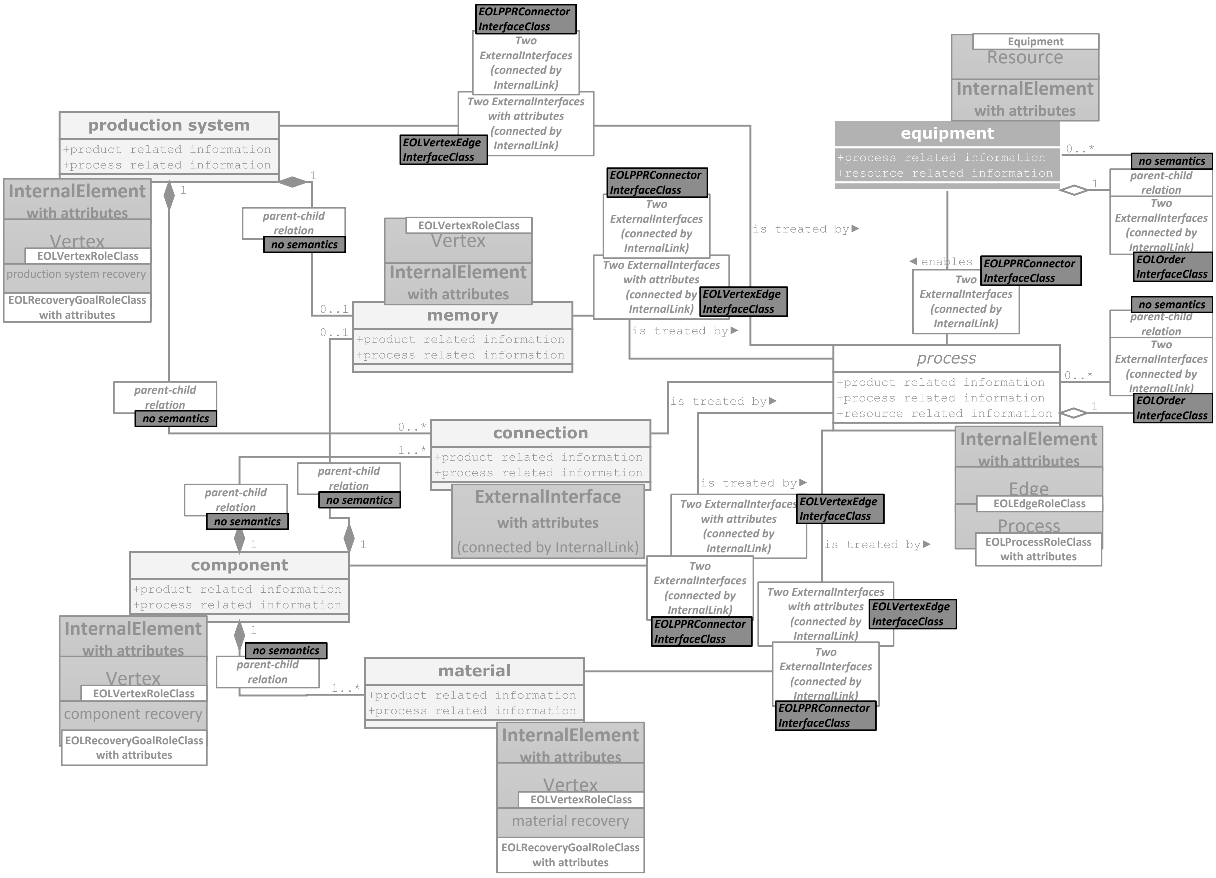

Figure 6.24: End-of-Life process data model (unified and generalized) mapped onto AutomationML regarding role and interface classes

The parent-child relations cannot store additional semantics.

End-of-Life data model

The result of the base EoL data model and the EoL process data model is shown in Figure 6.25. As the EoL strategy can also be considered as a process it can have an ExternalInterface of the interface class ’EOLPPRConnectorInterfaceClass’ that is associated by an InternalLink to the ExternalInterface of a resource.

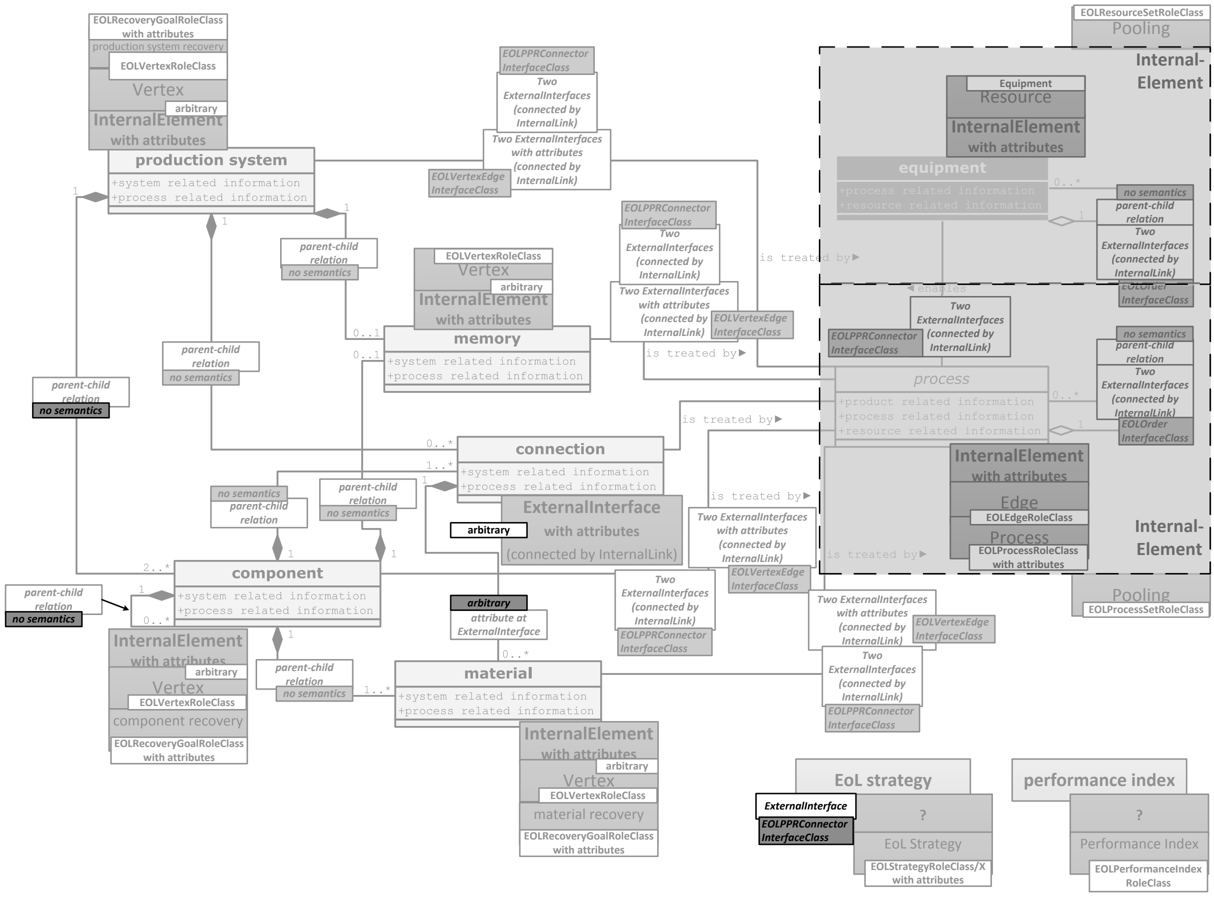

Figure 6.25: End-of-Life data model (unified and generalized) mapped onto AutomationML regarding role and interface classes

The next subsection considers possible system unit classes.

6.2.4 Development of End-of-Life specific system unit classes

System unit classes are describing objects with their technical realization. This includes the definition of the internal structure and concrete interfaces of a specific object class. Attributes usually have values. System unit classes are considered as templates that can be instantiated in the InstanceHierarchy. [DIN10] A prominent example for system unit classes are, e.g. product catalogs.
It is not required to create and use system unit classes. The technical realization can also be modeled right at the InstanceHierarchy. But in this case the reusability is not given. This needs to be decided use case dependently.

When recapitulating the previous subsections, where role and interface classes were defined, and Annex G, where the library concept for the RPM was explained, four different system unit class libraries seem reasonable for the author: library for recovery goals, library for EoL strategies, library for processes, and library for performance indices.

The processes have predefined attributes due to the role classes (see Table 6.4). When one of these role classes is assigned to a system unit class (as a SupportedRoleClass) these attributes can be filled in accordance with the use case.

For the recovery goals an auxiliary InternalElement is required. A recovery goal represents the future requirements on the production system, component, or material. That means that the attributes, which come with the role class ’EOLRecoveryGoalRoleClass’, need to get values. One possible way is to assign ’EOLRecoveryGoalRoleClass’ as ’RoleRequirements’ to the corresponding InternalElement. Only the ’RoleRequirements’ can have attribute values. In the attributes of the ’RoleRequirements’ the value regarding the future application is stored. This could, then, be mapped with a ’MappingObject’ to the attributes of the corresponding InternalElement. Thus, the specific requirements on the InternalElement can be stored. This mechanism supports the workflow in the engineering process regarding getting from a rough description to a detailed one. With these requirements an engineer can, later, choose an appropriate component or product, e.g. from a product catalog that meets exactly these requirements or exceeds them. However, these requirements can be kept, even though, a product is already chosen. In case, the product is not fully functional anymore and needs to be replaced by a new one but it is not available on the market anymore, then the ’RoleRequirements’ can be consulted again to choose a new product that meets the requirements.
But assuming, that the InternalElements of the base EoL data model already have a role class set as ’RoleRequirements’ from the previous life cycle phases, no more ’RoleRequirements’ are possible, only ’SupportedRoleClass’es. But ’SupportedRoleClass’es cannot store own attribute values. They have a ’MappingObject’ like the ’RoleRequirements’. This ’MappingObject’ also maps the attribute names of the role class to the attribute name of the InternalElement. But the attributes of the ’SupportedRoleClass’ have no value. [IEC14] Therefore, an auxiliary InternalElement needs to be created as a child InternalElement of the production system, component, or material InternalElement. Then it is possible again to assign ’EOLRecoveryGoalRoleClass’ as a ’RoleRequirements’. This is a similar approach as the PropertySet concept described in [IEC14]. It makes sense to model this auxiliary InternalElement as a system unit class. Additionally, the SystemUnitClass ’EOLRecoveryGoalRoleClass’ gets the following attributes (see Table6.16) to store the current state of the physical asset in the five performance indices categories. Via a ’MappingObject’ both sets of attributes (System-FutureX and SystemCurrentX) can be associated with each other.

Table 6.16: Attributes of SystemUnitClass EOLRecoveryGoalRoleClass

system unit class	**system unit class attributes**
EOLRecoveryGoalRoleClass	SystemCurrentEconomic, SystemCurrentEcological, SystemCurrentSocietal, SystemCurrentQuality

For the performance indices and EoL strategy library it is also different, because these two libraries are not indented to get instantiated. Both do not provide object semantics themselves. Instead they provide additional information about structuring or grouping of processes (EoL strategy) and attributes (performance indices). Therefore, both were not directly represented in the EoL data model (see Figure 6.25). EoL strategy system unit classes represent a process chain, which is a structure of InternalElements. Performance indices system unit classes on the other hand would provide specific metrics as indicated in Subsection 5.2.9 (out of the scope of this thesis).
However, in order to have the RPM extensible and adaptable it is reasonable to have them decoupled as own libraries.

This last piece finally completes the mapping of the EoL data model onto AML - see Figure 6.26.

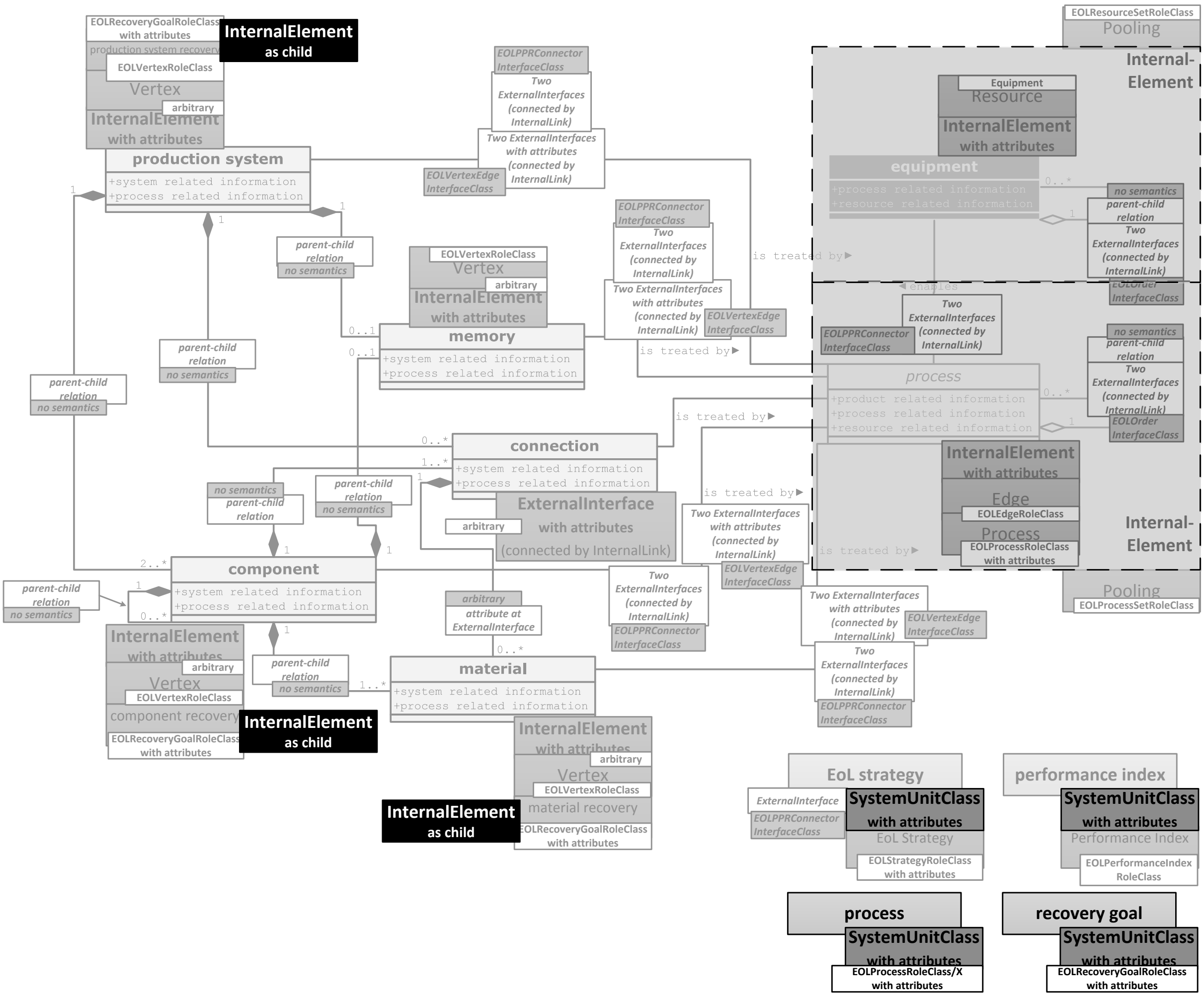

Figure 6.26: End-of-Life data model (unified and generalized) mapped onto AutomationML regarding role, interface, and system unit classes

It is very likely that the InternalElements of the base EoL data model and those for the equipment already have a relation to a system unit class. The class path is stored in ’RefBaseSystemUnitPath’ at the InternalElements. But the referenced system unit classes are coming from the previous life cycle phases of the production system and they are usually not exchanged with the InstanceHierarchy, as the class-instance relation (regarding SystemUnitClass) is just a copy of the class (no inheritance relation), i.e. when a system unit class is instantiated a copy of the class is created. This instance can, then, evolve, over time and independently from the class and vice versa. This enables a customization of the instance. It can be also called copy-class relation. [IEC14] Therefore, the exchange of the system unit class information together with the InstanceHierarchy is neither likely nor necessary.

Libraries are usually only developed and exchanged once and, then, they are set. The real project - the InstanceHierarchy - on the other hand can be exchanged more often. [LS17a]

The complete mapping of all three data models onto AML - at a glance - is provided by Annex K.

Based on the chosen structuring concepts and defined libraries, the EoL data model can be created in the InstanceHierarchy.

6.2.5 Creation of an exemplary End-of-Life data model by using RPM

In the previous subsections several libraries were created. Those are used in this subsection to create the EoL data model for a simple, generic example structure to visualize the concepts and how they are applied, since the case studies from Annex D do not provide specific structures. Figure 6.27 gives an overview of the example structure and all the libraries that were defined. The libraries are represented by the AutomationML Editor V 4.7.0 [Aut]. For the sake of clarity the supported role classes in the system unit classes are hidden except for one system unit class in each system unit class library. An XML representation of all libraries is given in Annex L.
If the use case makes it necessary, the libraries can be further specialized by building up a more detailed inheritance structure.

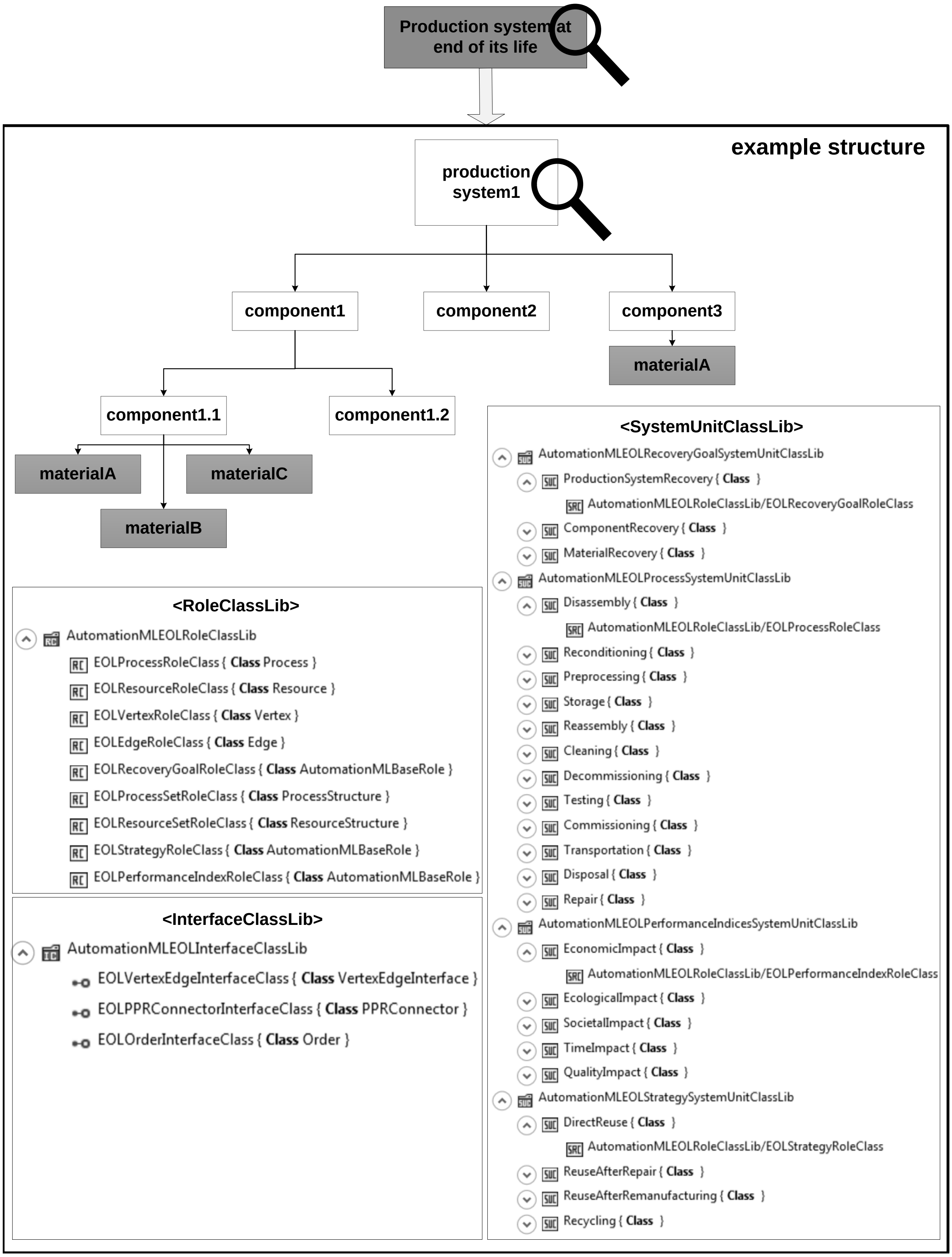

Figure 6.27: Example structure and End-of-Life specific libraries

To now create the EoL data model, the RPM from Chapter 5 is used. By doing this, the corresponding data is stored step by step in the EoL data model and is, thereby, made exchangeable. To create the *.aml file or EoL data model the AutomationML Editor V 4.7.0 [Aut] is used.

The workflow and the corresponding data writing into the EoL data model should be done by a software tool. Developing such a software tool is out of the scope of this thesis. However, this thesis should serve as a basis to develop one.

Step 1: analyzing production system data and identifying recovery goal

The base EoL data model contains the simple example structure from Figure 6.27 with the materials modeled as own InternalElements and with mechanical interfaces modeled as ExternalInterfaces - see Figure 6.28. InternalLinks are indicated as dashed lines.

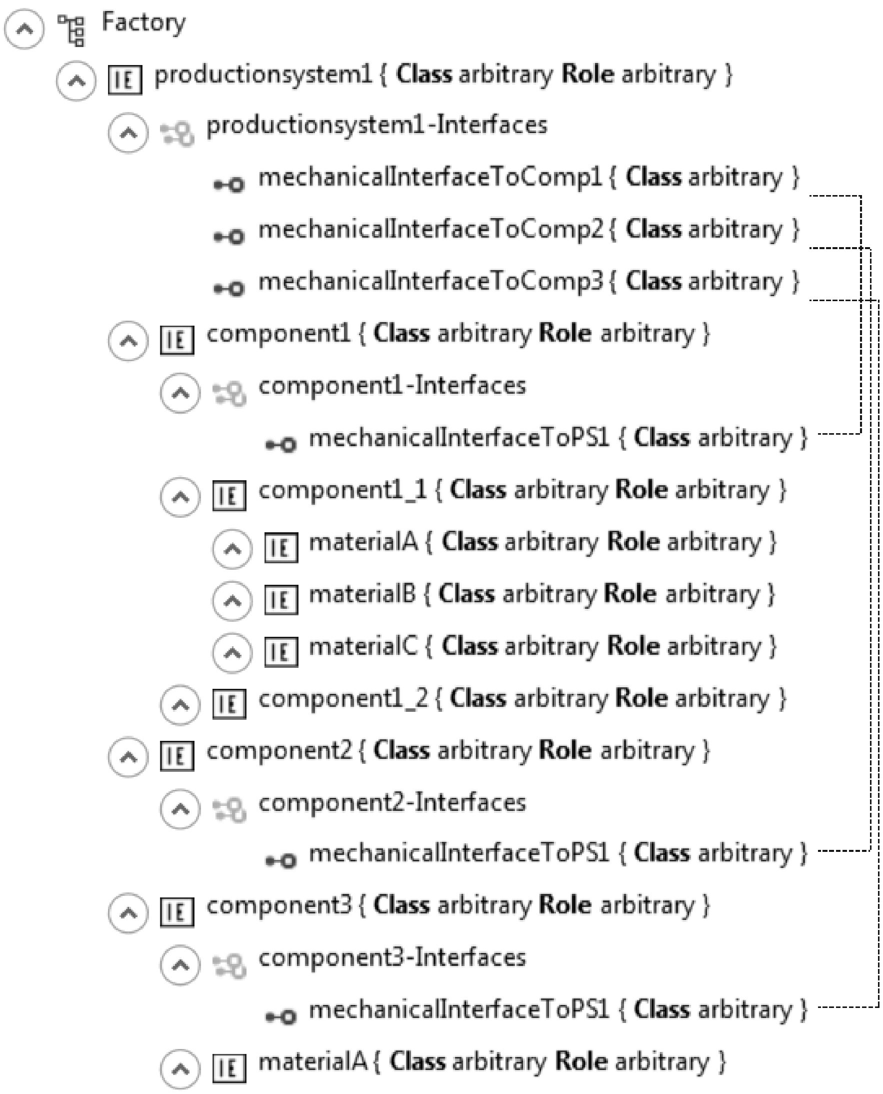

Figure 6.28: Base End-of-Life data model of example structure

The decision maker analyzes the structure of the production system and identifies potential physical assets to be recovered. This step should answer the question: Which physical asset is recovered? For this, at each physical asset of interest an auxiliary InternalElement is created as a child element. In this example 'production system1', 'component2', and 'materialB' is of interest and should be analyzed, which EoL treatment is the best. At each of these three a system unit class 'ProductionSystemRecovery',

’ComponentRecovery’, or ’MaterialRecovery’ is instantiated correspondingly with the role class ’EOLRecoveryGoalRoleClass’ set as ’RoleRequirements’. The auxiliary InternalElement has a set of attributes (namely SystemCurrentX) and the ’RoleRequirements’ has also a set of attributes (namely SystemFutureX). Both are associated with each other by creating five ’AttributeNameMapping’s at the ’MappingObject’, which was created before. Figure 6.29 shows this exemplary for the InternalElement ’materialB’.

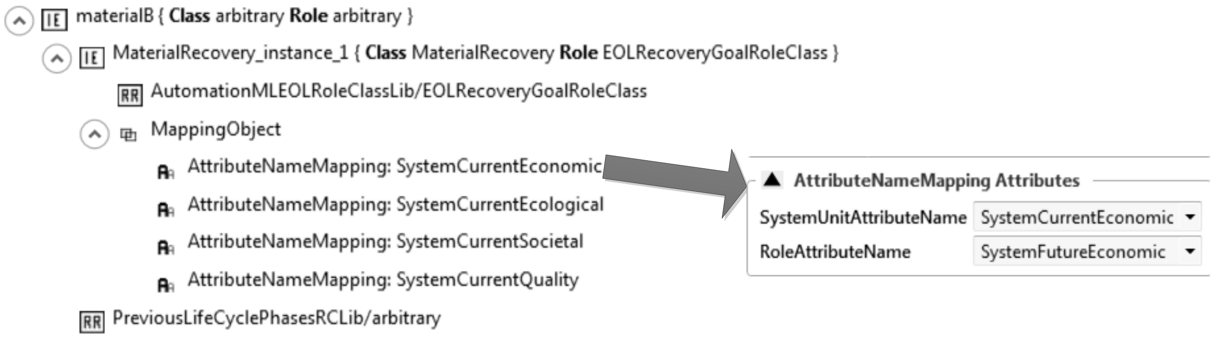

Figure 6.29: Recovery goal as a child element at the physical asset ’materialB’ (extract)

A prerequisite is that the performance indices are known before (in the software tool), so that the attribute can be grouped into the five categories economic, ecological, societal, time, and quality. Therefore, they are stored as system unit classes in a system unit class library - see Figure 6.27.

Step 2: characterizing recovery goal

This step should answer the question: How does the future application of the physical asset to be recovered look like? For this, each attribute of the ’RoleRequirements’ gets a value according to the future application of the physical asset, e.g. for ’materialB’ (see Figure 6.30).

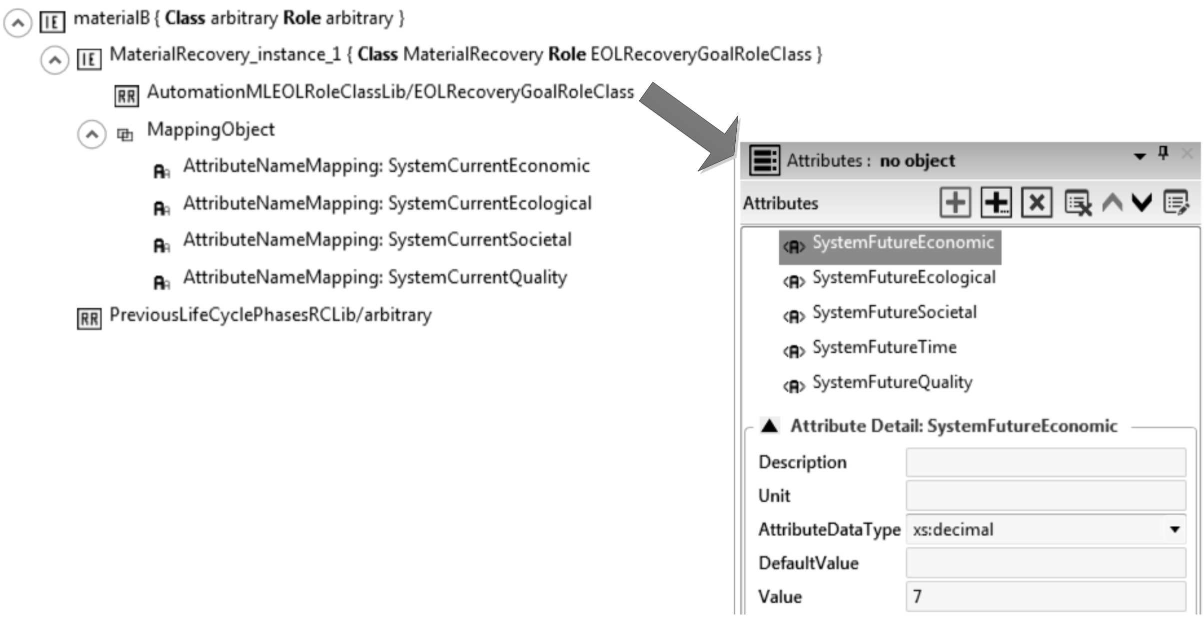

Figure 6.30: Characterizing recovery goal ’materialB’ (excerpt)

Step 3: comparing system with recovery goal

This step should answer the question: Is the actual physical asset capable to fulfill the future application with its requirements? For this, each attribute of the auxiliary InternalElement gets a value according to the current state of the physical asset, e.g. for 'materialB' (see Figure 6.31).

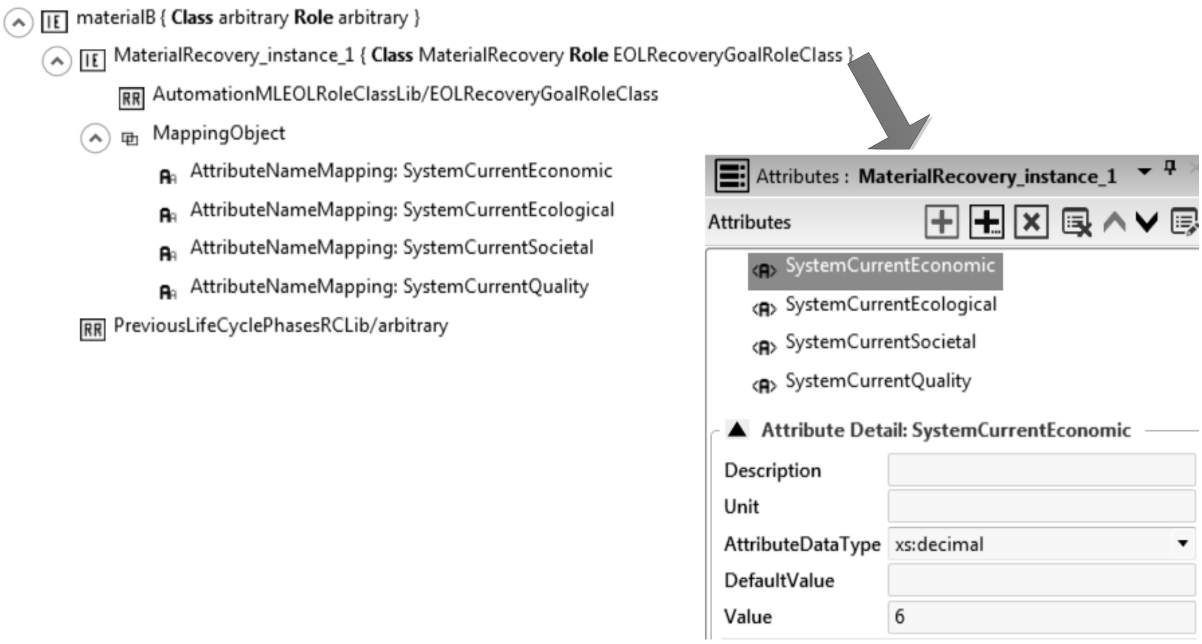

Figure 6.31: Characterizing current state of 'materialB' (excerpt)

The current state, however, should be determined from the attributes of the physical assets that came with the base EoL data model.
The coding of the decision tree, i.e. the rules to make a decision on an appropriate EoL strategy, needs to be done in the software tool. This is, therefore, tool functionality and not a subject to exchange.

Step 4: choosing EoL strategy

This step should answer the question: By which EoL strategy is the actual physical asset capable to fulfill the future application with its requirements? After the software tool has done the assessment and calculation on a proper EoL strategy, the decision maker chooses one. The following Figure 6.32 provides the modeling of one of the recovery goals: materialB. This is intended to get recovered by the EoL strategy recycling. This equals to EoL1 and correlates to the case studies CS **II** and CS **IIIb**.

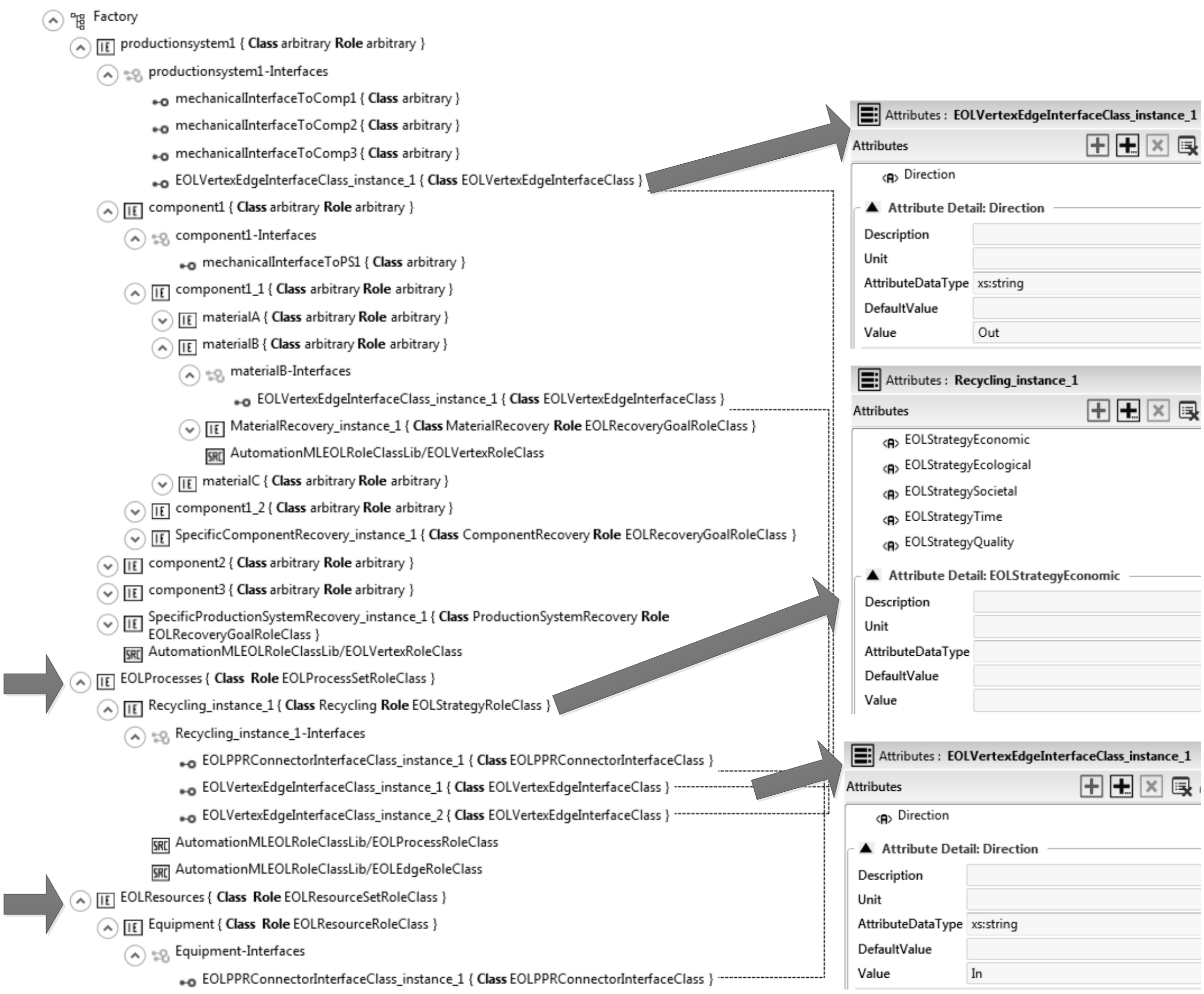

Figure 6.32: Storing End-of-Life strategies

Here, the pooling InternalElements 'EOLProcesses' and 'EOLResources' are created on the same level as the 'production system1'. The role classes 'EOLProcessSetRoleClass' and 'EOLResourceSetRoleClass' are assigned to them. Within the first one, the system unit class 'Recycling' is instantiated, which comes with a set of attributes 'EOLStrategyX' in the five categories of the performance indices (initially empty). The InternalElement 'Recycling instance 1' the following additional role classes as 'SupportedRoleClass': EOLProcessRoleClass and EOLEdgeRoleClass. The InternalElement 'Equipment' has the 'EOLResourceRoleClass' assigned to it. The InternalElement 'Recycling instance 1' and 'Equipment' have both an ExternalInterface 'EOLPPRConnectorInterfaceClass', which are connected by an InternalLink according to the PPR concept. The InternalElement 'production system1' and 'materialB' have additionally the role class 'EOLVertexRoleClass' assigned to them as well as each an ExternalInterface of 'EOLVertexEdgeInterfaceClass' with the attribute 'Direction' filled out correspondingly. On the other hand the InternalElement 'Recycling instance 1' has according to the graph concept two more ExternalInterfaces of 'EOLVertexEdgeInterfaceClass'. ExternalInterfaces of 'EOLVertexEdgeInterfaceClass' are connected by InternalLinks as indicated in Figure 6.32.

Basically it is a layer on top of the base EoL data model as shown in Figure 6.33.

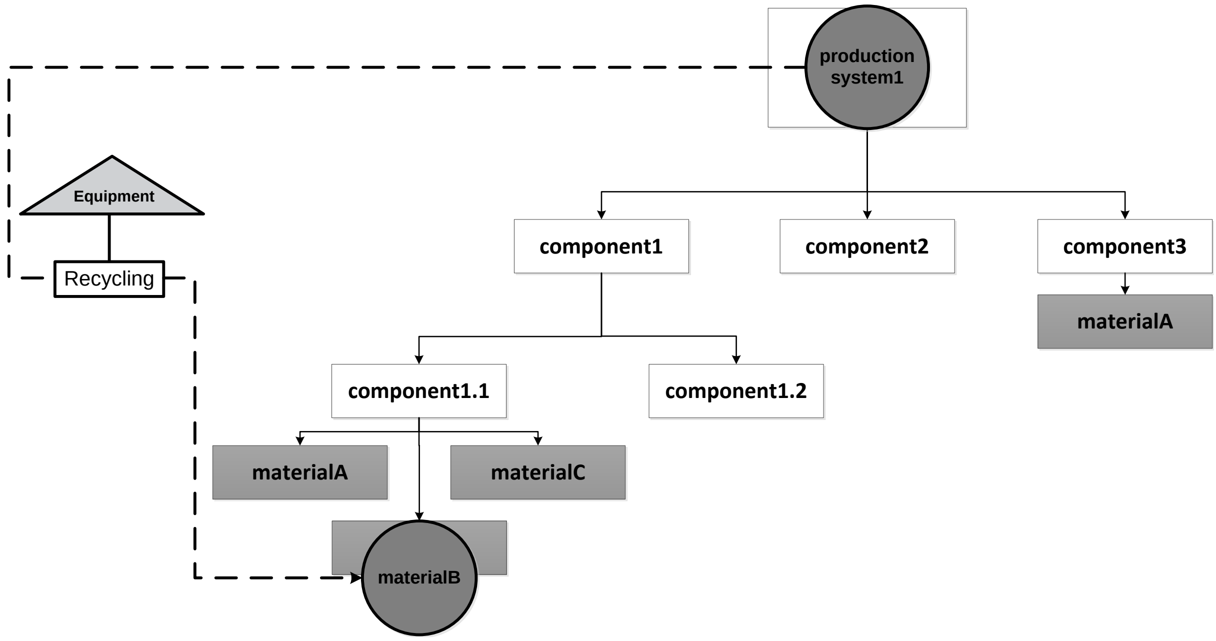

Figure 6.33: Storing End-of-Life strategy for 'materialB' in the simple example structure (dashed line)

By that, the storage of alternatives is also possible, i.e. more than one EoL strategy. They are simply considered as additional layers on top of the base EoL data model. However, in the end the EoL data model should only store exactly one EoL scenario (with one EoL strategy) after the software tool has determined an appropriate one. This rough description is, then, passed to the subsequent expert in the recovery process, who details this rough description.

In case that a process would need to be directly assigned to a connection, e.g. a disassembly process should be directly assigned to a mechanical interface, then the 'EOLVertexEdgeInterface' at the product InternalElement would be connected with its corresponding mechanical interface ExternalInterface via an InternalLink.
However, in this simple example, the disassembly could be done in a destructive manner without taking care of the connections the production system have, as indicated in Figure 6.33.

Step 5: creating process chain and characterizing processes

This step should answer the question: What is the process chain for the EoL strategy? Figure 6.34 shows the extensions made below the EoL strategy 'Recycling'.

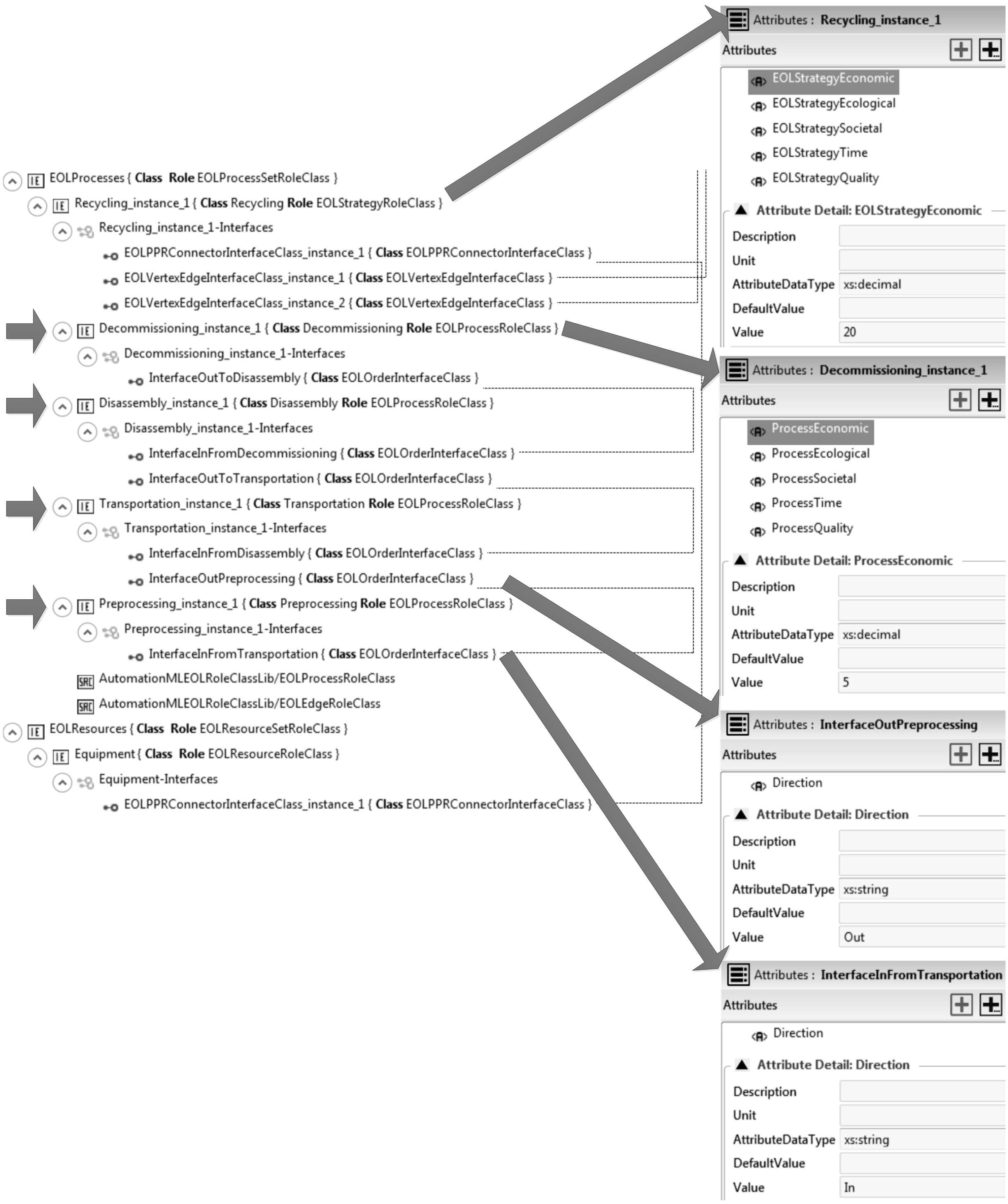

Figure 6.34: Process chain of the End-of-Life strategy 'Recycling' (excerpt)

Here, the decomposition of the EoL strategy 'Recycling' into the single processes is done. For this, the processes from the AutomationMLEOLProcessSystemUnitClassLib (see Figure 6.27) are instantiated as child elements at the InternalElement 'Recycling instance 1' - in accordance with the proposed process chain from Figure 5.20. Those four processes are coming from the AutomationMLEOLProcessSystemUnitClassLib with a set of attributes each (namely ProcessX following the categories of the performance indices) including values. These values in each category can be, then, summed up and store in the attribute set of the EoL strategy InternalElement (here: Recycling

instance 1). To represent the internal order of the processes in the EoL strategy, each process is equipped with ExternalInterfaces of 'EOLOrderInterfaceClass': One for the direction 'In' and one for 'Out'. The first and the last process on the other hand have only one ExternalInterface, because they represent the beginning or the end of the process sequence. These ExternalInterfaces are, then, connected by InternalLinks correspondingly. This is also indicated in Figure 6.34.
This is in line with the formalized process description approach.

In case, these processes should be used for the graph concept, which would be a detailing, they would be also equipped with ExternalInterface for the PPR concept and the graph concept, like the EoL strategy 'Recycling'.

Step 6: determining and weighting indices

This step should answer the question: How is the physical asset with its EoL strategy characterized? How is the EoL scenario characterized? For this, the values for the performance indices of the recovery goal and the EoL strategy are collected - see Figure 6.35.

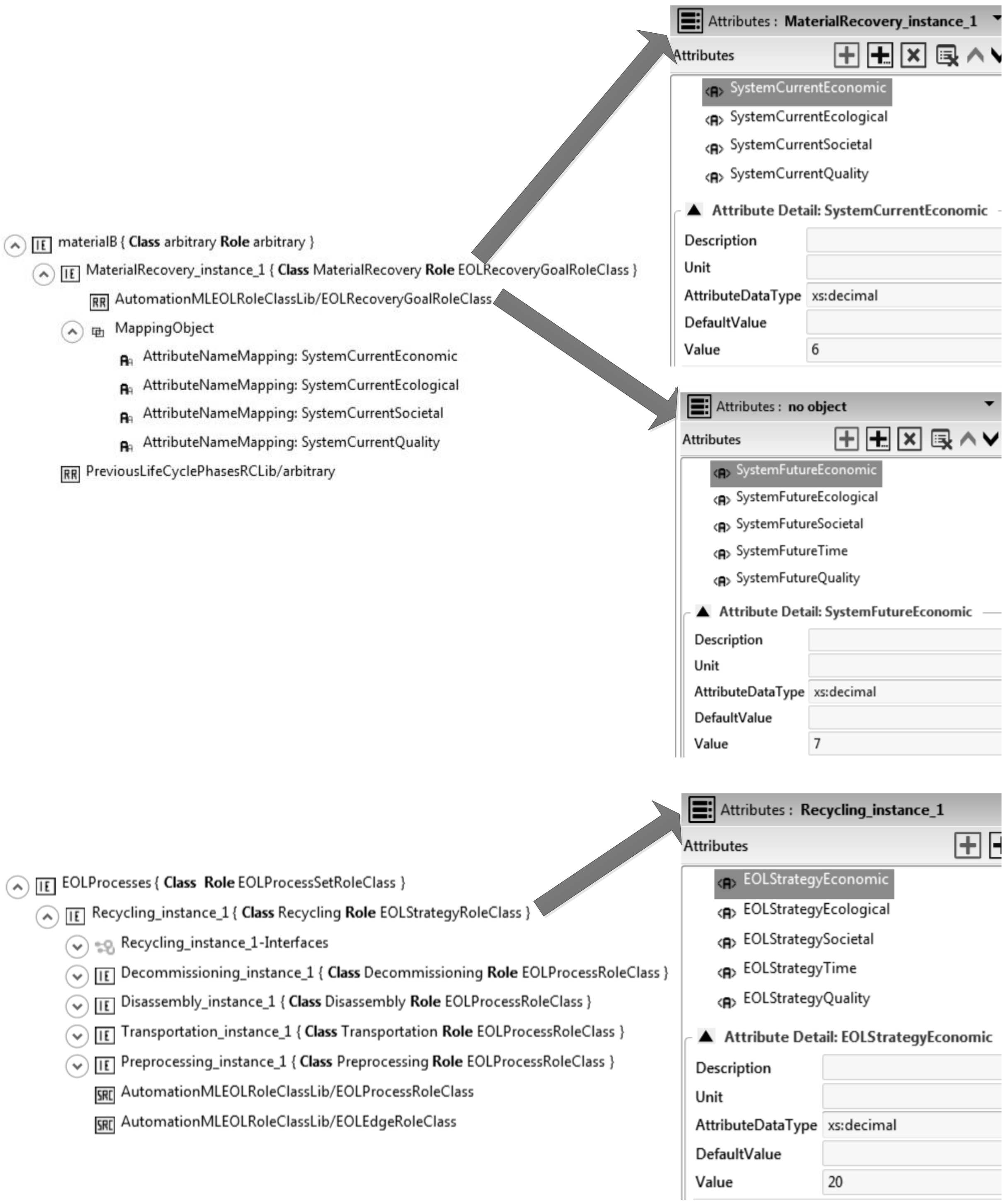

Figure 6.35: Determining performance indices for the recovery of 'materialB' (excerpt)

In this figure the fictive attributes values can be summed up now for the economic impact:
SystemCurrentEconomic + EOLStrategyEconomic = 26 with SystemFutureEconomic = 7, i.e. 26 > 7.
This needs to be calculated in each of these categories.

A weighting in the software tool is possible, e.g. in accordance with the corporate strategy. This could be stored as a value of the attribute like this: 2*6. But this would require an adaptation of the 'AttributeDataType', e.g. into xs:string.

Step 7: presenting characterized End-of-Life scenarios and choosing one

In this step the software tool would present the characterization of the EoL scenarios, so that the decision maker can make a choice about an appropriate one. When the decision is made, the EoL scenario is stored in the EoL data model or *.aml file and is passed to the next experts, who uses this as a basis to do the detailed planning in the project planning phase.

An XML representation of the simple example structure developed in this subsection is given in Annex M, including all necessary libraries.

6.3 Conclusion

This chapter has shown on the one hand how the EoL data model can be mapped onto AML and on the other hand - by using a simple example structure - how the Recovery Planning Method (RPM) can be applied to fill the data into the EoL data model. This model can be stored centrally or distributed. Both approaches are supported by AML. The RPM is designed extensible, so that additions are possible (e.g. of processes or performance indices), but also specializations (e.g. concretization of processes or recovery goals). This is also supported by the AML and how the EoL data model is modeled in AML.
Also the creation of new objects (i.e. InternalElements) is supported. The base EoL data model may be extended when through a certain process (e.g. disassembly process) an object is separated into subobjects but those are not yet represented in the base EoL data model. Especially when it comes to material recovery also destructive disassembly techniques can be applied, which do not need to care about existing mechanical interfaces and undoing them.

The EoL data model basically needs to store the EoL scenario, which is chosen for the recovery of a certain production system. The RPM is explicitly addressing this. As mentioned in Subsection 5.2.9, this EoL scenario represents a first draft or rough description of the recovery process of a production system. Due to the similarity of the recovery planning process to the engineering process, it is reasonable to make this first draft solution exchangeable. The EoL scenario is the result of the project preparation phase of the recovery planning process (see Figure 4.6). When exchanging this, the experts in the subsequent project planning phase can use this as a basis for a detailed planning - project planning.
Even though, it is the first draft of the recovery process of a production system, the data model or model in AML can become complex. The reason is the the combination of the different concepts and the complexity of the production system itself. Those concepts can be basically considered as different networks that are interconnected with each other. For the management of this structural complexity [LMB09] or [Win13] can be consulted. Based on such management approaches for structural complexity algorithms for clustering or partitioning could be applied for analysis purposes, as done by [Kle14]

for identifying reusable components of production systems or by [Sos00] and [PE94] - going one step further - for putting together development teams in accordance with the architecture of the product to be developed by aligning the product and team interfaces.

Also one result of the RPM and the EoL data model is that the recovered physical assets are described in AML. When the experts keep using these AML representations within the entire recovery planning process, they can finally deliver not just the recovered physical asset at the end of the project or the EoL phase but also its digital representation. As shown in Figure 2.5 there are a lot of recovery possibilities starting from the EoL phase. Following these arrows, several phases of the life cycle of a production system can benefit from that.
When looking at the engineering phase of a production system, experts are using product catalogs, e.g. from component manufacturers, to find concrete components that fit the production system's needs. Some component manufacturers start to provide their product catalogs virtually. When the experts order a component, they get the virtual representation of the just ordered component as well, which is in line the Industrie 4.0 paradigm [WGE+17]. Then, the expert can integrate the virtual model into the production system model. For example, the component manufacturer Festo AG & Co. KG is about providing the virtual models of their product catalog as an *.aml file [Fes]. Usually, when components from a product catalog are ordered, newly produced components will be delivered. But when thinking of the recovered physical assets with their digital representation, also those could be gathered in a product catalog (of used components) to provide the same look and feel for the experts that are engineering a production system. To keep the integrity of the UUID of each recovered physical asset, an approach - similar to this of GS1 Germany GmbH [GS117] - could be chosen. A central organization could create and distribute UUIDs but it also stores them and, by that, the UUIDs with their associated assets are made traceable and unambiguously identifiable worldwide.
A disadvantage of having a product catalog of recovered physical assets would be that these are already existent and need to be stored until an expert orders or uses it. This is in contrast to the ongoing trend to have a production with lot size one while minimizing the storage space. However, when it comes to a circular economy and to treat natural resources responsibly and sustainable, a rethinking of current trends and established practices may be necessary.

7 Summary and outlook

After motivating the inevitable developments emerging from the Industrie 4.0 paradigm, which comes along with the digitalization of the industry, in Chapter 1 the author chose to approach this paradigm by focusing on the Industrie 4.0 use case 'Circular Economy'. This has the scope to keep machines, products, components, and materials in cycles to preserve their inherent value. By this, energy and resources can be saved and waste and CO2 emissions can be reduced to meet global climate policy goals, which can turn today's continuous economy into a circular one.
Because of the lack of appropriate publications on keeping production systems of the manufacturing industry with their components and materials into cycles, the author chose this as the topic of this thesis.

These cycles start in the last phase of the production system life cycle - the EoL phase. To understand the role of this phase and its immense interconnections with the other life cycle phases regarding information and physical assets, Chapter 2 gave a general overview of the life cycle of production systems, but also on possible cycles or recovery possibilities in it.
After this, Chapter 3 is entirely dedicated to the EoL phase of production systems and reflects the literature based state of the art. Because of the already mentioned lack of appropriate publications on this topic, the author chose the inductive scientific approach to face this topic. In doing so, publications from similar or close fields of work were analyzed and adapted to the scope of this thesis.

Encouraged by this lack of publications the author conducted seven case studies in Chapter 4, in which experts were interviewed, whose daily business it is to close the cycle, i.e. to recover a production system, its components or materials. By this, the author could reveal that the recovery of production systems is actually done and that there is a huge business ecosystem with many stakeholders existing, which would benefit from a digitalization in the context of Industrie 4.0.

After having analyzed the literature based state of the art (regarding available methods to decide on how to close the cycle or recover production systems as well as on required information to do so) as well as after having analyzed the case studies (also regarding repetitive activities possibly avoidable by digitalization means) Chapter 5 proposed a method, with which a decision-making process could be supported. This method called recovery planning method (RPM) intends to reduce the efforts of the production system recovery or more precisely the efforts of the activities within the project preparation of the production system recovery process - or make them even obsolete.
But this method needs information to be applied but it also creates information that is describing the chosen cycle. This is subsumed in an EoL data model.

Since the mentioned description of the chosen cycle, which is the result of the RPM, represents just an early artifact in the beginning of the recovery process, data consistency was identified as important by the author. Information needs to be provided somehow

to the EoL phase on the one hand and the RPM on the other hand needs information to be handled within the EoL phase or within the recovery process itself, since the recovery process of production systems is comparable with the engineering process as the author found out. The recovery process is similar in complexity (regarding the activities) and heterogeneity (regarding the involved experts and software tools) to the engineering process.
Based on this finding, the author proposed to exchange the necessary information or the EoL data model by the data exchange format AutomationML (AML), which is known from its application with the engineering phase of production systems. Therefore, a mapping of the EoL data model onto AML was developed in Chapter 6 and how the result of the RPM is expressed by AML to make this artifact exchangeable between the different experts and software tools involved in the recovery process of production systems to create based on it the final and executable recovery process description; But also to make it exchangeable between the different stakeholders within the business ecosystem. Thereby, the requirement on closing cycles emerging from sustainable economy can be met.

As indicated, the scope of this thesis is the recovery of production system from the manufacturing industry. Only publications from the discrete manufacturing industry or its products were read as well as only experts from this industry were interviewed. The author does not exclude that the results could somehow correlate with other industries, like process industry or energy generation, but these usually come with different regulations and laws, which were not considered explicitly in this thesis.

In the next sections the results and findings of this thesis will be critically discussed, assessed, and put into relation to the research questions.

7.1 Answering the research questions

In Chapter 1 three research questions were given, which emerged from the motivation of this thesis and the intention to bring light into the EoL phase of production systems. These are answered in the following and correlate with the main findings or results of this thesis.

RQ1: How is the EoL phase of production systems characterized?

Due to internal and external reasons a production system can lose its purpose or its function, which means that it is no longer needed at a certain geographically location. The operation phase ends and the production stops. At this point in time, the production system enters the EoL phase, where it needs to be decided about a proper recovery option or EoL treatment.
The responsibility for the production system, which technically had turned into waste, has the production system owner that needs to take care of the legally proper treatment. The author has not found any specific directives, laws, or guidelines that need to be met when it comes to finding a proper recovery option - except for the KrWG law that basically deals with the support of a circular economy to save natural resources and to ensure the protection of humans and environment when producing and managing waste. That means avoiding waste before recycling waste before disposing waste. But

when the recovery process is planned in detail and is finally executed, a lot of laws need to be considered, e.g. VerpackV or ArbSchG.
With the decision on a recovery option or EoL treatment for the production system, the recovery process starts. The recovery process is comparable with the engineering process of production systems and it is structured like a project including a project preparation, project planning, project execution, and project completion phase, each with certain activities. Like the engineering process, also the recovery process has different experts and software tools involved to plan and execute the recovery process. Data consistency plays an important role. But in contrast to the engineering phase, the recovery process is highly dependent on information that needs to be provided by the prior life cycle phase namely the engineering phase and the operation phase. After having the production system, its components, or materials recovered the resulting or recovery physical assets can be used again in the production system life cycle or in other life cycles. Besides the physical asset itself also its (describing) information is recovered, so that this physical asset can be easily consumed or used again in other life cycle phases.
Behind the EoL phase there is a huge business ecosystem with many stakeholders, which can be involved in the recovery process of production systems.

RQ2: How can the decision making process be supported, so that a proper EoL treatment for a production system can be found and assessed?

To answer the question what a proper EoL treatment for a production system could be, it needs to be decomposed into the questions 'What?' and 'How?'. That means a characterization of the physical asset to be recovered is required as well as the characterization of the process that describes, how the physical asset is recovered.
The 'What?' is called the recovery goal and it can be the physical assets production system (EoL3), components (EoL2), or materials (EoL1). The recovery goal needs to be characterized regarding its future application, i.e. the requirements of its future use. And this needs to be compared with the current state of the physical asset. Through this comparison it is possible to choose a process called EoL strategy. The possible EoL strategies are: direct reuse (EoLxa), reuse after repair (EoLxb), reuse after remanufacturing (EoLxc), and recycling (EoL1). If the recovery goal cannot be reached by an EoL strategy (taking the current state into account) the physical asset needs to be disposed and the inherent value is lost. Another option would be to adapt the recovery goal and find another and new purpose for the physical asset. Recovery goal and EoL strategy is called EoL scenario.
To make EoL scenarios comparable they need to be characterized in certain categories, which can be, then, compared. In this thesis, the following categories were identified: societal impact, ecological impact, economic impact, quality impact, and time impact. These are called performance indices. By characterizing the recovery goal as well as the EoL strategy regarding those five performance indices, in each index the numbers can be summed up and weighted, if needed. This makes the EoL scenarios comparable. Based on this characterization and these performance indices, KPI based strategic planning methods could be used, additionally.
This procedure is described by the recovery planning method (RPM) that was developed in this thesis.

As mentioned before, the recovery process starts with this decision-making process on a proper EoL treatment of EoL scenario for a production system. Once the decision is made this EoL scenario can be planned in detail and can be executed. This decision-making process takes place in the project preparation phase of the recovery process. This phase comprises the following activities: inspection of physical asset, examination of documents, estimation of process costs, assessment of value of physical asset, and the collection of requirements. The last three activities are intended to be supported by the developed RPM. But to apply this method or to integrate it into software tools information is needed, which is represented by the first two activities 'inspection of physical asset' and 'examination of documents'.

RQ3: Which information is required to enable or support this decision making process and how can it be made available?

The first two activities 'inspection of physical asset' and 'examination of documents' of the project preparation phase of the recovery process of production systems represent the required information from the prior life cycle phases namely the engineering phase and the operation phase. This information is stored in the base EoL data model. It has the five classes *production system*, *component*, *material*, *connection*, and *memory*. Each of them has as properties the four types of information: type (e.g. function), technical characteristics (e.g. weight, size), instance information (e.g. ID, location designation), and measured data (e.g. deterioration state, condition). It represents the product related information, because in the recovery process the production system is considered as the product, as it is the intended subject of recovery.
Besides this information from the prior phases of the production system life cycle, additional information is needed, which needs to be gathered during the EoL phase or the recovery process. This information is stored in the EoL process data model. It also has the five classes from the base EoL data model, but additionally one *process* class and one *equipment* class. This correlates with a PPR concept (product-process-resource concept). The *equipment* class has as properties the two information types: process related information (e.g. cost, revenue) and resource related information (e.g. skill, capacity). The *process* class has as properties the three information types: product related information (e.g. problematic asset), process related information (e.g. instruction, difficulty), and resource related information (e.g. cost, time). The five product related classes, which correlate with those from the base EoL data model, have as properties the two information types: product related information (e.g. market value, regulation) and process related information (e.g. harmfulness, order).
In the end, both data models are aggregated to one model - the EoL data model. This comprises the information, which is needed from the prior life cycle phases, the information from the EoL phase itself, but it also represents the information necessary to describe the result of the RPM - an early artifact in the beginning of the recovery process.
It was shown that AML can be used to store the relevant information or the EoL data model.

In the next section the results of this thesis are critically analyzed and discussed.

7.2 Assessment of the results

The author thinks that the EoL phase with its stakeholders can benefit from the digitalization demanded by the Industrie 4.0 paradigm, because this phase is highly dependent on information from the engineering and operation phase. An increase of the quality or the availability of this information can improve the quality of the results of the EoL phase or recovery process - through easier information acquisition. But an easier information acquisition in the EoL phase comes along with a proper information documentation and information maintenance in the engineering and operation phase. But why should the production system owner and operator spend money on that and increase the total cost of ownership (TCO), when the production system is sold at the end of the operation phase anyway? Why spending money when only other stakeholders would benefit from that? Figure 7.1 visualizes a today's documentation process with an idealized one.

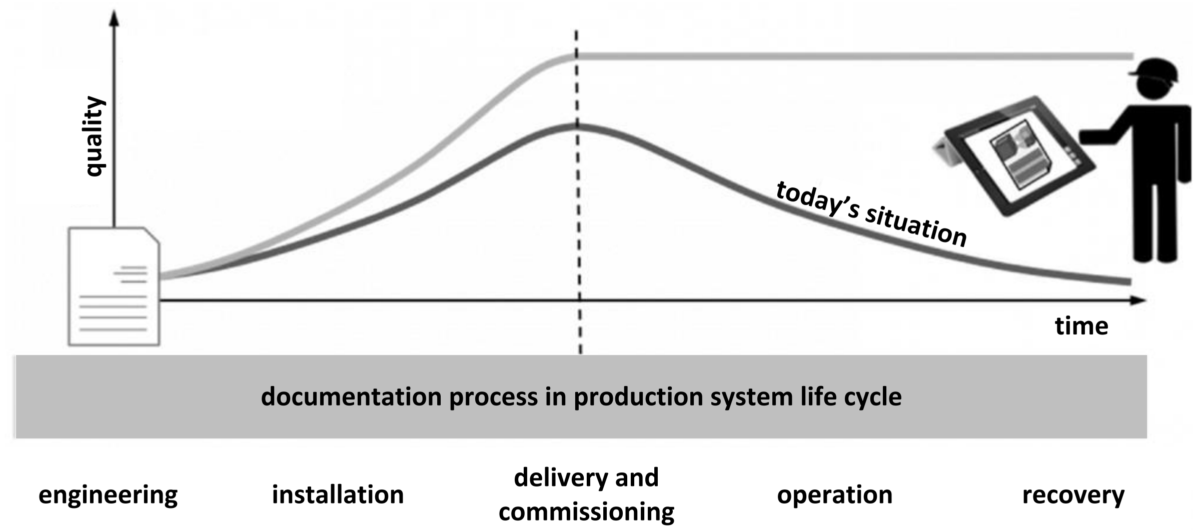

Figure 7.1: Documentation process in the production system life cycle (based on [MKD17])

The answer could be that these costs for information documentation and information maintenance can be directly considered in the negotiations between the production system owner and the service provider that intends to buy the production system for recovery purposes. Because a well-documented production system is more valuable for the service provider since recovery costs and revenues achieved by the recovered physical assets can be calculated more accurate. But the same holds also for the production system owner better knowing the value of the production system. By that, the quality of the negotiations and, in the end, of the offer could be increased, which would reduce the risk on both sides. Because the creation of an offer for an already existing production system that is intended to be further used, extended, modernized, or somehow manipulated always comes along with a high risk [Tit16], because both sides cannot estimate or determine the residual value and functionality of the production system reliably. When the offer creation process could also be shortened by that, it would be beneficial for the service provider, as this process is usually not paid when the production system owner rejects the offer [Tit16].

For legacy systems it is not likely that the digital representation of an already operating production system is created afterward, in that way that the EoL phase could have a benefit from that. The costs for acquiring information need to be considered [Syl17], which would be likely too high for legacy systems due to the complexity of production systems. But according to [Hub14] on the other hand, also the value of an information

needs to be considered. The value (for having and using a certain information for a decision) is calculated by the costs in case of having made a wrong decision multiplied by the probability that the decision is made wrongly.
However, it might be possible that in the course of Industrie 4.0 sensors with smart software will be developed that capture the current state of the production system and via "reverse engineering" the digital representation can be generated (semi-) automatically. Once the technologies are mature enough, business cases will occur. [MKD17], for example, has presented an approach to enable a smart technical documentation - also by using AML with OPC UA.
For new production systems the author thinks that it is reasonable to put the creation and maintenance of the digital representation of the production system on the agenda right from the beginning. Since the EoL phase is not the only phase that can benefit from a digital representation, also the operation phase (e.g. for condition based maintenance e.g. [HJKB16]), and since the information of the base EoL data model is not exclusively usable in the EoL phase, production system owners will have created and maintained the digital representation in order to facilitate other business cases. And the EoL phase could be supported by that at the same time.

All in all, this thesis shall encourage the production system owner and/or operator but also the other stakeholders (also like government agencies and regulatory bodies) to start thinking about the EoL phase and get a life cycle view on the production system. They should provide or demand the EoL relevant data today, so that the information is available tomorrow and new business cases get the chance to occur as well as future software tools.

The RPM, developed in this thesis, could be a basis for such a future software tool. The RPM realized as a tool can lead a user step by step through the decision-making process about finding a proper EoL scenario for a certain production system. But a challenge of this tool would be to appropriately visualize and guide the user through this information intensive method [LE13].
Also, the RPM is not applicable out of the box for a certain use case or production system. Due to the inductive scientific approach, which has led to a generalization of several specific methods, the RPM should be rather seen as a guideline. It can guide experts to get a systematic and holistic view on the use case and production system. A use case specific adaption is required in terms of finding appropriate performance indices and metrics, implementable EoL strategies, desired recovery goals etc.
After having interviewed some of the experts again and after having presented them the developed RPM, it can be concluded that this method is generally considered as consistent and conclusive. However, open questions remain regarding how applying the RPM to a specific use case or production system, which is in line with the limitations mentioned by the author, mainly resulting from the generalization and, thus, out of the scope of this thesis. But these open questions provide a good overview of future research work. The questions are: Who acquires all the information; Is it worth the effort? How is the information acquired; Who provides this; Are these exact numbers or based on statistics? Is it necessary that every physical asset of the production system can and need to communicate to make the RPM work? Who decides, which recovery goals are considered? Is it possible to find generally valid criteria for characterizing the physical assets according to the performance indices; Is it actually not only money-driven? The

experts also pointed out that the RPM is not applicable at the moment due to information lack. And some information won't be available even in the future, because some physical assets will remain 'dump' or non-communicative. An automatically generated base EoL data model of the physical asset is considered as a prerequisite due to information acquisition costs. The EoL data model needs to be dynamic, not static, since spare parts and prices are changing frequently.
These limitations may be a good starting point for other theses or scientific work.

The Industrie 4.0 paradigm and the today's approaches towards a circular economy go hand in hand. Even new business ecosystems could emerge. Thinking of, when a customer does not want to possess the physical asset anymore (e.g. a car or a production system). Instead the customer wants to buy only the service a physical asset can offer (e.g. mobility or a car production of x cars a day). Then, the physical asset owner stays in control of the asset throughout the entire life cycle and can take it back, when the service of the physical asset is no longer requested. But when it is taken back, it is imaginable that the physical asset owner got the physical asset back also with all the EoL relevant information, because the physical asset owner would be in charge of taking care of creating and maintaining the digital representation. [Pla16a]
A life cycle thinking, holistic consideration or optimization is supported at best, when one stakeholder possesses the physical asset throughout its life cycle. Otherwise it can only come down to an optimization of one or several life cycle phases and the valuable interconnections between the life cycle phases of a production system are not used adequately or at all, which otherwise would support the life cycle reduction.

When having all EoL relevant information at hand a proper recovery and reuse of the physical asset is possible. According to the bathtub curve (Weibull distribution) the failure rate of a production system is higher at the end (e.g. because of wear and tear), but also at the beginning (e.g. because of construction mistakes) [LK03] [Her10]. When knowing of the remaining useful life, a recovered physical asset might be preferable to a new one [COR16]. And when it even comes with the digital representation a second, third, or fourth usage or reintegration of the physical asset in another life cycle is eased. Since the introduction of new physical assets or technologies in a company usually come with a time consuming release or approval process, a recovery of them might be worth thinking about it.

Even though, this thesis is basically focusing on the recovery of a physical asset including its digital representation, there is another non negligible part: the software. That means know-how is not only found in the construction, the mechanical part, or the electrical part of a physical asset. It is also steadily put know-how into the software of physical assets. [Sen09] But how can software be recovered properly? And what if when the production system is decomposed into smaller components to be recovered, how does this affect the software of the production system? Can the software also be decomposed in the same way? Can it be cut on the same connections? Does the software architecture need to correlate with the physical architecture of the physical asset, so that it can be cut on the same connections? This is only rudimentarily addressed in this thesis by the existence of the UML class *memory* in the EoL data model. But these questions remain unanswered and could be a starting point for future research.

Bibliography

[ABR07] M. S. Abu Bakar and S. Rahimifard. Computer-aided recycling process planning for end-of-life electrical and electronic equipment. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(8):1369–1374, 2007.

[ABR08] M. S. Abu Bakar and S. Rahimifard. An integrated framework for planning of recycling activities in electrical and electronic sector. *International Journal of Computer Integrated Manufacturing*, 21(6):617–630, 2008.

[AG12] R. Attri and S. Grover. A comparison of production system life cycle models. *Frontiers of Mechanical Engineering*, 7(3):305–311, 2012.

[Aki11] Ö. Akin. *Embedded Commissioning of Building Systems*. Artech House Publishers, 2011.

[AMR13] F. Ardente, F. Mathieux, and M. Recchioni. Combining Five Criteria to Identify Relevant Products Measures for Resource Efficiency of an Energy Using Product. In A. Y. C. Nee, B. Song, and S.-K. Ong, editors, *Re-engineering manufacturing for sustainability*, pages 111–116. Springer Singapore, 2013.

[AS15] W. Albrecht and C. Schwitalla. Rückbau, Recycling und Verwertung von WDVS. Möglichkeiten der Wiederverwertung von Bestandteilen des WDVS nach dessen Rückbau durch Zuführung in den Produktionskreislauf der Dämmstoffe bzw. Downcycling in die Produktion minderwertiger Güter bis hin zur energetischen Verwertung. Technical report, Forschungsinitiative Zukunft Bau, Fraunhofer IRB Verlag, 2015.

[Ass17] AssetOrb GmbH. `https://assetorb.com/de`, 2017. Online; accessed 15-September-2017.

[Aut] AutomationML e.V., editor. AutomationML Editor V4.7.0. `https://www.automationml.org/o.red.c/dateien.html?cat=1`. Online; accessed 14-October-2017.

[Aut14a] AutomationML e.V., editor. *Best Practice Recommendations: Modelling of graph-based structures*. AutomationML e.V., September 2014. [internal].

[Aut14b] AutomationML e.V., editor. Whitepaper AutomationML Communication V1.0.0. `https://www.automationml.org/o.red/uploads/dateien/1494834508-WP_Communication_V1.0.0.zip`, September 2014. Online; accessed 02-November-2017.

[Aut16] AutomationML e.V., editor. OPC Unified Architecture Information Model for AutomationML (Whitepaper) V1.0.0. `https://www.automationml.org/o.red/uploads/dateien/1485865685-WP_OPCUAforAutomationML_V1.0.0.zip`, March 2016. Online; accessed 28-September-2017.

[Aut17a] AutomationML e.V., editor. Application Recommendations: Automation Project Configuration V1.0.0. https://www.automationml.org/o.red/uploads/dateien/1494835012-AR_001E_Automation_Project_Configuration_V1.0.0.zip, April 2017. Online; accessed 02-November-2017.

[Aut17b] AutomationML e.V., editor. Part 4: AutomationML Logic (Whitepaper), V1.5.0. https://www.automationml.org/o.red/uploads/dateien/1490340053-AML_Whitepaper_Logic_V1.5.0.zip, January 2017. Online; accessed 30-October-2017.

[BBS+13] M. Bierkandt, T. Bley, A. Schiffleitner, S. Vetrov, D. Herzberg, S. Ebner, K. Resel, R. Schneider, and M. Fritz. Sustainhub deliverable 3.2 - data model for eco-/sustainability aspects. http://www.sustainhub-research.eu/images/downloads/2013-10-18_v06_Deliverable_3-2.pdf, November 2013. Online; accessed 14-July-2017.

[BE95] B. Bras and J. Emblemsvåg. Use of Activity-Based Costing, Uncertainty, and Disassembly Action Charts in Demanufacture Cost Assessments. In *Proceedings of the 1995 ASME Design Engineering Technical Conferences*, pages 285–292, 1995.

[Bet97] Betriebsorganisation u. Unternehmensentwicklung REFA Verband f. Arbeitsgestaltung, editor. *REFA Methodenlehre der Betriebsorganisation, Datenermittlung: Bd. 2* . Fachbuchverlag Leipzig, München, 1997.

[BH96] B. Bras and R. Hammond. Towards Design for Remanufacturing: Metrics for Assessing Remanufacturing. In S. D. Flapper and A. J. de Ron, editors, *Proceedings of the 1st International Workshop on Reuse*, pages 5–22. Eindhoven, The Netherlands, 1996.

[BIO14] BIO Intelligence Service; in collaboration with Arcadis, Ecologic, Institute for European Environmental Policy (IEEP), Umweltbundesamt (UBA). Development of Guidance on Extended Producer Responsibility (EPR) - final report. Technical report, European Commission - DG Environment, 2014.

[BKKB15] A. Borrmann, M. König, C. Koch, and J. Beetz, editors. *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Springer Vieweg, 2015.

[Bla15] M. Blanz. *Forschungsmethoden und Statistik für die Soziale Arbeit, Grundlagen und Anwendungen*. Verlag W. Kohlhammer, 2015.

[BMB00] BMBF-Verbundprojekt ProMeKreis. Leitfaden zur recyclingrecycling Produktetwicklung - Methoden zur Verbesserung der Kreislaufeignung von Einfach- und Komplexgeräten unter besonderer Berücksichtigung der Recyclingkosten und unter Einbeziehung iinnovative Verwertungstechniken bei der Produktentwicklung. Technical report, BMBF, 2000.

[BNS+15] T. Buchert, S. Neugebauer, S. Schenker, K. Lindow, and R. Stark. Multicriteria decision making as a tool for sustainable product development - benefits and obstacles. *Procedia CIRP*, 26:70 – 75, 2015.

[BOM] BOMcheck.net. Bomcheck.net. `https://www.bomcheck.net/en`. Online; accessed 17-July-2017.

[BtHVH14] T. Bauernhansl, M. ten Hompel, and B. Vogel-Heuser, editors. *Industrie 4.0 in Produktion, Automatisierung und Logistik - Anwendung · Technologien · Migration*. Springer Fachmedien Wiesbaden, 2014.

[CDYM+07] A. F. Cutting-Decelle, R. I. M. Young, J. J. Michel, R. Grangel, J. Le Cardinal, and J. P. Bourey. Iso 15531 mandate: A product-process-resource based approach for managing modularity in production management. *Concurrent Engineering*, 15(2):217–235, 2007.

[COR16] CORDIS - community research and development information service. ReBORN - Innovative Reuse of modular knowledge Based devices and technologies for Old, Renewed and New factories. `http://cordis.europa.eu/project/rcn/108898_en.html?isPermaLink=true`, March 2016. Online; accessed 22-July-2017.

[CRM12] J. Crone, B. Rosemann, and M. Mörtl. Recyclinggerechtes Konstrukieren. In F. Rieg and R. Steinhilper, editors, *Handbuch Konstruktion*, pages 485–509. Carl Hanser Verlag, 2012.

[CT05] M. Colledani and T. Tolio. A Decomposition Method to Support the Configuration / Reconfiguration of Production Systems. *CIRP Annals*, 54:441 – 444, 2005.

[CT13] M. Colledani and T. Tolio. Integrated process and system modelling for the design of material recycling systems. *{CIRP} Annals - Manufacturing Technology*, 62(1):447 – 452, 2013.

[CTMM07] J. Cassina, M. Tomasella, M. Marquard, and A. Metin. Dr9.6a: Specification of the system object model (revised). Report, 507100 PROMISE - A Project of the 6th Framework Programme Information Society Technoliges (IST), 2007.

[Däh17] S. Dähnrich. Analyse des Rückbaus von Produktionssystemen. Masterthesis, Otto-von-Guericke-University Magdeburg, September 2017.

[Dic09] M. Dickerhof. *Ein neues Konzept für das bedarfsgerechte Informations- und Wissensmanagement in Unternehmenskooperationen der Multimaterial-Mikrosystemtechnik*. Dissertation, Instituts für Angewandte Informatik - Automatisierungstechnik, Karlsruher Institut für Technologie, 2009.

[DIN03a] DIN 8580. *Manufacturing processes - Terms and definitions, division*. Beuth Verlag, September 2003.

[DIN03b] DIN 8591. *Manufacturing processes disassembling - Classification, subdivision, definitions*. Beuth Verlag, September 2003.

[DIN03c] DIN8593-0. *Manufacturing processes joining - Part 0: General - Classification, subdivision, terms and definitions*. Beuth Verlag, September 2003.

[DIN05] DIN EN ISO 9000. *DIN EN ISO 9000: Quality management systems - Fundamentals and vocabulary*. Beuth Verlag, December 2005.

[DIN10] DIN EN 62424 Ed. 1.0. *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*. Beuth Verlag, January 2010.

[DIN13] DIN EN ISO 14031. *Environmental management - Environmental performance evaluation - Guidelines*. Beuth Verlag, December 2013.

[DIN16a] DIN EN 62424 Ed. 1.0. *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*. IEC, July 2016.

[DIN16b] DIN SPEC 16592. *Combining OPC Unified Architecture and Automation Markup Language*. Beuth Verlag, December 2016.

[DLH+10] K. Dencovski, U. Löwen, T. Holm, M. Amberg, M. Maurmaier, and P. Göhner. Production System's Life Cycle-Oriented Innovation of Industrial Information Systems. In Javier Silvestre-Blanes, editor, *Factory Automation*. InTech, Rijeka, 2010.

[DLS09] U. Dombrowski, T. Lacker, and S. Sonnentag. *Modernisierung kleiner und mittlerer Unternehmen - Ein ganzheitliches Konzept*. Springer-Verlag Berlin Heidelberg, 2009.

[Dov17] GoIndustry DoveBid. GoIndustry DoveBid - a liquidity services marketplace. http://www.go-dove.com/de/, 2017. Online; accessed 15-September-2017.

[Dra10] R. Drath, editor. *Datenaustausch in der Anlagenplanung mit AutomationML, Integration von CAEX, PLCopen XML und COLLADA*. Springer-Verlag Berlin Heidelberg, 2010.

[DSK+08] J. R. Duflou, G. Seliger, S. Kara, Y. Umeda, A. Ometto, and B. Willems. Efficiency and feasibility of product disassembly: A case-based study. In *{CIRP} Annals - Manufacturing Technology*, volume 57, issue 2, pages 583 – 600, 2008.

[eba17] ebay. ebay. http://www.ebay.de, 2017. Online; accessed 15-September-2017.

[eCl] eCl@ss e.V., editor. eCl@ss classification and product description. https://www.eclass.eu/en.html. Online; accessed 03-November-2017.

[EG07] K. M. Eisenhardt and M. E. Graebner. Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50(1):25–32, 2 2007.

[Eur94] European Commission. EUROPEAN PARLIAMENT AND COUNCIL DIRECTIVE 94/62/EC of 20 December 1994 on packaging and packaging waste. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:31994L0062, December 1994. Online; accessed 29-July-2017.

[Eur16] European Commission. REACH. http://ec.europa.eu/environment/chemicals/reach/reach_en.htm, August 2016. Online; accessed 29-July-2017.

[Eur17] European Commission. Circular economy - implementation of the circular economy action plan. http://ec.europa.eu/environment/circular-economy/index_en.htm, June 2017. Online; accessed 17-June-2017.

[Fed12] Federal Ministry of Justice and Consumer Protection. Verordnung über das Europäische Abfallverzeichnis (Abfallverzeichnis-Verordnung - AVV). https://www.gesetze-im-internet.de/avv/, December 2012. Online; accessed 29-July-2017.

[Fed14a] Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (BMUB). GreenTech made in Germany 4.0 - Umwelttechnologie-Atlas für Deutschland. Technical report, BMUB, July 2014.

[Fed14b] Federal Ministry of Justice and Consumer Protection. Verordnung über die Vermeidung und Verwertung von Verpackungsabfällen (Verpackungsverordnung - VerpackV). https://www.gesetze-im-internet.de/verpackv_1998/BJNR237900998.html, July 2014. Online; accessed 29-July-2017.

[Fed15a] Federal Ministry of Justice and Consumer Protection. Gesetz über das Inverkehrbringen, die Rücknahme und die umweltverträgliche Entsorgung von Batterien und Akkumulatoren (Batteriegesetz - BattG). https://www.gesetze-im-internet.de/battg/, November 2015. Online; accessed 29-July-2017.

[Fed15b] Federal Ministry of Justice and Consumer Protection. Gesetz über das Inverkehrbringen, die Rücknahme und die umweltverträgliche Entsorgung von Elektro- und Elektronikgeräten (Elektro- und Elektronikgerätegesetz - ElektroG). https://www.gesetze-im-internet.de/elektrog_2015/, October 2015. Online; accessed 28-July-2017.

[Fed15c] Federal Ministry of Justice and Consumer Protection. Gesetz über die Durchführung von Maßnahmen des Arbeitsschutzes zur Verbesserung der Sicherheit und des Gesundheitsschutzes der Beschäftigten bei der Arbeit (Arbeitsschutzgesetz - ArbSchG). https://www.gesetze-im-internet.de/arbschg/, August 2015. Online; accessed 29-July-2017.

[Fed16a] Federal Ministry of Justice and Consumer Protection. Gesetz zum Schutz vor schädlichen Umwelteinwirkungen durch Luftverunreinigungen, Geräusche, Erschütterungen und ähnliche Vorgänge (Bundes-Immissionsschutzgesetz - BImSchG). https://www.gesetze-im-internet.de/bimschg/, July 2016. Online; accessed 29-July-2017.

[Fed16b] Federal Ministry of Justice and Consumer Protection. Gesetz zur Förderung der Kreislaufwirtschaft und Sicherung der umweltverträglichen Bewirtschaftung von Abfällen (Kreislaufwirtschaftsgesetz - KrWG). https://www.gesetze-im-internet.de/krwg/, April 2016. Online; accessed 28-July-2017.

[Fed16c] Federal Ministry of Justice and Consumer Protection. Verordnung über die Überlassung, Rücknahme und umweltverträgliche Entsorgung von Altfahrzeugen (Altfahrzeug-Verordnung - AltfahrzeugV). http://www.gesetze-im-internet.de/altautov/index.html, December 2016. Online; accessed 29-July-2017.

[Fed17] Federal Ministry of Justice and Consumer Protection. Gesetz über die friedliche Verwendung der Kernenergie und den Schutz gegen ihre Gefahren (Atomgesetz). https://www.gesetze-im-internet.de/atg/, June 2017. Online; accessed 29-July-2017.

[Fes] Festo AG & Co. KG. Forschung zu Industrie 4.0 - Produktionssysteme der Zukunft. https://www.festo.com/group/de/cms/10966.htm. Online; accessed 15-October-2017.

[FMK+13] S. Fukushige, T. Mizuno, E. Kunii, Y. Matsuyama, and Y. Umeda. Quantitative Design Modification for the Recyclability of Products. In A. Y. C. Nee, B. Song, and S.-K. Ong, editors, *Re-engineering manufacturing for sustainability*, pages 27–33. Springer Singapore, 2013.

[Foe13] M. Foehr. *Integrated Consideration of Product Quality within Factory Automation Systems*. Dissertation, Otto-von-Guericke University Magdeburg, 2013.

[Fou] PLCopen Foundation. PLCopen adds independent schemes to IEC 61131-3. http://www.plcopen.org/pages/tc6_xml/xml_intro/index.htm. Online; accessed 21-July-2017.

[Fra] Fraunhofer IOSB. FA 7.21 Begriffe - Industrie 4.0. http://i40.iosb.fraunhofer.de/FA7.21%20Begriffe%20-%20Industrie%204.0/. Online; accessed 28-September-2017.

[FSD+16] A. Fay, S. Schröck, O. Drumm, U. Löwen, A. Schertl, R. Eckardt, G. Gutermuth, D. Krumsiek, T. Makait, T. Mersch, T. Schindler, and M. Schleipen. Durchgängigkeit in Wertschöpfungsketten von Industrie 4.0 - Seamless Value Chains in Industrie 4.0. In *Kongress Automation 2016*, Baden-Baden, Germany, June 7-8 2016. VDI.

[FTM99] K. Feldmann, S. Trautner, and O. Meedt. Innovative Disassembly Strategies based on Flexible Partial Destructive Tools. *Annual Reviews in Control*, 23:159–164, 1999.

[GAB+11] T. E. Graedel, J. Allwood, J.-P. Birat, B. K. Reck, S. F. Sibley, G. Sonnemann, M. Buchert, and C. Hagelüken. recycling rates of metal - a status report. a report of the working group on global metal flows to the international resource panel. http://www.resourcepanel.org/file/381/download?token=he_rldvr, May 2011. Online; accessed 31-October-2017.

[Gad13] A. Gadatsch. *Grundkurs Geschäftsprozess-Management, Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker*. Vieweg+Teubner Verlag, 2013.

[Gar87] D. A. Garvin. Competing on the eight dimensions of quality. *Harvard Business Review*, 65:101–109, November 1987.

[GB12] E. Geisberger and M. Broy, editors. *agendaCPS - Integrierte Forschungsagenda Cyber-Physical Systems*. Springer-Verlag Berlin Heidelberg, 2012. acatech STUDIE.

[GDKL06] P. Ghodous, R. Dieng-Kuntz, and G. Loureiro. *Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering.* Number v. 143 in Frontiers in Artificical Intelligence and Applications. IOS Press, 2006.

[Gep14] M. Gepp. *Standardisierungsprogramme als Ansatz zur Steigerung der Wirtschaftlichkeit im industriellen Anlagen-Engineering*. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2014.

[Gla91] R. Glazer. Marketing in an Information-Intensive Environment: Strategic Implications of Knowledge as an Asset. *The Journal of Marketing*, 55(4):1–19, October 1991.

[GS117] GS1 Germany GmbH. GS1 Germany. `https://www.gs1-germany.de/`, 2017. Online; accessed 15-October-2017.

[GXN09] P. Gu, D. Xue, and A. Y. C. Nee. Adaptable design: Concepts, methods, and applications. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(11):1367–1387, 2009.

[Hau87] V. Hauff, editor. *Our Common Future. The Brundtland Report of the World Commission on Environment and Development*, Greven, 1987.

[Her10] C. Herrmann. *Ganzheitliches Life Cycle Management - Nachhaltigkeit und Lebenszyklusorientierung in Unternehmen*. Springer-Verlag Berlin Heidelberg, 2010.

[Hil01] A. V. Hill. *The Encyclopedia of Operations Management: A Field Manual and Glossary of Operations Management Terms and Concepts*. Ft Press, July 2001.

[HJA05a] M. Hauschild, J. Jeswiet, and L. Alting. From life cycle assessment to sustainable production: Status and perspectives. *CIRP Annals - Manufacturing Technology*, 54(2):1 – 21, 2005.

[HJA05b] M. Hauschild, J. Jeswiet, and L. Alting. From Life Cycle Assessment to Sustainable Production: Status and Perspectives. *CIRP Annals*, 54(2):1 – 21, 2005.

[HJKB16] D. Hästbacka, E. Jantunen, M. Karaila, and L. Barna. Service-based condition monitoring for cloud-enabled maintenance operations. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, October 23-26 2016.

[HLS16] F. Halstenberg, K. Lindow, and R. Stark. Utilization of product lifecycle data from PLM systems in platforms for Industrial Symbiosis. In *14th Global Conference on Sustainable Manufacturing, GCSM*, Stellenbosch, South Africa, October 2-5 2016.

[HMP+10] M. Hestin, S. Mudgal, V. Portugal, G. Obersteiner, S. Scherhaufer, P. Beigl, A. Pertl, L. Barruetabeña, O. Salas, C. Delgado, R. Peagam, R. Orsato, L. Van Wassenhove, T. Teichert, D. Pokorny, B. Kopacek, F. Becker, K. Schischke, and T. Woolman. deliverable 1.1.: literature review - approaches to zero waste. Technical report, 226752 ZeroWIN: Towards Zero Waste in Industrial Networks, 2010.

[HS94] M. Hartel and D. Spath. *Serienfertigung feinwerktechnischer Produkte, von der Produktplanung bis zum Recycling*, volume VDI-Berichte, 1171, chapter Öko-Portfolio : Methode zur Beurteilung der Recyclingeignung technischer Serienprodukte, pages 371 – 392. VDI Verlag Düsseldorf, 1994.

[HS14] R. Henßen and M. Schleipen. Interoperability between OPC UA and AutomationML. In *8th international conference on digital enterprise technology (det 2014), disruptive innovation in manufacturing engineering towards the 4th industrial revolution*, pages 297–304, 2014.

[HSF+17] C. Hildebrandt, A. Scholz, A. Fay, T. Hadlich, C. Diedrich, M. Dubovy, C. Eck, and R. Wiegand. Semantische Allianz 4.0: Semantische Inhalte für Industrie 4.0. In *Kongress Automation 2017*, Baden-Baden, Germany, June 27. - 28. 2017. VDI.

[Hub01a] A. Huber. *Demontageplanung und -steuerung: Planung und Steuerung industrieller Demontageprozesse mit PPS-Systemen*. Dissertation, Otto-von-Guericke University Magdeburg, Magdeburg, 2001.

[Hub01b] M.-A. Hubig. Erarbeitung einer Methode zur deduktiven Ableitung Strategischer Recyclinganforderungen mit Hilfe der Szenariotechnik. Seminar paper, Centrum für Produktionstechnik CCK of the University Kaiserslautern, 2001.

[Hub14] D. W. Hubbard. *How to measure anything - finding the value of "intangibles" in business*. John Wiley & Sons, 3. edition, 2014.

[Hun12] L. Hundt. *Durchgängiger Austausch von Daten zur Verhaltensbeschreibung von Automatisierungssystemen*. Dissertation, Otto-von-Guericke University, Magdeburg, 2012.

[IEC06] IEC 61987-1. *Industrial-process measurement and control - Data structures and elements in process equipment catalogues - Part 1: Measuring equipment with analogue and digital output*. VDE Verlag, December 2006.

[IEC12] IEC 62337. *Commissioning of electrical, instrumentation and control systems in the process industry - Specific phases and milestones*. IEC, February 2012.

[IEC13] IEC 62264-1. *Enterprise-control system integration - Part 1: Models and terminology*. ISO, May 2013.

[IEC14] IEC 62714-1 ed.1.0. *Engineering data exchange format for use in industrial automation systems engineering - Automation markup language - Part 1: Architecture and general requirements*. IEC, June 2014.

[IEC15a] IEC 62714-2 ed.1.0. *Engineering data exchange format for use in industrial automation systems engineering - Automation markup language - Part 2: Role class libraries*. IEC, March 2015.

[IEC15b] IEC Market Strategy Board (MSB) factory of the future project team in cooperation with the Fraunhofer Institute for Manufacturing Engineering and Automation IPA. Factory of the future. http://www.iec.ch/whitepaper/pdf/iecWP-futurefactory-LR-en.pdf, 2015. Online; accessed 28-October-2017.

[IEC16] IEC TS 62832. *Industrial-process measurement, control and automation - Digital factory framework - Part 1: General principles*. IEC, December 2016.

[ind10] industrietec. Firma industrietec. https://www.kuka.com/de-de/services/gebrauchte-maschinen/kuka-used-roboter, 2010. Online; accessed 02-August-2017.

[Int98] International Electrotechnical Commission. Electropedia: The world's online electrotechnical vocabulary - iev ref: 101-12-03 "data". hhttp://www.electropedia.org/iev/iev.nsf/display?openform&ievref=101-12-03, April 1998. Online; accessed 26-September-2017.

[Int01] International Electrotechnical Commission. Electropedia: The world's online electrotechnical vocabulary - iev ref: 151-11-25 "equipment". http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=151-11-25, July 2001. Online; accessed 05-February-2017.

[Int11] International Electrotechnical Commission. Electropedia: The world's online electrotechnical vocabulary - iev ref: 113-05-05 "matter". http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=113-05-05, April 2011. Online; accessed 05-February-2017.

[Int13] International Electrotechnical Commission. Electropedia: The world's online electrotechnical vocabulary - iev ref: 01-07-12 "life cycle". http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=901-07-12, November 2013. Online; accessed 05-February-2017.

[Int15] International Electrotechnical Commission. Electropedia: The world's online electrotechnical vocabulary - iev ref: 01-07-12 "process". hhttp://www.electropedia.org/iev/iev.nsf/display?openform&ievref=351-42-33, October 2015. Online; accessed 08-October-2017.

[Int17a] International Electrotechnical Commission. Tc 65/sc 65b/wg 7. http://www.iec.ch/dyn/www/f?p=103:14:23371420397373::::FSP_ORG_ID,FSP_LANG_ID:2598,25, 2017. Online; accessed 28-September-2017.

[Int17b] International Electrotechnical Commission. Tc 65/sc 65e/wg 9. http://www.iec.ch/dyn/www/f?p=103:14:0::::FSP_ORG_ID,FSP_LANG_ID:7309,25, 2017. Online; accessed 28-September-2017.

[ISO99] ISO 10303-225. *Industrial automation systems and integration - Product data representation and exchange - Part 225: Application protocol: Building elements using explicit shape representation*. ISO, December 1999.

[ISO04] ISO 18629-1. *Industrial automation systems and integration - Process specification language - Part 1: Overview and basic principles*. ISO, November 2004.

[ISO05] ISO 10303-227. *Industrial automation systems and integration - Product data representation and exchange- Part 227: Application protocol: Plant spatial configuration*. ISO, October 2005.

[ISO07] ISO 10303-221. *Industrial automation systems and integration - Product data representation and exchange - Part 221: Application protocol: Functional data and their schematic representation for process plants*. ISO, December 2007.

[ISO12] ISO/PAS 17506. *Industrial automation systems and integration - COLLADA digital asset schema specification for 3D visualization of industrial data*. ISO, July 2012.

[ISO13] ISO 16739. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. ISO, April 2013.

[ISO14a] ISO 10303-242. *Industrial automation systems and integration - Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering*. ISO, December 2014.

[ISO14b] ISO 55000. *Asset management - Overview, principles and terminology*. ISO, March 2014.

[ISO15] ISO 15746-1. *Automation systems and integration - Integration of advanced process control and optimization capabilities for manufacturing systems - Part 1: Framework and functional model*. ISO, April 2015.

[JAA+93] F. Jovane, L. Alting, A. Armillotta, W. Eversheim, K. Feldmann, G. Seliger, and N. Roth. A Key Issue in Product Life Cycle: Disassembly. *CIRP Annals*, 42(2):CIRP Annals, 1993.

[JCSF12] T. Jäger, L. Christiansen, M. Strube, and A. Fay. Durchgängige Werkzeugunterstützung von der Anforderungserhebung bis zur Anlagenstrukturbeschreibung mittels formalisierter Prozessbeschreibung und AutomationML. In *Proceedings of EKA 2012 - Entwurf komplexer Automatisierungssysteme*, Magdeburg, Germany, May, 8-10 2012.

[KGR16] J. Koch, A. Gritsch, and G. Reinhart. Process design for the management of changes in manufacturing: Toward a manufacturing change management process. *CIRP Journal of Manufacturing Science and Technology*, 14(Supplement C):10 – 19, 2016.

[KIK08] H. Kaebernick, S. Ibbotson, and S. Kara. Cradle-to-cradle manufacturing. In P.W. Newton, editor, *Transitions: Pathways towards Sustainable Urban Development in Australia*, pages 521–536. CSIRO Press, 2008.

[KK14] S. Knoch and A. Kröner. A mobile mashup for accessing distributed recycling knowledge. *International Journal on Advances in Internet Technology (IntTech)*, 7(1 + 2):124 – 135, 2014.

[Kle14] P. W. Klein. *Methode zum Engineering von Produktionsanlagen durch Wiederverwendung von Modulen*. Dissertation, University Siegen, Siegen, 2014.

[KM93] G. Keoleian and D. Menerey. Life cycle design guidance manual - environmental requirements and the product system. Technical report, United States Environmental Protection Agency, 1993.

[KM08] D. Kiritsis and B. Moseng. Promise final activity report. Report, 507100 PROMISE - A Project of the 6th Framework Programme Information Society Technoliges (IST), 2008.

[KUK] KUKA. KUKA Gebrauchtroboter direkt vom Hersteller. `https://www.kuka.com/de-de/services/gebrauchte-maschinen/kuka-used-roboter`. Online; accessed 02-August-2017.

[KWH13] H. Kagermann, W. Wahlster, and J. Helbig. Securing the future of German manufacturing industry, Recommendations for implementing the strategic initiative INDUSTRIE 4.0, Final report of the Industrie 4.0 Working Group. Technical report, acatech - National Academy of Science and Engineering, April 2013.

[LBL+16] K. Lenkenhoff, A. Barthelmey, K. Lemmerz, B. Kuhlenkötter, and J. Deuse. Communication architecture for automatic plant documentation updates. *Procedia CIRP*, 44(Supplement C):365 – 370, 2016. 6th CIRP Conference on Assembly Technologies and Systems (CATS).

[LE13] M. Lindahl and S. Ekermann. Structure for Categorization of EcoDesign Methods and Tools. In A. Y. C. Nee, B. Song, and S.-K. Ong, editors, *Re-engineering manufacturing for sustainability*, pages 117–122. Springer Singapore, 2013.

[LFH+11] A. Lüder, M. Foehr, L. Hundt, M. Hoffmann, Y. Langer, and S. Frank. Aggregation of engineering processes regarding the mechatronic approach. In *16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–7, Toulouse, France, September 5-9 2011. IEEE.

[LFKB12] A. Lüder, M. Foehr, A. Köhlein, and B. Böhm. Application of engineering processes analysis to evaluate benefits of mechatronic engineering. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012)*, pages 1–4, Sept 2012.

[LHK10] A. Lüder, L. Hundt, and A. Keibel. Description of manufacturing processes using AutomationML. In *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, Bilbao, Spain, September 13-16 2010.

[Lif09] Life Cycle Initiative. guidelines for social life cycle assessment of products. Technical report, United Nations Environment Programme, 2009.

[Lin09] U. Lindemann. *Methodische Entwicklung technischer Produkte - Methoden flexibel und situationsgerecht anwenden*. Springer-Verlag Berlin Heidelberg, 2009.

[LK03] M. A. Levin and T. T. Kalal. *Improving Product Reliability: Strategies and Implementation*. Wiley, 2003.

[LMB09] U. Lindemann, M. Maurer, and T. Braun. *Structural Complexity Management - An Approach for the Field of Product Design*. Springer-Verlag Berlin Heidelberg, 2009.

[LMF16] Y. Lu, KC Morris, and S. Frechette. Current standards landscape for smart manufacturing systems. Technical report, NIST - National Institute of Standards and Technology, February 2016.

[log17] logi.cals GmbH. AML.hub - Software-Plattform für das parallele Engineering von Anlagen für Industrie 4.0. `http://www.amlhub.at/`, 2017. Online; accessed 16-September-2017.

[LRS+14] A. Lüder, R. Rosendahl, N. Schmidt, D. Ryashentseva, J. Vollmer, M. Foehr, M. Heinecke, and T. Petzold. Integrated Engineering Landscape "IntEL". Technical report, Siemens AG and Otto-von-Guericke University Magdeburg, 2014.

[LS17a] A. Lüder and N. Schmidt. AutomationML in a Nutshell. In B. Vogel-Heuser, T. Bauernhansl, and M. ten Hompel, editors, *Handbuch Industrie 4.0 Bd.2 - Automatisierung*, pages 213 – 258. Springer Vieweg, 2. edition, 2017.

[LS17b] A. Lüder and N. Schmidt. Herstellerunabhängiger Austausch von Entwurfsdaten für Steuerungssysteme mittels AutomationML. In C. Manzei, L. Schleupner, and R. Heinze, editors, *Industrie 4.0 im internationalen Kontext - Kernkonzepte, Ergebnisse, Trends*, pages 115 – 142. VDE Beuth Verlag, 2017.

[LSC+16] A. Lüder, N. Schmidt, A. Calá, D. Ryashentseva, and N. Schulz. Qualitätsorientierte Modellierung von Produktionssystemverhalten. In *Entwurf komplexer Automatisierungssysteme : EKA 2016 ; Beschreibungsmittel, Methoden, Werkzeuge und Anwendungen, 14. Fachtagung mit Tutorium*, Magdeburg, Germany, May 24-25 2016.

[LSD17] A. Lüder, N. Schmidt, and R. Drath. Standardized Information Exchange Within Production System Engineering. In S. Biffl, A. Lüder, and D. Gerhard, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems - Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 235 – 257. Springer International Publishing, 2017.

[LSH+17a] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza. Description Means for Information Artifacts Throughout the Life Cycle of CPPS. In S. Biffl, A. Lüder, and D. Gerhard, editors, *Multi-Disciplinary Engineering* *for Cyber-Physical Production Systems - Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 169–183. Springer International Publishing, 2017.

[LSH+17b] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza. Fundamentals of Artifact Reuse in CPPS. In S. Biffl, A. Lüder, and D. Gerhard, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems - Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 113–138. Springer International Publishing, 2017.

[LSH+17c] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza. Identification of Artifacts in Life Cycle Phases of CPPS. In S. Biffl, A. Lüder, and D. Gerhard, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems - Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 139–167. Springer International Publishing, 2017.

[LSR+13] A. Lüder, N. Schmidt, R. Rosendahl, S. Biffl, and H. Steininger. Entwicklung von Empfehlungen zur Entwicklungsprozessverbesserung im Bereich Programmiersysteme "E3BP". Technical report, Christian Doppler Laboratory "Software Engineering Integration for Flexible Automation Systems" (TU Vienna), Otto-von-Guericke-University Magdeburg, 2013.

[LSS+17] A. Lüder, M. Schleipen, N. Schmidt, J. Pfrommer, and R. Henßen. One step towards an Industry 4.0 component. In *CASE 2017- 13th International Conference on Automation Science and Engineering*, page 6 pp., Xi'an, China, August, 20 - 23 2017.

[LSsB06] M. Lindahl, E. Sundin, J. Östlin, and M. Björkman. Concepts and definitions for product recovery - analysis and clarification of the terminology used in academia and industry. In D. Brissaud, S. Tichkiewitch, and P. Zwolinski, editors, *Innovation in Life Cycle Engineering and Sustainable Development*, pages 123 – 138. Springer Netherlands, Dordrecht, 2006.

[Mar11] H. Martens. *Recyclingtechnik - Fachbuch für Lehre und Praxis*. Springer Spektrum, 2011.

[Mat15] D. Matzke. Modellierung von Prozessbeschreibungen zur automatischen Suche nach mechatronischen Einheiten in AutomationML basierten Bibliotheken. Seminar paper, Otto-von-Guericke-University Magdeburg, February 2015.

[MB92] W. McDonough and M. Braungart. The Hannover Principles - Design for Sustainability. copyrighted document., 1992.

[MBS+11] H. Mersch, D. Behnen, D. Schmitz, U. Eplle, C. Brecher, and M. Jarke. Gemeinsamkeiten und Unterschiede der Prozess- und Fertigungstechnik - Interdisziplinäre Aspekte der Produktionsmodellierung. *at - Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik*, 59(1):7–17, 2011.

[MCK+10] A. Middendorf, P. Chancerel, C. Kersten, M. Hestin, A. Deprouw, X. Vallvé, A. D. Bojarski, O. Collell, A. Ting, W. Lin, L. Chang, T. Nittka, and M. Webbels.

deliverable 3.1.: End-of-Life Impacts under appropriate and uncontrolled treatment. Technical report, 226752 ZeroWIN: Towards Zero Waste in Industrial Networks, 2010.

[MKD17] S. Magerstedt, B. Kuhlenkötter, and J. Deuse, editors. *Technische Dokumentation, intelligent erstellen und nutzen - Abschlussbericht zum Industrie 4.0 Projekt Cyber System Connector*. Shaker, April 2017.

[MMB02] P. G. Maropoulos, K. R. McKay, and D. D. G. Bramall. Resource-aware aggregate planning for the distributed manufacturing enterprise. *CIRP Annals - Manufacturing Technology*, 51:363 – 366, 2002.

[MMM+12] M. Marwede, A. Middendorf, P. Maher, J. Ospina, A. Ting, M. Yang, P. Arranz, and C. Jehoulet. deliverable 3.2.: Guidance Document on D4R for PV Systems and Flat Panel Products. Technical report, 226752 ZeroWIN: Towards Zero Waste in Industrial Networks, 2012.

[MP15] C. Muggeo and M. Pfenning. Die Rolle von MBSE und PLM im Industrial Internet. In S.-O. Schulze and C. Muggeo, editors, *Tag des Systems Engineering*, pages 279–287. Carl Hanser Verlag GmbH & Co. KG, 2015.

[MT02] D. Mangun and D. L. Thurston. Incorporating component reuse, remanufacture, and recycle into product portfolio design. *IEEE Transactions on Engineering Management*, 49(4):479–490, November 2002.

[Nah13] S. Nahkala. Aligning product design methods and tools for sustainability. In A. Y. C. Nee, B. Song, and S.-K. Ong, editors, *Re-engineering manufacturing for sustainability*, pages 53–58. Springer Singapore, 2013.

[NC94] D. Navin-Chandra. The Recovery Problem in Product Design. *Journal of Engineering Design*, 5(1):65–86, 1994.

[Nic96] W. Nickel, editor. *Recycling-Handbuch - Strategien, Technologien, Produkte*. Springer-Verlag Berlin Heidelberg, 1996.

[Nit09] M. Nitzsche. *Graphen für Einsteiger*. Vieweg+Teubner Verlag, 2009.

[NRA08] P. Nyhuis, G. Reinhart, and E. Abele, editors. *Wandlungsfähige Produktionssysteme: Heute die Industrie von morgen gestalten*. TEWISS, 2008.

[NTW09] J. Niemann, S. Tichkiewitch, and E. Westkämper, editors. *Design of Sustainable Product Life Cycles*. Springer-Verlag Berlin Heidelberg, 2009.

[Off00] Official Journal of the European Union. DIRECTIVE 2000/53/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 18 September 2000 on end-of life vehicles. `http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32000L0053&qid=1501327197388`, September 2000. Online; accessed 29-July-2017.

[Off06a] Official Journal of the European Union. DIRECTIVE 2006/66/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 6 September 2006 on batteries and accumulators and waste batteries and accumulators and repealing Directive 91/157/EEC. `http://eur-lex.europa.eu/`

legal-content/EN/ALL/?uri=CELEX:32006L0066, September 2006. Online; accessed 29-July-2017.

[Off06b] Official Journal of the European Union. Regulation (EC) No 1907/2006 of the European Parliament and of the Council of 18 December 2006 concerning the Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH). http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1501337481958&uri=CELEX:32006R1907, December 2006. Online; accessed 29-July-2017.

[Off08] Official Journal of the European Union. DIRECTIVE 2008/98/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 19 November 2008 on waste and repealing certain Directives. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32008L0098, November 2008. Online; accessed 28-July-2017.

[Off11] Official Journal of the European Union. DIRECTIVE 2011/65/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 8 June 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment. http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1501272820509&uri=CELEX:32011L0065, June 2011. Online; accessed 28-July-2017.

[Off12] Official Journal of the European Union. DIRECTIVE 2012/19/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 4 July 2012 on waste electrical and electronic equipment (WEEE). http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32012L0019, July 2012. Online; accessed 28-July-2017.

[OHB+13] M. Obst, T. Holm, S. Bleuel, U. Claussnitzer, L. Evertz, T. Jäger, and T. Nekolla. Automatisierung im Life Cycle modularer Anlagen: Welche Veränderungen und Chancen sich ergeben. *Atp-Edition : automatisierungstechnische Praxis*, 1/2:24–31, 2013.

[OHK10] C. O'Connor, M. Hestin, and R. Kuehr. deliverable 8.1: policy synthesis report. Technical report, 226752 ZeroWIN: Towards Zero Waste in Industrial Networks, 2010.

[Ohl06] M. Ohlendorf. *Simulationsgestützte Planung und Bewertung von Demontagesystemen*. Dissertation, Technische Universität Braunschweig, Braunschweig, 2006.

[Ott16] Otto-von-Guericke Universität Magdeburg, Lehrstuhl für mechanische Verfahrenstechnik, J. Tomas. Vorlesung Aufbereitungstechnik und Recycling. http://www.mvt.ovgu.de/Lehre/Vorlesung+ATRC.html, June 2016. Online; accessed 02-November-2017.

[otZAD12] System Aspects Working Group of the ZVEI Automation Division. Life-Cycle-Management for Automation Products and Systems. Guideline, The German Electrical and Electronic Manufacturers' Association (ZVEI) Automation Division, Frankfurt, September 2012.

[Oxf17a] Oxford Dictionaries. Definition of asset in oxford dictionaries. https://en.oxforddictionaries.com/definition/asset, 2017. Online; accessed 26-September-2017.

[Oxf17b] Oxford Dictionaries. Definition of connection in oxford dictionaries. https://en.oxforddictionaries.com/definition/connection, 2017. Online; accessed 28-September-2017.

[Oxf17c] Oxford Dictionaries. Definition of digital in oxford dictionaries. https://en.oxforddictionaries.com/definition/digital, 2017. Online; accessed 28-September-2017.

[Oxf17d] Oxford Dictionaries. Definition of material in oxford dictionaries. https://en.oxforddictionaries.com/definition/material, 2017. Online; accessed 26-September-2017.

[Oxf17e] Oxford Dictionaries. Definition of matter in oxford dictionaries. https://en.oxforddictionaries.com/definition/matter, 2017. Online; accessed 26-September-2017.

[Oxf17f] Oxford Dictionaries. Definition of matter in oxford dictionaries. https://en.oxforddictionaries.com/definition/virtual, 2017. Online; accessed 28-September-2017.

[Pan13] V. Pandey. *Decision Based Design*. CRC Press, 2013.

[Pan15] International Resource Panel. How metals determine our lives. http://www.resourcepanel.org/file/385/download?token=yHAwYrBO, 2015. Online; accessed 31-October-2017.

[PAS04] PAS 1049. *transmission of recycling relevant product Information between producers and recyclers - the recycling passport*. Beuth Verlag, December 2004.

[PBFG07] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, editors. *Engineering Design: A Systematic Approach*. Springer Verlag, 3. edition edition, 2007.

[PE94] T. U. Pimmler and S. D. Eppinger. Integration analysis of product decompositions. In *ASME Design Theory and Methodology Conference*, pages 343–351, Minneapolis, Minnesota, USA, September 11-14 1994.

[PER] PERFoRM project. PERFoRM - Production harmonizEd Reconfiguration of Flexible Robots and Machinery. http://www.horizon2020-perform.eu/. Online; accessed 27-December-2017.

[PL11] J. Ponn and U. Lindemann. *Konzeptentwicklung und Gestaltung technischer Produkte - Systematisch von Anforderungen zu Konzepten und Gestaltlösungen*. Springer, 2011.

[Pla16a] Plattform Industrie 4.0. Aspects of the research roadmap in application scenarios - working paper. https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.pdf?__blob=publicationFile&v=8, April 2016. Online; accessed 04-February-2017.

[Pla16b] Plattform Industrie 4.0. Aspects of the research roadmap in application scenarios - working paper - circular economy use case description, April 2016.

[PLC] PLCopen association. TC6 - expressable XML Schemes. http://www.plcopen.org/pages/tc6_xml/. Online; accessed 28-September-2017.

[RABW09] S. Rahimifard, M. S. S. Abu Bakar, and D. J. Williams. Recycling process planning for the end-of-life management of waste from electrical and electronic equipment. *CIRP Annals - Manufacturing Technology*, 58(1):5–8, 2009.

[RMTF99] B. Rosemann, H. Meerkamm, S. Trautner, and K. Feldmann. Design for Recycling, Recycling Data Management and Optimal End-of-Life Planning based on Recycling-Graphs. In *International Conference on Engineering Design (ICED 1999)*, pages 1–6, Munich, Germany, August 24 - 26 1999.

[RSI00] C. M. Rose, A. Stevels, and K. Ishii. A new approach to end-of-life design advisor (ELDA). In *Electronics and the Environment, 2000. ISEE 2000. Proceedings of the 2000 IEEE International Symposium on*, pages 99–104, San Francisco, CA, USA, May 8 - 10 2000. IEEE.

[Ruh06] K. Ruhland. *Methoden und Werkzeuge zur recyclinggerechten Automobilentwicklung*. Dissertation, University Kaiserslautern, 2006.

[SBK01] G. Seliger, B. Basdere, and T. Keil. e-Cycling platform for profitable reuse. In *Assembly and Task Planning, 2001, Proceedings of the IEEE International Symposium on*, pages 453–457, Fukuoka, Japan, May 28 - 29 2001. IEEE.

[SBSW12] A. Schiffleitner, T. Bley, R. Schneider, and D.-P. Wimpff. Stakeholder perspectives on business model requirements for a sustainability data exchange platform across supply chains. In *Proceedings of the Joint International Conference and Exhibition "Electronics Goes Green"*, page 5 pp., Berlin, Germany, September 9 - 12 2012. Fraunhofer Verlag.

[Sch99] E. Schnieder. *Methoden der Automatisierungstechnik - Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Vieweg+Teubner Verlag, 1999.

[Sch04] F. Schultmann. Industrielles produktions- und logistikmanagement. In *Produktion und Umwelt: Festschrift für Otto Rentz: Festschrift Für Otto Rentz*. Springer, 2004.

[Sch05] M. Schröter. *Strategisches Ersatzteilmanagement in Closed-Loop Supply Chains - Ein systemdynamischer Ansatz*. Dissertation, TU Braunschweig, 2005.

[SD09] M. Schleipen and R. Drath. Three-view-concept for modeling process or manufacturing plants with AutomationML. In *2009 IEEE Conference on Emerging Technologies Factory Automation*, pages 1–4, Sept 2009.

[SDF02] M. Santochi, G. Dini, and F. Failli. Computer Aided Disassembly Planning: State of the Art and Perspectives. *CIRP Annals*, 51:507 – 529, 2002.

[Sen09] U. Sendler. *Das PLM-Kompendium: Referenzbuch des Produkt-Lebenszyklus-Managements*. Springer, 2009.

[Sen13] U. Sendler, editor. *Industrie 4.0 - Beherrschung der industriellen Komplexität mit SysLM*. Springer-Vieweg, 2013.

[Sen16] U. Sendler, editor. *Industrie 4.0 grenzenlos*. Springer Vieweg, September 2016.

[SFLS13] T. Schäffler, M. Foehr, A. Lüder, and K. Supke. Engineering process evaluation: Evaluation of the impact of internationalisation decisions on the efficiency and quality of engineering processes. In *2013 IEEE International Symposium on Industrial Electronics*, pages 1–6, May 2013.

[SFR02] F. Schultmann, M. Fröhlich, and O. Rentz. Demontageplanung und -steuerung mit Enterprise-Resource- und Advanced-Planning-Systemen. *Wirtschaftsinformatik (WI-Aufsatz)*, 6(44):557–565, 2002.

[Sim] Simulation Interoperability Standards Organization - SISO. Cmsd psg - core manufacturing simulation data. https://www.sisostds.org/StandardsActivities/SupportGroups/CMSDPSG-CoreManufacturingSimulationData.aspx. Online; accessed 21-July-2017.

[Ski12] SkillPro project. Skill-based Propagation of "Plug&Produce"-Devices in Reconfigurable Production Systems by AML. http://www.skillpro-project.eu/, 2012. Online; accessed 27-December-2017.

[SL15] N. Schmidt and A. Lüder. AutomationML in a Nutshell. https://www.automationml.org/o.red/uploads/dateien/1447420977-AutomationML%20in%20a%20Nutshell_151104.pdf, November 2015. Online; accessed 30-September-2017.

[SL16] N. Schmidt and A. Lüder. Development of a generic model for End-of-Life scenarios of production systems. In *14th Global Conference on Sustainable Manufacturing, GCSM*, Stellenbosch, South Africa, October 2-5 2016.

[SL17a] N. Schmidt and A. Lüder. End-of-Life support for production systems. In *13. Magdeburger Maschinenbau-Tage 2017: AUTONOM - VERNETZT - NACHHALTIG*, pages 438–444, Magdeburg, Germany, September 27-28 2017.

[SL17b] N. Schmidt and A. Lüder. Recovery Planning Method as End-of-Life Support for Production Systems. In *IEEE 15th International Conference of Industrial Informatics INDIN'2017*, Emden, Germany, July, 24-26 2017.

[SLH+16] N. Schmidt, A. Lüder, K. Hell, H. Röpke, and J. Zawisza. A generic model for the End-of-Life phase of production systems. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 5693–5698, Florence, Italy, October 23-26 2016.

[SLR+15] N. Schmidt, A. Lüder, R. Rosendahl, D. Ryashentseva, M. Foehr, and J. Vollmer. Surveying integration approaches for relevance in cyber physical production systems. In *20th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Luxembourg, September 8 - 11 2015. IEEE.

[SO13] E. O. Simolowo and C. E. Onovughe. Automation of Generation of Models for Disassembly Process Planning for Recycling. In *Proceedings of the World Congress on Engineering (WCE 2013)*, volume III, pages 1–5, London, U.K., July 3 - 5 2013.

[Sos00] M. E. Sosa. *Analyzing the effects of product architecture on technical communication in product development organizations*. Dissertation, Massachusetts Institute of Technology, Cambridge, 2000.

[SPPR97] T. Spengler, H. Püchert, T. Penkuhn, and O. Rentz. Environmental integrated production and recycling management. *European Journal of Operational Research*, 97(2):308 – 326, 1997.

[SR16] M. Schütz and H. Röbken. *Bachelor- und Masterarbeiten verfassen - Abschlussarbeiten in Organisationen*. Springer Fachmedien Wiesbaden, 2016.

[SSZ+14] M. Shuaib, D. Seevers, X. Zhang, F. Badurdeen, K. E. Rouch, and I. S. Jawahir. Product Sustainability Index (ProdSI) - A Metrics-based Framework to Evaluate the Total Life Cycle Sustainability of Manufactured Products. *Journal of Industrial Ecology*, 18(4):491–507, 2014.

[Str14] M. Strube. *Modellgestützte Modernisierungsplanung industrieller Automatisierungslösungen*. Dissertation, Helmut Schmidt university, Hamburg, 2014.

[Sus] SustainHub consortium. Sustainhub. http://www.sustainhub-research.eu/. Online; accessed 16-July-2017.

[SWM10] M. Schenk, S. Wirth, and E. Müller. *Factory Planning Manual - Situation-Driven Production Facility Planning*. Springer Berlin Heidelberg, 2010.

[Syl17] E. Syllwasschy. Identifikation von relevanten Informationen zur Rückgewinnung von Automatisierungskomponenten aus Produktionssystemen. Bachelorthesis, Hochschule Magdeburg Stendal, February 2017.

[The16] The Federal Government. Logo Bundesregierung Leitfaden zur Stilllegung, zum sicheren Einschluss und zum Abbau von Anlagen oder Anlagenteilen nach § 7 des Atomgesetzes. http://www.verwaltungsvorschriften-im-internet.de/bsvwvbund_23062016_RSI31703161.htm, June 2016. Online; accessed 29-July-2017.

[the17] the recycling company ALBA Group. Recycling in Deutschland. https://www.recyclingnews.info/wp-content/uploads/RecyclingInDeutschland_DINA1_042017_D.pdf, April 2017. Online; accessed 18-October-2017.

[Tit16] F. Titov. *Technologiegestützte Angebotserstellung für den Umbau komplexer Investitionsgüter*. Dissertation, Technical University Hamburg, 2016.

[VDA97] VDA 231-106. *Material Classification in Motor Vehicle Construction*. Dokumentation Kraftfahrwesen e.V., January 1997.

[VDA07] VDA 260. *Components of motor vehicles; Marking of material*. Dokumentation Kraftfahrwesen e.V., April 2007.

[VDI01a] VDI 2343. *Recycling of electrical and electronic products: Principles and terminology*. Beuth Verlag, May 2001.

[VDI01b] VDI 4431. *Life-cycle management in the manufacturing industry*. Beuth Verlag, July 2001.

[VDI02] VDI 2243. *Recycling-oriented product development*. Beuth Verlag, July 2002.

[VDI04] VDI 2206. *Design methodology for mechatronic systems*. Beuth Verlag, June 2004.

[VDI05] VDI 2884. *Purchase, operating and maintenance of production equipment using Life Cycle Costing (LCC)*. Beuth Verlag, December 2005.

[VDI14] VDI 2074. *Recycling in the building service*. Beuth Verlag, July 2014.

[VDI15a] VDI/VDE 3682 Part 1. *Formalised process descriptions - Concept and graphic representation*. Beuth Verlag, May 2015.

[VDI15b] VDI/VDE 3682 Part 2. *Formalised process descriptions - Information model*. Beuth Verlag, May 2015.

[VDI16a] VDI 3810 Part 7 (draft). *Betreiben und Instandhalten von Gebäuden und gebäudetechnischen Anlagen - Verwertung - Abbruch, Abriss, Rückbau, Demontage*. Beuth Verlag, March 2016.

[VDI16b] VDI 6210 Part 1. *Demolition of civil constructions and technical facilities*. Beuth Verlag, February 2016.

[VDM06] VDMA 34160. *Prognosemodell für die Lebenszykluskosten von Maschinen und Anlagen*. Beuth Verlag, June 2006.

[VDM15] VDMA Arbeitsgemeinschaft Großanlagenbau, Lagebericht 2014/2015. Lagebericht 2014/2015: Weltweite Krisen meistern - lokale Chancen nutzen, Beiträge zum Industrieanlagenbau. Technical report, VDMA Arbeitsgemeinschaft Großanlagenbau, 2015.

[VDM17a] VDMA Arbeitsgemeinschaft Großanlagenbau, Lagebericht 2016/2017. Lagebericht 2016/2017: Die Welt im Wandel - Flexibel agieren, Beiiträge zum Industrieanlagenbau. Technical report, VDMA Arbeitsgemeinschaft Großanlagenbau, 2017.

[VDM17b] VDMA Industrie 4.0 Forum and Fraunhofer Application Center Industrial Automation (IOSB-INA). Industrie 4.0 Communication Guideline Based on OPC UA. Technical report, VDMA Verlag GmbH, 2017.

[Ves02] F. Vester. *Die Kunst vernetzt zu denken: Ideen und Werkzeuge für einen neuen Umgang mit Komplexität Ein Bericht an den Club of Rome*. dtv Verlagsgesellschaft, 2002.

[Web08] K. H. Weber. *Dokumentation verfahrenstechnischer Anlagen: Praxishandbuch mit Checklisten und Beispielen*. Springer Berlin Heidelberg, 2008.

[WGE+17] C. Wagner, J. Grothoff, U. Epple, R. Drath, S. Malakuti, S. Grüner, M Hoffmeister, and R. Zimmermann. The role of the Industrie 4.0 Asset Administration Shell and the Digital Twin during the life cycle of a plant. In *22th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, September 12-15 2017.

[Win13] P. Winzer. *Generic Systems Engineering - Ein methodischer Ansatz zur Komplexitätsbewältigung*. Springer Vieweg, 2013.

[WLH10] D. Wünsch, A. Lüder, and M. Heinze. Flexibility and re-configurability in manufacturing by means of distributed automation systems - an overview. In H. Kühnle, editor, *Distributed Manufacturing*, pages 51–70. Springer London, 2010.

[WRN15] H.-P. Wiendahl, J. Reichardt, and P. Nyhuis. *Handbook Factory Planning and Design*. Springer-Verlag Berlin Heidelberg, 2015.

[WSCL13] E. Westkämper, D. Spath, C. Constantinescu, and L. Lentes, editors. *Digitale Produktion*. Springer Vieweg, 2013.

[WSDB03] B. Willems, G. Seliger, J. Duflou, and B. Basdere. Contribution to design for adaptation: Method to Assess the Adaptability of Products (MAAP). In *Proceedings of EcoDesign2003: Third International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, pages 589–596, Tokyo, Japan, December 8 - 11 2003. IEEE.

[WUL+16] S. Wassilew, L. Urbas, J. Ladiges, A. Fay, and T. Holm. Transformation of the NAMUR MTP to OPC UA to allow plug and produce for modular process automation. In *21th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, September 6-9 2016.

[WW10] E. Westkämper and H.-J. Warnecke. *Einführung in die Fertigungstechnik*. Vieweg+Teubner, 8. edition, 2010.

A Requirements and information on recovery

This annex provides the requirements and information gathered through a literature survey. If no publications are named the author has added it. Neither requirements nor information make a claim to be complete or exhaustive. The following classification is used: system structure (SS), connection structure (CS), material structure (MS), and system design and characterization (SD). The assignment of the requirements to the categories of the classification was made by the author as well as the information to the requirements.
Even though, this annex shows all gathered requirements with their information, only this information was modeled in UML class diagrams, which are relevant for the feed forward of artifacts (see the cases 16 and 17 from Figure 2.6).

A.1 System structure

This section provides the requirements and information on the system structure of a production system. For this, Table A.1 comprises the requirements with giving the corresponding sources. From Table A.2 to Table A.11 each requirement is, then, considered separately regarding the information that comes along with it as well as the modeling of it in the UML class diagram. Additionally, the resulting UML class diagram for the system structure is depicted in Figure A.1.

Table A.1: Requirements on system structure

No.	Requirements	Source
SS1	Minimize complexity of working principles in order to reduce the number of components	[PBFG07]
SS2	Select carefully between function integration (e.g. recovery unfriendly composites) and function separation (e.g. sub-function modules)	[PBFG07], [Hub01b]
SS3	Consider functional modular structure	[VDI02], [PBFG07], [Hub01b], [Ruh06], [DSK+08], [WSDB03], [NTW09], [Hub01a]
SS4	Separate heavy and low duty components and areas in order to limit damage zones in case of problems	[PBFG07]
SS5	Create a reconditioning friendly structure	[CRM12], [WSDB03]
SS6	Create a preprocessing friendly structure	[CRM12]
SS7	Optimize the system structure	[CRM12]
SS8	Standardize components and assemblies that have similar or identical functions	[PBFG07], [Hub01b], [Hub01a]
SS9	Specify system structure	[CRM12], [PAS04], [VDI02], [SFR02], [PBFG07], [Hub01b], [Ruh06], [FTM99], [RMTF99], [DSK+08], [SO13], [HS94], [Hub01a], [WSDB03]
SS10	Easy identification of system and components	[CRM12], [PAS04], [PBFG07], [KK14], [Hub01a], [Pla16a]

Table A.2: SS1 - Minimize complexity of working principles in order to reduce the number of components

Information	Source	Representation in UML class diagram
Identifiability of components in a production system	-	Create a class for *production system* and *component* as well as a relation between them.

Table A.3: SS2 - Select carefully between function integration (e.g. recovery unfriendly composites) and function separation (e.g. sub-function modules)

Information	Source	Representation in UML class diagram
Give the function	[PBFG07], [VDM06], [SBK01], [Pla16a]	Create an attribute *function* at class *production system* and *component*.

Table A.4: SS3 - Consider functional modular structure

Information	Source	Representation in UML class diagram
Design system as decoupled functions and with decoupled interfacial, function relevant relationships (necessary for modularity)	[WSDB03], [HS94]	On the one hand create an attribute *function* at class *production system* and *component* (see Table A.3) and on the other hand create a class *connection* with a relation to class *production system* and *component*.

Table A.5: SS4 - Separate heavy and low duty components and areas in order to limit damage zones in case of problems

Information	Source	Representation in UML class diagram
Give the function	-	Create an attribute *function* at class *production system* and *component* (see Table A.3).

Table A.6: SS5 - Create a reconditioning friendly structure

Information	Source	Representation in UML class diagram
Identifiability of components with their connections in a production system	-	On the one hand create a class for *production system* and *component* (see Table A.2) and on the other hand create a class *connection* (see Table A.4).
Provide information about reconditioning options of the components	[SFR02]	This is recovery process relevant and will be considered later.
Provide information about reconditioning steps and costs	[VDI02], [Hub01a]	This is recovery process relevant and will be considered later.

Table A.7: SS6 - Create a preprocessing friendly structure

Information	Source	Representation in UML class diagram
Identifiability of components and their materials	-	On the one hand create a class for *production system* and *component* (see Table A.2) and on the other hand create a class *material* with a relation to class *component*.
Provide information about preprocessing options of the components	[SFR02]	This is recycling process relevant and will be considered later.
Provide information about reconditioning steps, technologies, and costs	[Ruh06], [Hub01a]	This is recycling process relevant and will be considered later.
Calculate recycling costs	[CRM12], [VDI02],[Hub01b], [VDM06], [Hub01a]	This is recycling process relevant and will be considered later.
Provide process parameter of the preprocessing process	[Ruh06]	This is recycling process relevant and will be considered later.
Provide information about the material and property loss due to the preprocessing process	[Ruh06]	This is recycling process relevant and will be considered later.

Table A.8: SS7 - Optimize the system structure

Information	Source	Representation in UML class diagram
Certain models can be used to optimize the system structure. These need the system structure and its properties.	[CRM12]	In general, it is realized by UML classes with their properties and their relations.
Provide information about recovery options for the components	[SFR02], [SPPR97], [Hub01a]	This is recovery process relevant and will be considered later.
Provide certain instructions for the process	[PAS04]	This is recovery process relevant and will be considered later.

Table A.9: SS8 - Standardize components and assemblies that have similar or identical function

Information	Source	Representation in UML class diagram
Give the function	-	Create an attribute *function* at class *production system* and *component* (see Table A.3).
Modeling of component hierarchy	-	The class *component* can consist of other components.

Table A.10: SS9 - Specify system structure

Information	Source	Representation in UML class diagram
Give number of components	[HS94]	Create a class for *component*.
Modeling of the assembly sequence	[SBK01]	On the one hand create a class for *component* (see Table A.2) and on the other hand create a class *connection* (see Table A.4).
Give the complexity of the system structure	[HS94]	In general, it is realized by UML classes with their relations.

Table A.11: SS10 - Easy identification of system and components

Information	Source	Representation in UML class diagram
Provide information to distinguish a system as 'new', 'used', or 'remanufactured' (or as primary commodity, used system, or secondary commodity)	[SFR02], [Hub01b]	Create an attribute *recovery state* at class *production system* and *component*.
Provide information if a component is reusable, recyclable, or needs to be disposed	[SFR02]	This is recovery process relevant and will be considered later.
Provide information about recycling strategy	[Hub01b]	This is recovery process relevant and will be considered later.
Use of location designations (what is the current location)	[otZAD12], [SO13], [SPPR97]	Create an attribute *location designation* at class *production system* and *component*.
Provide vendor information	[PAS04], [KK14], [Hub01a]	Create an attribute *manufacturer information* at class *production system* and *component*.
Give each component a number	[PAS04]	Create an attribute *ID* at class *production system* and *component*.
Name laws, guidelines, directives that are met by the system	[PAS04]	Create an attribute *certificate* at class *production system* and *component*.
Give quantity of components in inventory	[SO13]	This is recovery process relevant and will be considered later.
Provide a visualization	[PAS04], [KK14], [Pla16a]	Create an attribute *2D/3D representation* at class *production system* and *component*.

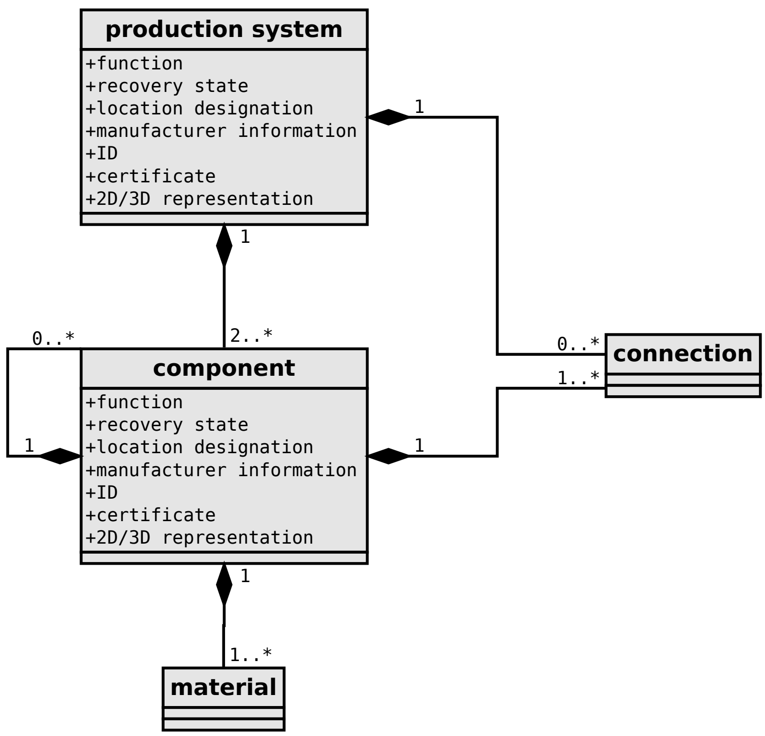

Figure A.1: Resulting UML class diagram based on requirements and information on system structure

A.2 Connection structure

This section provides the requirements and information on the connection structure of a production system. For this, Table A.12 comprises the requirements with giving the corresponding sources. From Table A.13 to Table A.33 each requirement is, then, considered separately regarding the information that comes along with it as well as the modeling of it in the UML class diagram. Additionally, the resulting UML class diagram for the connection structure is depicted in Figure A.2.

Table A.12: Requirements on connection structure

No.	Requirements	Source
CS1	Ensure accessibility of components and connections	[CRM12], [VDI02], [PBFG07], [Hub01b], [Ruh06], [DSK+08], [WSDB03]
CS2	No impairment of disassembly	[CRM12]
CS3	Enable nondestructive disassembly which eases recovery	[CRM12], [VDI02], [PBFG07], [Hub01a]
CS4	Easy disassembly results in reduced costs	[VDI02]
CS5	Ensure disassembility for components (destructive, nondestructive)	[VDI02], [Ruh06], [FTM99]
CS6	Disassembly time and costs are reduced when tool changes are low	[VDI02]
CS7	Easy to access and easy to disassemble results in good replaceability	[VDI02]
CS8	Create a disassembly friendly structure	[CRM12], [PAS04], [PBFG07], [Hub01b], [Ruh06], [FTM99], [DSK+08], [HS94], [Hub01a]
CS9	Reducing disassembly time results in reduced costs	[VDI02], [Hub01a]
CS10	Proper choice of disassembly techniques dependent on the future usage of the disassembly components	[SFR02], [SO13]
CS11	Create a preprocessing friendly connection structure	[CRM12]
CS12	Specify connection structure	[CRM12], [PAS04], [VDI02], [SFR02], [PBFG07], [Hub01b], [Ruh06], [HS94], [Hub01a], [WSDB03]
CS13	Analyze connection structure of the system	[CRM12]
CS14	Documentation of the connection types	[CRM12]
CS15	Realize a shallow disassembly depth	[CRM12], [PBFG07], [HS94], [WSDB03]
CS16	Create a reassembly friendly structure	[CRM12]
CS17	Reduce interfaces/connections	[PBFG07], [Hub01b]
CS18	Specify defined connections	[CRM12], [PBFG07], [Hub01b]
CS19	Usage of standard tools, avoid specialized tools - This eases the disassembly	[VDI02], [PBFG07], [Hub01a], [WSDB03]
CS20	Standardize connection elements that have similar or identical functions	[PBFG07], [Hub01b], [Hub01a]
CS21	Create easy to separate components	[PBFG07]

Table A.13: CS1 - Ensure accessibility of components and connections

Information	Source	Representation in UML class diagram
How can the system/component be opened/accessed?	[PAS04], [HS94]	Create an attribute *order* at class *connection*.
Provide construction/design information	[VDI02], [Hub01b], [Hub01a], [SBK01], [WSDB03]	Create an attribute *2D/3D representation* at class *production system* and class *component* (see Table A.11).
Provide assembly plans	[Hub01a]	Create the attributes *type* and *order* at the class *connection*. The class *connection* has a relation to the classes *production system* and *component*.

Table A.14: CS2 - No impairment of disassembly

Information	Source	Representation in UML class diagram
Give information about deterioration state (property loss) of the connection	-	Create an attribute *deterioration state* at class *connection*.

Table A.15: CS3 - Enable nondestructive disassembly which eases recovery

Information	Source	Representation in UML class diagram
Give information about connections that can be undone nondestructively	-	This can be derived by the attribute *type* of the class *connection*.
Which components are faulty	[SFR02]	Create an attribute *deterioration state* at class *component*. This attribute can, then, be used to derive the information about faulty components.
Provide ratio of faulty parts after disassembling	[SFR02]	This is disassembly process relevant and will be considered later.
Degree of quality reduction due to be process	[PBFG07], [SBK01]	This is disassembly process relevant and will be considered later.
Which components can be recovered	[Hub01b], [Pla16a]	This can be derived by the attribute *deterioration state* of the class *component*.

Table A.16: CS4 - Easy disassembly results in reduced costs

Information	Source	Representation in UML class diagram
Give information about how the connection can be undone	-	This can be derived by the attribute *type* of the class *connection*.

Table A.17: CS5 - Ensure disassembility for components (destructive, nondestructive)

Information	Source	Representation in UML class diagram
Give information about connections that can be undone nondestructively or destructively	-	This can be derived by the attribute *type* of the class *connection*.

Table A.18: CS6 - Disassembly time and costs are reduced when tool changes are low

Information	Source	Representation in UML class diagram
Give information about with which tool the connections can be undone	-	This can be derived by the attribute *type* of the class *connection*.

Table A.19: CS7 - Easy to access and easy to disassemble results in good replaceability

Information	Source	Representation in UML class diagram
Give information about the disassembly complexity	-	This can be derived by the attribute *type* and *order* of the class *connection*.
Give information about working surface for the tools	[Ruh06]	This can be derived by the attribute *2D/3D representation* of the class *connection*.

Table A.20: CS8 - Create a disassembly friendly structure

Information	Source	Representation in UML class diagram
Give information about the system structure complexity	[PAS04], [VDI02], [FTM99], [RMTF99], [HS94]	This can be derived by the classes *production system*, *component*, and *connection* as well as their relations among them.
Give disassembly technique	[VDI02], [Hub01b], [Ruh06], [FTM99], [RMTF99], [DSK+08], [SO13], [Hub01a], [WSDB03]	This can be derived by the attribute *type* of the class *connection*.
What is input and output of the process	[Hub01a]	This is disassembly process relevant and will be considered later.
Give information about disassembly costs	[CRM12], [VDI02], [SFR02], [Ruh06], [DSK+08], [SPPR97], [VDM06]	This is disassembly process relevant and will be considered later.

Table A.21: CS9 - Reducing disassembly time results in reduced costs

Information	Source	Representation in UML class diagram
Give information about applicable disassembly techniques	-	This can be derived by the attribute *type* of the class *connection*.

Table A.22: CS10 - Proper choice of disassembly techniques dependent on the future usage of the disassembly components

Information	Source	Representation in UML class diagram
Give information about disassembly order and applicable disassembly techniques	[CRM12], [VDI02], [SFR02], [PBFG07], [Ruh06], [HS94], [SPPR97], [KK14], [Hub01a]	This can be derived by the attribute *type* and *order* of the class *connection*.
Give information about disassembly time	[CRM12], [VDI02], [Ruh06], [RMTF99], [HS94], [Hub01a], [WSDB03]	This is disassembly process relevant and will be considered later.
Give information about removal directions per component	[SO13], [WSDB03]	Create an attribute *removal direction* at class *component*.
Provide disassembly plans	[CRM12], [VDI02], [SFR02], [RMTF99], [HS94], [SPPR97], [Hub01a]	This can be derived by the attribute *type* and *order* of the class *connection*.
Give information about alternative disassembly orders	[SFR02], [Hub01a]	This can be derived by the attribute *type* and *order* of the class *connection*.
Provide disassembly direction	[PBFG07], [RMTF99], [WSDB03]	Create an attribute *disassembly direction* at class *connection*.
Prioritize the to be disassembled connections	[Ruh06], [RMTF99], [SO13], [SPPR97], [WSDB03]	Create an attribute *order* at class *connection* (see Table A.13).
How difficult is it to undone a connection?	[RMTF99], [HS94]	This can be derived by the attribute *type* of the class *connection*.
Give information about disassembly tooling: manual operations, partially automated, fully automated operations	[DSK+08], [SPPR97]	This can be derived by the attribute *type* of the class *connection*.
Give information about disassembly depth	[SFR02], [DSK+08], [Hub01a]	This can be derived by the classes *production system*, *component*, and *connection* as well as their relations among them.
Provide safety instructions on the disassembly	[VDI02], [RMTF99]	This is disassembly process relevant and will be considered later.
Provide information about right amount of force to be applied to undo connection	[SO13], [Hub01a]	Create an attribute *condition* at the class *connection*.

Table A.23: CS11 - Create a preprocessing friendly connection structure

Information	Source	Representation in UML class diagram
Provide information about the type of material used	-	Create an attribute *type* at class *material*.

Table A.24: CS12 - Specify connection structure

Information	Source	Representation in UML class diagram
Provide information about the connection and how it was assembled/joined	-	Create an attribute *type* at the class *connection* (see Table A.13).
Provide information about certain disassembly technique needed to undo a connection	cite02PAS10492004, [VDI02]	This can be derived by the attribute *type* of the class *connection*.
Provide information about the connection and with which extra material is was joined	-	Create the classes *material* and *connection* with a relation between them.

Table A.25: CS13 - Analyze connection structure of the system

Information	Source	Representation in UML class diagram
Provide information about the type of connection	[HS94]	This can be derived by the attribute *type* of the class *connection*.

Table A.26: CS14 - Documentation of the connection types

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	This can be derived by the attribute *type* of the class *connection*.

Table A.27: CS15 - Realize a shallow disassembly depth

Information	Source	Representation in UML class diagram
Give information about disassembly depth	-	This can be derived by the classes *production system*, *component*, and *connection* as well as their relations among them.

Table A.28: CS16 - Create a reassembly friendly structure

Information	Source	Representation in UML class diagram
Give information about reassembly time	[WSDB03]	This is reassembly process relevant and will be considered later.

Table A.29: CS17 - Reduce interfaces/connections

Information	Source	Representation in UML class diagram
Give number of connections	-	This can be derived by the class *connection*.

Table A.30: CS18 - Specify defined connections

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	This can be derived by the attribute *type* of the class *connection*.

Table A.31: CS19 - Usage of standard tools, avoid specialized tools - This eases the disassembly

Information	Source	Representation in UML class diagram
Provide information about tools needed to undo a connection	-	This is disassembly process relevant and will be considered later.

Table A.32: CS20 - Standardize connection elements that have similar or identical functions

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	This can be derived by the attribute *type* of the class *connection*.

Table A.33: CS21 - Create easy to separate components

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	This can be derived by the attribute *type* of the class *connection*.

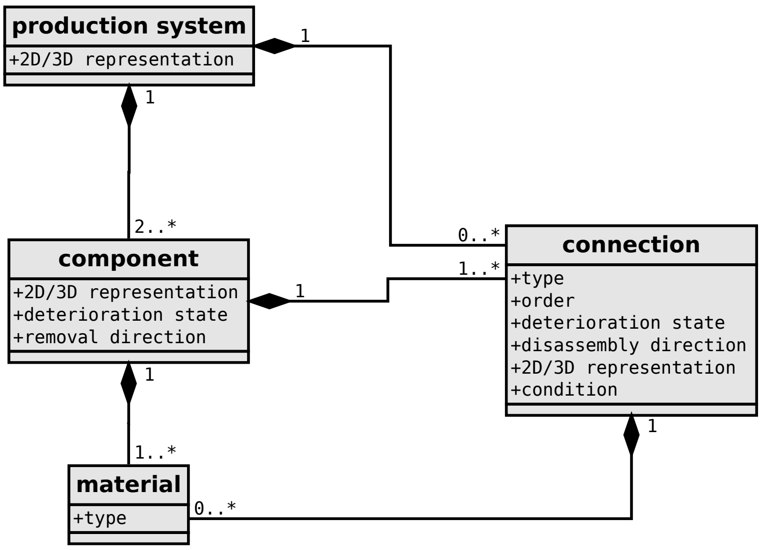

Figure A.2: Resulting UML class diagram based on requirements and information on connection structure

A.3 Material structure

This section provides the requirements and information on the material structure of a production system. For this, Table A.34 comprises the requirements with giving the corresponding sources. From Table A.35 to Table A.53 each requirement is, then, considered separately regarding the information that comes along with it as well as the modeling of it in the UML class diagram. Additionally, the resulting UML class diagram for the material structure is depicted in Figure A.3.

Table A.34: Requirements on material structure

No.	Requirements	Source
MS1	Reduce harmful substances	[Hub01a]
MS2	Meet the REACH standard on chemicals	[CRM12]
MS3	Identify to declare and forbidden substances (constantly updated by EU or other directives)	[CRM12]
MS4	Ensure wear resistance	[CRM12], [DSK+08]
MS5	Ensure corrosion resistance	[CRM12], [PBFG07]
MS6	Reduce amount of material used and reduce used material types	[CRM12], [PBFG07], [Hub01b], [HS94]
MS7	An increased amount of material results in a profitable material recovery	[VDI02]
MS8	Use recycling compatible materials and/or recyclable materials or components	[CRM12], [VDI02], [PBFG07], [Hub01b], [Ruh06], [DSK+08], [Hub01a]
MS9	Define surface finish (e.g. varnish)	[Ruh06]
MS10	Use recyclates	[CRM12], [MB92]
MS11	Enable material recovery	[PAS04]
MS12	Use material combinations that are easy to separate	[CRM12], [Hub01b], [Ruh06], [RMTF99], [Hub01a]
MS13	Ensure sortability	[CRM12], [PBFG07], [RMTF99], [Hub01a]
MS14	Arrange in assemblies consisting of components and material that are compatible regarding reprocessing	[PBFG07]
MS15	Consider recyclability of the system in the early phase of the engineering process	[VDI02], [PBFG07], [HS94], [Hub01a], [Pla16a]
MS16	Specify recycling friendly structures	[HS94], [MB92]
MS17	Ensure deformation resistance	[Hub01b]
MS18	Ensure aging resistance of materials and connections	[Hub01b]
MS19	No impairment of the operability of recycling plants	[CRM12]

Table A.35: MS1 - Reduce harmful substances

Information	Source	Representation in UML class diagram
Provide information about harmful substances/material used	[CRM12], [CRM12], [PAS04], [PBFG07], [Ruh06], [Hub01a]	This can be derived by the attribute *type* of the class *material*.
Name laws, guidelines, directives that are met by the system	-	Create an attribute *certificate* at class *material*.

Table A.36: MS2 - Meet the REACH standard on chemicals

Information	Source	Representation in UML class diagram
Identifiability of alarming substances (SVHC)	[CRM12]	This can be derived by the attribute *type* of the class *material*.

Table A.37: MS3 - Identify to declare and forbidden substances (constantly updated by EU or other directives)

Information	Source	Representation in UML class diagram
Provide complete list of all materials used in a system	[CRM12]	This can be derived by the attribute *type* of the class *material* which has a relation to the class *component* which has a relation to the class *production system*.

Table A.38: MS4 - Ensure wear resistance

Information	Source	Representation in UML class diagram
Identify actual deterioration state/state of wear	[PBFG07], [Hub01b]	Create an attribute deterioration state at classes *material*, also at *connection*, *component*, and *production system* (see Table A.15). Between the class *material* and *connection* is a relation since material is used to realize a connection.

Table A.39: MS5 - Ensure corrosion resistance

Information	Source	Representation in UML class diagram
Identify actual deterioration state/state of wear (material loss)	[PBFG07], [Hub01b]	Create the attributes *type* and *deterioration state* at the class *material*.

Table A.40: MS6 - Reduce amount of material used and reduce used material types

Information	Source	Representation in UML class diagram
Identifiability of materials used according to DIN	[CRM12], [Hub01b], [Ruh06]	Create an attribute *type* at class *material* (see Table A.23).
Give information about statistical life time of raw material	[HS94]	This can be derived by the attribute *type* of the class *material*.
Provide information about compound	[PAS04], [VDI02], [SFR02],[Ruh06], [FTM99], [RMTF99], [SO13]	This can be derived by the attribute *type* of the class *material*.
Provide vendor information	-	Create an attribute *manufacturer information* at class *material*.

Table A.41: MS7 - An increased amount of material results in a profitable material recovery

Information	Source	Representation in UML class diagram
Provide information about amount of material used	-	This can be derived by the attribute *type* of the class *material* and the attribute *2D/3D representation* at class *component* that needs to include volume information (see Table A.11).

Table A.42: MS8 - Use recycling compatible materials and/or recyclable materials or components

Information	Source	Representation in UML class diagram
Provide more detailed information about the material used	-	This can be derived by the attribute *type* of the class *material*.

Table A.43: MS9 - Define surface finish (e.g. varnish)

Information	Source	Representation in UML class diagram
Provide more detailed information about the material used	-	This can be derived by the attribute *type* of the class *material*.

Table A.44: MS10 - Use recyclates

Information	Source	Representation in UML class diagram
Provide information about materials used	-	Create an attribute *recovery state* at class *material*.

Table A.45: MS11 - Enable material recovery

Information	Source	Representation in UML class diagram
Provide information about valuable materials used	[CRM12], [PAS04], [PBFG07], [Hub01a]	This can be derived by the attribute *type* of the class *material*.
Provide a visualization	[PAS04], [VDI02], [Hub01a]	Create an attribute *2D/3D representation* at class *production system* and *component*.

Table A.46: MS12 - Use material combinations that are easy to separate

Information	Source	Representation in UML class diagram
Provide information about materials used	-	This can be derived by the attribute *type* of the class *material*.

Table A.47: MS13 - Ensure sortability

Information	Source	Representation in UML class diagram
Provide information about sorting techniques	-	This is recycling process relevant and will be considered later.

Table A.48: MS14 - Arrange in assemblies consisting of components and material that are compatible regarding reprocessing

Information	**Source**	**Representation in UML class diagram**
Provide information about component's materials used	-	This can be derived by the attribute *type* of the class *material* which has a relation to the class *component*.

Table A.49: MS15 - Consider recyclability of the system in the early phase of the engineering process

Information	**Source**	**Representation in UML class diagram**
Provide for each component a certain recycling technique	[VDI02], [SFR02], [PBFG07], [Hub01b], [Ruh06], [RMTF99], [SPPR97], [KK14]	This is recycling process relevant and will be considered later.
Provide for each component a recycling option	[CRM12], [PAS04], [SFR02], [Ruh06]	This is recycling process relevant and will be considered later.
How much waste is produced during recycling process?	[Ruh06], [Hub01a]	This is recycling process relevant and will be considered later.
What is input and output of the process?	[Ruh06]	This is recycling process relevant and will be considered later.
What is process sequence?	[CRM12], [SFR02], [Ruh06], [Hub01a]	This is recycling process relevant and will be considered later.

Table A.50: MS16 - Specify recycling friendly structures

Information	**Source**	**Representation in UML class diagram**
Provide degree of recycling	[Ruh06]	This is recycling process relevant and will be considered later.
Provide information about recycling rate	[Ruh06]	This is recycling process relevant and will be considered later.
Consider material upgrading possibilities	[PBFG07]	This is recycling process relevant and will be considered later.
Give recycling classes/categories	[CRM12], [PAS04], [SFR02], [Ruh06]	This is recycling process relevant and will be considered later.

Table A.51: MS17 - Ensure deformation resistance

Information	**Source**	**Representation in UML class diagram**
Provide information about materials used	-	This can be derived by the attribute *type* of the class *material*.

Table A.52: MS18 - Ensure aging resistance of materials and connections

Information	Source	Representation in UML class diagram
Provide information about materials used	-	This can be derived by the attribute *type* of the class *material* that is associated with the class *component* and *connection*.
Provide information about materials used for the connections/interface	-	Create the classes *connection* and *material* with the relation between them.

Table A.53: MS19 - No impairment of the operability of recycling plants

Information	Source	Representation in UML class diagram
Identification of extraneous and problematic material	[CRM12], [PAS04], [VDI02], [RMTF99]	This can be derived by the attribute *type* of the class *material*.
Specify classes of hazardous substances	[SFR02]	This can be derived by the attribute *type* of the class *material*.
Hazardous substances according to which directive/law etc.?	[PAS04]	This can be derived by the attribute *type* of the class *material*.
Specify degree of contamination	[PBFG07]	Create an attribute *condition* at the classes *production system*, *component*, and *material*.
Provide information about capacity constraints (capacity restrictions and costs)	[SPPR97]	This is recycling process relevant and will be considered later.
Identification of material that is critical for environmental and recycling	[VDI02]	This can be derived by the attribute *type* of the class *material*.

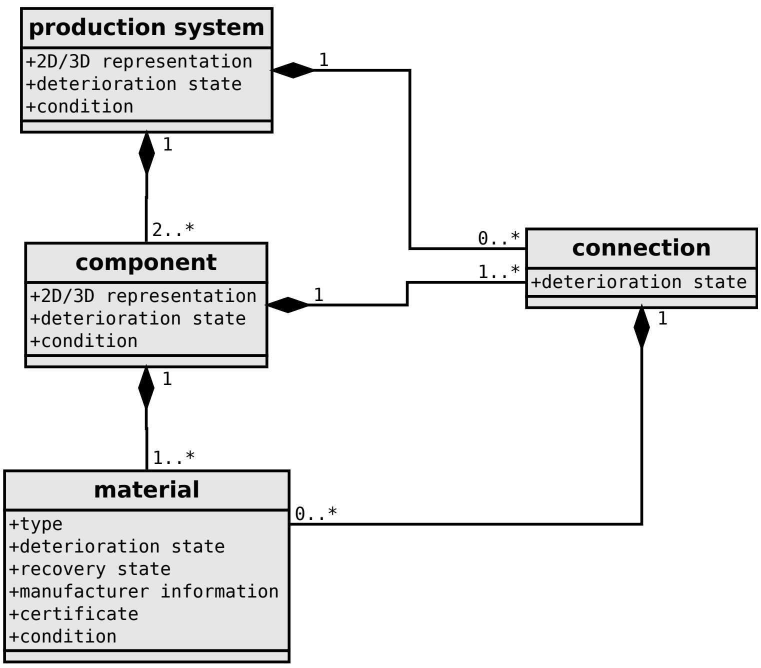

Figure A.3: Resulting UML class diagram based on requirements and information on material structure

A.4 System design and characterization

This section provides the requirements and information on the design and characterization of a production system and its components. In some cases, it comprises the system, connection, and material structure. For this, Table A.54 comprises the requirements with giving the corresponding sources. From Table A.55 to Table A.81 each requirement is, then, considered separately regarding the information that comes along with it as well as the modeling of it in the UML class diagram. Additionally, the resulting UML class diagram for the system design and characterization is depicted in Figure A.4.

Table A.54: Requirements on design and characterization

No.	Requirements	Source
SD1	System must be digitally cleanable (protection of know-how)	[OHB+13]
SD2	Use a consistent naming	[PAS04]
SD3	Enable depollution and material recovery	[PAS04], [Ruh06], [DSK+08]
SD4	Design for transportation	[PBFG07]
SD5	Reduce production waste	[Hub01a]
SD6	Increase operation time	[Hub01a]
SD7	Which influence does the product have on the system that has produced it?	-
SD8	Reduce or avoid environmental impact	[VDI02]
SD9	Operating liquids within the system must be drained easy and fast.	[VDI02]
SD10	Select working principles that produce little wear in order to extend product life within the same boundary conditions	[PBFG07]
SD11	Optimize the system usage throughout all life cycles	[Hub01a]
SD12	Evaluation of suitability for further use phases	[SBK01]
SD13	Components must be capable of being reused in the next generation of the system. This requires compatibility of the components.	[Hub01a]
SD14	Enable replaceability of components	[CRM12], [PBFG07], [DSK+08]
SD15	Enable remanufacturing	[CRM12], [PBFG07], [LSsB06], [Hub01a], [SBK01], [WSDB03]
SD16	Enable repair-ability	[CRM12], [PBFG07]
SD17	Ensure clean-ability, e.g. by avoiding to design gaps	[CRM12], [PBFG07], [DSK+08], [OHB+13]
SD18	Ensure testability, e.g. check deterioration state	[CRM12], [PBFG07], [Hub01b], [DSK+08]
SD19	Create the possibility to modernize a system during operation phase	[Hub01a]
SD20	Enable the decommissioning	-
SD21	Avoid to have costly process step afterwards	[VDI02]
SD22	Consider environmental friendliness of the system in early phases	[VDI02], [HS94]
SD23	Design a proper reverse logistics network phases	[SO13], [KK14], [Hub01a], [Pla16a]
SD24	Reduce emissions	[VDI02], [Pla16a]
SD25	Design for reusability	[DSK+08]
SD26	Components must meet the safety regulations that are applied for new components.	[Hub01a]
SD27	Changes of the system during operation in operation phase need to be included in the documentation (which components were added, deterioration state, work load)	[SFR02], [Hub01b], [Ruh06], [FTM99], [DSK+08], [SO13], [OHB+13], [Hub01a], [SBK01]

Table A.55: SD1 - System must be digitally cleanable (protection of know-how)

Information	Source	Representation in UML class diagram
Provide information about how the system can be digitally cleaned	-	Create a class *memory* with an attribute *type* that has a relation to the classes *production system* and *component*. Then, this can be derived by the attribute *type* of the classes *production system* or *memory*.

Table A.56: SD2 - Use a consistent naming

Information	Source	Representation in UML class diagram
Give IDs	-	Create an attribute *ID* at class *production system* and *component* (see Table A.11).

Table A.57: SD3 - Enable depollution and material recovery

Information	Source	Representation in UML class diagram
Assign recycling activities to components or the system	[CRM12], [SFR02], [Ruh06]	This is recycling process relevant and will be considered later.
Identification of materials and substances	[CRM12], [PAS04], [VDI02], [PBFG07], [Hub01b], [Ruh06], [HS94], [SPPR97], [KK14], [CRM12], [Pla16a]	Create an attribute *type* at class *material* (see Table A.23).
Assign recycling techniques to components or the system	[FTM99], [DSK+08], [Ruh06]	This is recycling process relevant and will be considered later.
Give information about the total amount/weight of a certain material or percentage share	[CRM12], [PAS04], [VDI02], [Ruh06], [RMTF99], [HS94], [KK14], [Hub01a]	This can be derived by the attribute *type* of the class *material* and by the attribute *2D/3D representation* of the classes *production system* or *component*.
Give information about the quantitative relations of single components	[SFR02]	This can be derived by the attribute *type* of the class *material* and by the attribute *2D/3D representation* of the classes *production system* or *component*.
Provide information about the size	[CRM12], [PAS04], [Ruh06],[RMTF99]	This can be derived by the attribute *type* of the class *material* and by the attribute *2D/3D representation* of the classes *production system* or *component*.
Provide information about the costs for the final disposal	[DSK+08]	This is recycling process relevant and will be considered later.
Provide information about special tools necessary to use	[PAS04], [VDI02], [Ruh06]	This is disassembly process relevant and will be considered later.
Provide information about the assembly of the components	[PAS04], [VDI02], [SFR02]	In general, it is realized by UML classes with their relations.
Provide information about disassembly technologies	[RMTF99], [Hub01a]	This is disassembly process relevant and will be considered later.
Which statutory provisions need to be met (prohibited materials, obligatory disassembly activities, limitation of energetic recovery)?	[Ruh06]	This is recovery process relevant and will be considered later.

Table A.58: SD4 - Design for transportation

Information	Source	Representation in UML class diagram
Give the function of components and connection types	-	On the one hand create an attribute *function* at class *production system* and *component* and on the other hand create an attribute *type* at class *connection*.

Table A.59: SD5 - Reduce production waste

Information	Source	Representation in UML class diagram
Provide information about the design of the system or components	-	Create an attribute *2D/3D representation* at the classes *production system* and *component*.
Provide software used	-	This can be derived by the attribute *type* of the class *memory*.

Table A.60: SD6 - Increase operation time

Information	Source	Representation in UML class diagram
Provide information about the design of the system or components	-	Create an attribute *2D/3D representation* at the classes *production system* and *component*.
Provide software used	-	This can be derived by the attribute *type* of the class *memory*.

Table A.61: SD7 - Which influence does the product have on the system that have produced it?

Information	Source	Representation in UML class diagram
Provide information about deterioration state	-	Create an attribute *deterioration state* at the classes *production system*, *component*, and *material*.

Table A.62: SD8 - Reduce or avoid environmental impact

Information	Source	Representation in UML class diagram
Provide information about the design of the system or components	-	Create an attribute *2D/3D representation* at the classes *production system* and *component*.
Provide software used	-	This can be derived by the attribute *type* of the class *memory*.

Table A.63: SD9 - Operating liquids within the system must be drained easy and fast.

Information	Source	Representation in UML class diagram
Provide information about relevant connections/interfaces	-	This can be derived by the attribute *type* and *order* of the class *connection*.

Table A.64: SD10 - Select working principles that produce little wear in order to extend product life within the same boundary conditions

Information	Source	Representation in UML class diagram
Provide information about the design of the system or components	-	Create an attribute *2D/3D representation* at the classes *production system* and *component*.

Table A.65: SD11 - Optimize the system usage throughout all life cycles

Information	Source	Representation in UML class diagram
Provide information about the design of the system or components	-	Create an attribute *2D/3D representation* at the classes *production system* and *component*.
Provide software used	-	This can be derived by the attribute *type* of the class *memory*.

Table A.66: SD12 - Evaluation of suitability for further use phases

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	This is recovery process relevant and will be considered later.
Give information about the condition of the system and components (e.g. load history, failure history, maintenance / repair history)	-	Create an attribute deterioration state at classes *material*, also at *connection*, *component*, and *production system* (see Table A.15). Between the class *material* and *connection* is a relation since material is used to realize a connection.

Table A.67: SD13 - Components must be capable of being reused in the next generation of the system. This requires compatibility of the components.

Information	Source	Representation in UML class diagram
Provide information about recovery options for the system or components for identical or different functions	[PBFG07]	This is recovery process relevant and will be considered later.

Table A.68: SD14 - Enable replaceability of components

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create an attribute *function* at the classes *production system* and *component*. Create an attribute *type* at the class *connection*.

Table A.69: SD15 - Enable reconditioning

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	This is recovery process relevant and will be considered later.

Table A.70: SD16 - Enable repair-ability

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	This is recovery process relevant and will be considered later.

Table A.71: SD17 - Ensure clean-ability, e.g. by avoiding to design gaps

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	This is recovery process relevant and will be considered later.

Table A.72: SD18 - Ensure testability, e.g. check deterioration state

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	This is recovery process relevant and will be considered later.

Table A.73: SD19 - Create the possibility to modernize a system during operation phase

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create an attribute *function* at the classes *production system* and *component*. Create an attribute *type* at the class *connection*.

Table A.74: SD20 - Enable the decommissioning

Information	Source	Representation in UML class diagram
Provide information about decommissioning costs	[VDM06]	This is recovery process relevant and will be considered later.

Table A.75: SD21 - Avoid to have costly process step afterwards

Information	Source	Representation in UML class diagram
Provide certain instructions to depollute fast and easy	-	This is recovery process relevant and will be considered later.
Provide information about the working time required	[CRM12], [VDI02]	This is recovery process relevant and will be considered later.
Provide information about the capacities of the waste disposal contractor	[SPPR97], [Hub01a]	This is recovery process relevant and will be considered later.
Economic assessment of the recycling processes including disassembly, separation, preprocessing, and logistics	[VDI02], [SBK01]	This is recycling process relevant and will be considered later.
Provide information about status of operating capacity of the disassembly facilities	[SO13]	This is disassembly process relevant and will be considered later.
Give information about commodity market fluctuations	[SO13]	This is recovery process relevant and will be considered later.
Identify target fraction of the recycler	[CRM12]	This is recovery process relevant and will be considered later.
Give information about the target fraction for each component	[VDI02], [Ruh06]	This is recycling process relevant and will be considered later.
Give information about process parameters (e.g. degree of purity of fractions)	[VDI02], [Ruh06]	This is recycling process relevant and will be considered later.
Give information about indivisible materials	[Ruh06]	This can be derived by the attribute *function* of the class *component*.
Identify target valuable material and non-target material	[CT13]	This can be derived by the attribute *type* of the class *material* that has a relation to the class *component*.
Provide information about reconstruction costs	[VDM06]	This is recovery process relevant and will be considered later.
Provide guided benchmarking including clearly defined KPIs (e.g. energy efficiency)	[SBSW12]	This is recovery process relevant and will be considered later.
Provide information about costs and revenues	[PAS04], [Ruh06], [RMTF99], [DSK+08], [HS94], [SPPR97], [VDM06], [Hub01a]	This is recovery process relevant and will be considered later.
Give information about market	[Hub01a]	This is recovery process relevant and will be considered later.
Determine residual value of materials, components, or production system	[VDM06], [Hub01a]	This is recovery process relevant and will be considered later.
Assess environmental impact	[HS94]	This is recovery process relevant and will be considered later.
Consider service agreements (disposal commitments)	[Hub01a]	This is recovery process relevant and will be considered later.
Give information about disposal costs and options	[VDI02], cite17Spengler1997, cite18VDMA341602006, cite21Huber2001	This is recovery process relevant and will be considered later.

Table A.76: SD22 - Consider environmental friendliness of the system in early phases

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create an attribute *function* at the classes *production system* and *component*. Create an attribute *type* at the class *material*.

Table A.77: SD23 - Design a proper reverse logistics network

Information	Source	Representation in UML class diagram
Provide information about transportation costs and route	[SPPR97], [VDM06], [Hub01a]	This is recovery process relevant and will be considered later.
Provide information about the size of the transportable system, components, or material	-	This can be derived by the attribute *2D/3D representation* of the classes *production system* or *component*.
Provide information about warehouse capacities	[SPPR97], [Hub01a]	This is recovery process relevant and will be considered later.

Table A.78: SD24 - Reduce emissions

Information	Source	Representation in UML class diagram
Provide information about emissions	[HS94]	Create an attribute *certificate* at class *production system* and *component* (see Table A.11).

Table A.79: SD25 - Design for reusability

Information	Source	Representation in UML class diagram
Give the function of components and connection types	-	On the one hand create an attribute *function* at class *production system* and *component* (see Table A.3) and on the other hand create an attribute *type* at class *connection*.

Table A.80: SD26 - Components must meet the safety regulations that are applied for new components.

Information	Source	Representation in UML class diagram
Provide information about recovery options for the components	[SFR02], [SPPR97], [Hub01a]	This is recovery process relevant and will be considered later.

Table A.81: SD27 - Changes of the system during operation in operation phase need to be included in the documentation (which components were added, deterioration state, work load)

Information	Source	Representation in UML class diagram
Give information about quality properties of the system	[PBFG07], [SBK01]	This can be derived by the attribute *deterioration state* of the classes *production system* and *component*.
Assessment of the load of the used systems includes properties like mobility, functionality, degree of contamination, harmfulness, deterioration state, deformation state, corrosion state	[Hub01a]	This can be derived by the attributes *deterioration state* and *condition* of the classes *production system*, *component*, and *connection*.
Give information about probability that the system is used again	[Hub01a]	This is recovery process relevant and will be considered later.
Provide information about the estimated life time or useful life	[PAS04], [PBFG07], [Hub01b], [DSK+08], [OHB+13], [HS94], [KK14], [Pla16a]	This can be derived by the attribute *condition* of the classes *production system* and *component*.
Give information about the storability	[SFR02]	This is recovery process relevant and will be considered later.
Provide information about condition and life time dependent quality properties	[SFR02], [SBK01]	This can be derived by the attribute *deterioration state* and *condition* of the classes *production system*, *component*, and *connection*.

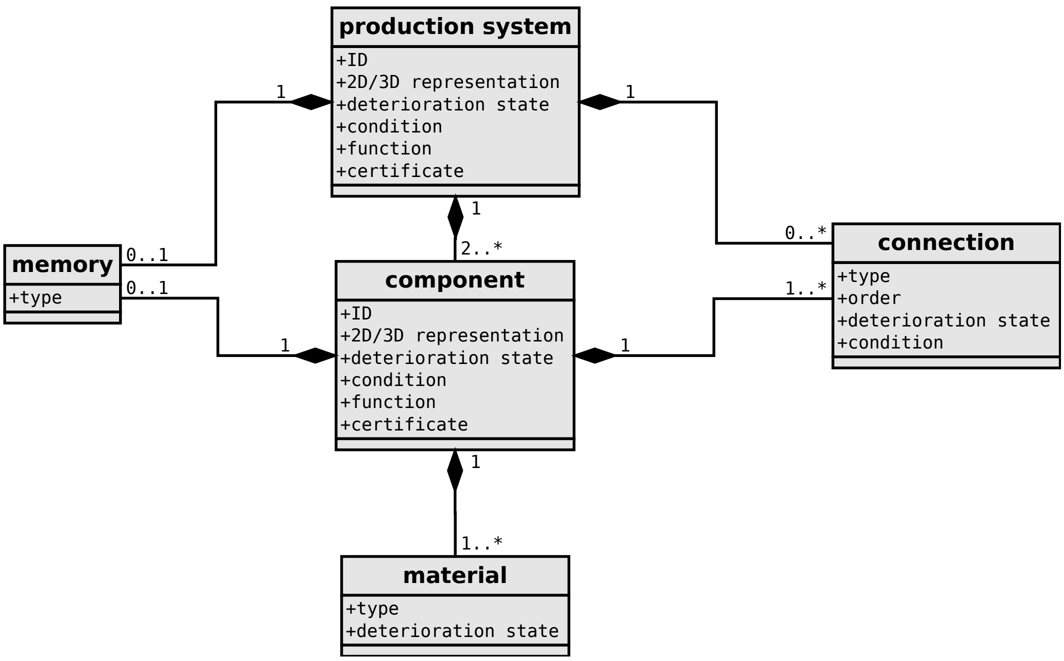

Figure A.4: Resulting UML class diagram based on requirements and information on system design and characterization

B End-of-Life process specific extensions

This annex provides the requirements and information that are particularly EoL process relevant, i.e. the feed within of artifacts (see case 18 from Figure 2.6). All 'not considered' requirements and information from Annex A are analyzed in this annex and are modeled in UML class diagrams.

B.1 System structure with EoL process view

This section provides the requirements and information about the system structure of a production system that are recovery process relevant. In the following tables only those requirements and information are listed that are recovery process relevant. Information is, then, modeled in a UML class diagram. The result for the system structure is depicted in Figure B.1.

Table B.1: SS5 - Create a reconditioning friendly structure (EoL view)

Information	**Source**	**Representation in UML class diagram**
Provide information about reconditioning options of the components	[SFR02]	Create a class *reconditioning* with the attribute *type*. This has a relation to the classes *production system* and *component*. Several classes of *reconditioning* can be associated to these classes.
Provide information about reconditioning steps and costs	[VDI02], [Hub01a]	Create a class *reconditioning* with the attributes *cost* and *technical characteristics*. The class *reconditioning* can consist of other *reconditionings*.

Table B.2: SS6 - Create a preprocessing friendly structure (EoL view)

Information	**Source**	**Representation in UML class diagram**
Provide information about preprocessing options of the components	[SFR02]	Create a class *preprocessing* with the attribute *type*. This has a relation to the classes *production system* and *component*. Several classes of *preprocessing* can be associated to these classes.
Provide information about reconditioning steps, technologies, and costs	[Ruh06], [Hub01a]	Create a class *preprocessing* with the attribute *cost* and *technical characteristics*. Create an attribute *type* at the class *preprocessing*. The class *preprocessing* can consist of other *preprocessings*.
Calculate recycling costs	[CRM12], [VDI02],[Hub01b], [VDM06], [Hub01a]	Create a class *preprocessing* with the attribute *cost*.
Provide process parameter of the preprocessing process	[Ruh06]	Create an attribute *technical characteristics* at the class *preprocessing*.
Provide information about the material and property loss due to the preprocessing process	[Ruh06]	Create a class *preprocessing* with an attribute *deterioration*.

Table B.3: SS7 - Optimize the system structure (EoL view)

Information	Source	Representation in UML class diagram
Provide information about recovery options for the components	[SFR02], [SPPR97], [Hub01a]	Create an abstract class *recovery* for the classes *reconditioning* and *preprocessing*. In the following, this will be neglected.
Provide certain instructions for the process	[PAS04]	Create an attribute *instruction* at the classes *reconditioning* and *preprocessing*.

Table B.4: SS10 - Easy identification of system and components (EoL view)

Information	Source	Representation in UML class diagram
Provide information if a component is reusable, recyclable, or needs to be disposed	[SFR02]	Create the attribute *recovery goal* at the classes *production system* and *component*. This decision-making process relevant and will be considered later.
Provide information about recycling strategy	[Hub01b]	Create a class *preprocessing* with the attributes *type* and *technical characteristics*.
Give quantity of components in inventory	[SO13]	Create a class *storage* with the attribute *type*. This has a relation to the classes *production system* and *component*. Create a class *equipment* with the attribute *stock*. This has a relation to the class *storage*.

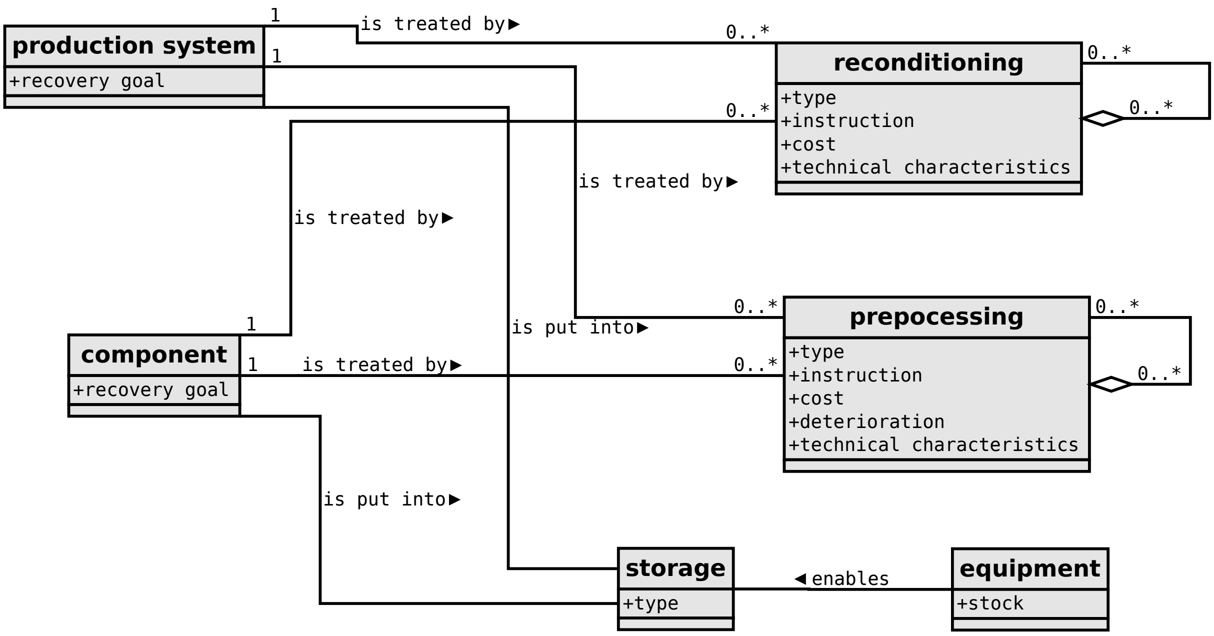

Figure B.1: Resulting UML class diagram based on requirements and information on system structure focused on End-of-Life

B.2 Connection structure with EoL process view

This section provides the requirements and information about the connection structure of a production system that are recovery process relevant. In the following tables only those requirements and information are listed that are recovery process relevant.

Information is, then, modeled in a UML class diagram. The result for the connection structure is depicted in Figure B.2.

Table B.5: CS3 - Enable nondestructive disassembly which eases recovery (EoL view)

Information	Source	Representation in UML class diagram
Give information about connections that can be undone nondestructively	-	Create a class *disassembly* with the attribute *type*.
Which components are faulty	[SFR02]	Create an attribute *recovery goal* at the classes *production system* and *component*.
Provide ratio of faulty parts after disassembling	[SFR02]	Create a class *disassembly* with the attribute *deterioration*. Create a class *component* with the attribute *recovery goal*.
Degree of quality reduction due to be process	[PBFG07], [SBK01]	Create an attribute *deterioration* at the class *disassembly*. This is relevant for characterization and will be considered later.
Which components can be recovered	[Hub01b], [Pla16a]	Create an attribute *recovery goal* at the classes *production system*, *component*, and *material*. The class *material* has a relation to the class *component*. This decision-making process relevant and will be considered later.

Table B.6: CS4 - Easy disassembly results in reduced costs (EoL view)

Information	Source	Representation in UML class diagram
Give information about how the connection can be undone	-	Create a class *disassembly* with the attribute *type*.

Table B.7: CS5 - Ensure disassembility for components (destructive, nondestructive) (EoL view)

Information	Source	Representation in UML class diagram
Give information about connections that can be undone nondestructively or destructively	-	Create a class *disassembly* with the attribute *type*.

Table B.8: CS6 - Disassembly time and costs are reduced when tool changes are low (EoL view)

Information	Source	Representation in UML class diagram
Give information about with which tool the connections can be undone	-	Create a class *equipment* with the attribute *type*. This has a relation to the class *disassembly*.

Table B.9: CS7 - Easy to access and easy to disassemble results in good replaceability (EoL view)

Information	Source	Representation in UML class diagram
Give information about the disassembly complexity	-	The class *disassembly* can consist of other *disassemblys*.
Give information about working surface for the tools	[Ruh06]	Create a class *connection* with the attribute *working surface*.

Table B.10: CS8 - Create a disassembly friendly structure (EoL view)

Information	Source	Representation in UML class diagram
Give disassembly technique	[VDI02], [Hub01b], [Ruh06], [FTM99], [RMTF99], [DSK+08], [SO13], [Hub01a], [WSDB03]	Create a class *disassembly* with the attribute *type*.
What is input and output of the process	[Hub01a]	The class *disassembly* has a relation to the class *connection* that has again a relation to the classes *production system* and *component*. Create an attribute *order* at the classes *production system* and *component*.
Give information about disassembly costs	[CRM12], [VDI02], [SFR02], [Ruh06], [DSK+08], [SPPR97], [VDM06]	Create a class *disassembly* with the attribute *cost*.

Table B.11: CS9 - Reducing disassembly time results in reduced costs (EoL view)

Information	Source	Representation in UML class diagram
Give information about applicable disassembly techniques	-	Create a class *disassembly* with the attribute *type*. Several classes of *disassembly* can be associated to the class *connection*.

Table B.12: CS10 - Proper choice of disassembly techniques dependent on the future usage of the disassembly components (EoL view)

Information	Source	Representation in UML class diagram
Give information about disassembly order and applicable disassembly techniques	[CRM12], [VDI02], [SFR02], [PBFG07], [Ruh06], [HS94], [SPPR97], [KK14], [Hub01a]	Create an attribute *disassembly order* at the class *connection* that has a relation to the class *disassembly*. Create a class *disassembly* with the attribute *type*. Several classes of *disassembly* can be associated to the class *connection*.
Give information about disassembly time	[CRM12], [VDI02], [Ruh06], [RMTF99], [HS94], [Hub01a], [WSDB03]	Create a class *disassembly* with the attribute *time*.
Provide disassembly plans	[CRM12], [VDI02], [SFR02], [RMTF99], [HS94], [SPPR97], [Hub01a]	Create an attribute *disassembly order* at the class *connection* that has a relation to the class *disassembly*. Create a class *disassembly* with the attribute *type*. The class *disassembly* can consist of other *disassembly*s.
Give information about alternative disassembly orders	[SFR02], [Hub01a]	Create an attribute *disassembly order* at the class *connection* that has a relation to the class *disassembly*. Create a class *disassembly* with the attribute *type*. The class *disassembly* can consist of other *disassembly*s. Several classes of *disassembly* can be associated to the class *connection*.
How difficult is it to undone a connection?	[RMTF99], [HS94]	Create a class *disassembly* with the attribute *difficulty*.
Give information about disassembly tooling: manual operations, partially automated, fully automated operations	[DSK+08], [SPPR97]	Create an attribute *type* at the class *equipment*.
Give information about disassembly depth	[SFR02], [DSK+08], [Hub01a]	Create an attribute *disassembly depth* at the class *production system* and *component*. Create an attribute *disassembly order* at the class *connection* that has a relation to the class *disassembly*. Create a class *disassembly* with the attribute *type*.
Provide safety instructions on the disassembly	[VDI02], [RMTF99]	Create an attribute *instruction* at the class *disassembly*.

Table B.13: CS12 - Specify connection structure (EoL view)

Information	Source	Representation in UML class diagram
Provide information about certain disassembly technique needed to undo a connection	cite02PAS10492004, [VDI02]	Create a class *disassembly* with the attribute *type*. Several classes of *disassembly* can be associated to the class *connection*.

Table B.14: CS13 - Analyze connection structure of the system (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the type of connection	[HS94]	Create a class *connection* with the attribute *technical characteristics*.

Table B.15: CS14 - Documentation of the connection types (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	Create a class *connection* with the attribute *technical characteristics*.

Table B.16: CS15 - Realize a shallow disassembly depth (EoL view)

Information	Source	Representation in UML class diagram
Give information about disassembly depth	-	Create an attribute *disassembly depth* at the class *production system* and *component*. Create an attribute *disassembly order* at the class *connection* that has a relation to the class *disassembly*. Create a class *disassembly* with the attribute *type*.

Table B.17: CS16 - Create a reassembly friendly structure (EoL view)

Information	Source	Representation in UML class diagram
Give information about reassembly time	[WSDB03]	Create the class *reassembly* with the attributes *type* and *time* that has a relation to the class *connection*.

Table B.18: CS17 - Reduce interfaces/connections (EoL view)

Information	Source	Representation in UML class diagram
Give number of connections	-	Create the classes *production system* and *component* that have a relation to the class *connection*.

Table B.19: CS18 - Specify defined connections (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	Create a class *connection* with the attribute *technical characteristics*.

Table B.20: CS19 - Usage of standard tools, avoid specialized tools - This eases the disassembly (EoL view)

Information	Source	Representation in UML class diagram
Provide information about tools needed to undo a connection	-	Create a class *equipment* with the attribute *type*. This has a relation to the class *disassembly*.

Table B.21: CS20 - Standardize connection elements that have similar or identical functions (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	Create a class *connection* with the attribute *technical characteristics*.

Table B.22: CS21 - Create easy to separate components (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the type of connection	-	Create a class *connection* with the attribute *technical characteristics*.

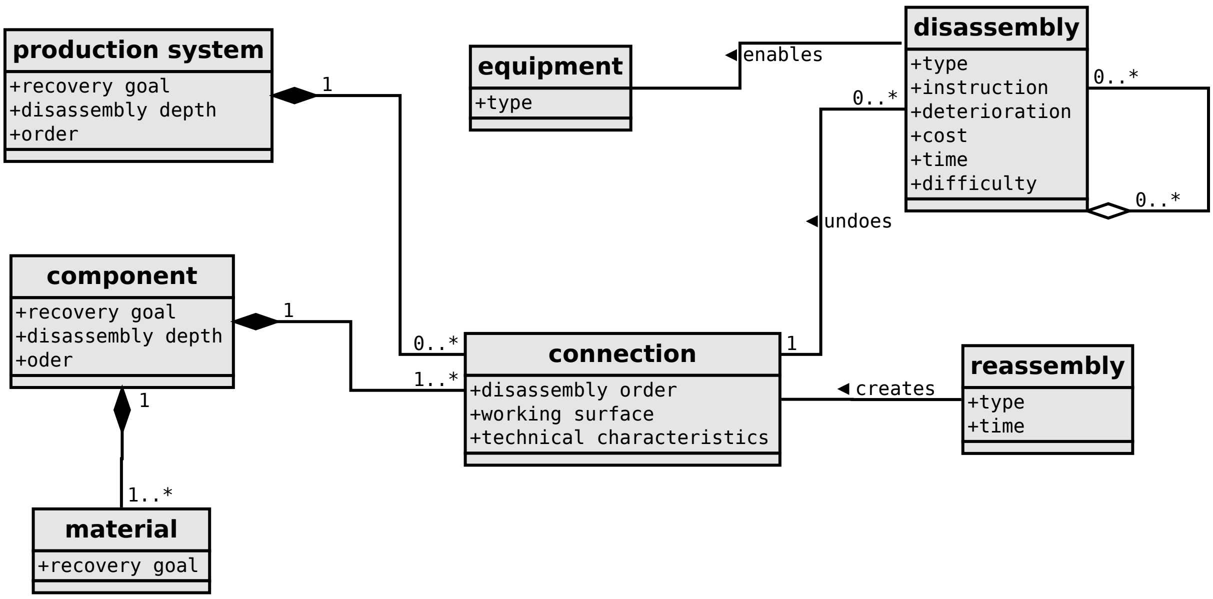

Figure B.2: Resulting UML class diagram based on requirements and information on connection structure focused on End-of-Life

B.3 Material structure with EoL process view

This section provides the requirements and information about the connection structure of a production system that are recovery process relevant. In the following tables only those requirements and information are listed that are recovery process relevant. Information is, then, modeled in a UML class diagram. The result for the connection structure is depicted in Figure B.3.

Table B.23: MS1 - Reduce harmful substances (EoL view)

Information	Source	Representation in UML class diagram
Provide information about harmful substances/material used	[CRM12], [CRM12], [PAS04], [PBFG07], [Ruh06], [Hub01a]	Create the class *material* with the attribute *harmfulness* that has a relation to the class *preprocessing*. Create the class *preprocessing* with the attribute *problematic material*.

Table B.24: MS2 - Meet the REACH standard on chemicals (EoL view)

Information	Source	Representation in UML class diagram
Identifiability of alarming substances (SVHC)	[CRM12]	Create the class *material* with the attribute *harmfulness* that has a relation to the class *preprocessing*. Create an attribute *problematic material* at the class *preprocessing*.

Table B.25: MS6 - Reduce amount of material used and reduce used material types (EoL view)

Information	Source	Representation in UML class diagram
Give information about statistical life time of raw material	[HS94]	Create the class *material* with the attribute *life expectation*.

Table B.26: MS7 - An increased amount of material results in a profitable material recovery (EoL view)

Information	Source	Representation in UML class diagram
Provide information about amount of material used	-	Create the class *material* with the attribute *weight*.

Table B.27: MS8 - Use recycling compatible materials and/or recyclable materials or components (EoL view)

Information	Source	Representation in UML class diagram
Provide more detailed information about the material used	-	Create the class *material* with the attribute *technical characteristics*.

Table B.28: MS9 - Define surface finish (e.g. varnish) (EoL view)

Information	Source	Representation in UML class diagram
Provide more detailed information about the material used	-	Create the class *material* with the attribute *technical characteristics*.

Table B.29: MS11 - Enable material recovery (EoL view)

Information	Source	Representation in UML class diagram
Provide information about valuable materials used	[CRM12], [PAS04], [PBFG07], [Hub01a]	Create the class *material* with the attribute *market value*.

Table B.30: MS12 - Use material combinations that are easy to separate (EoL view)

Information	Source	Representation in UML class diagram
Provide information about materials used	-	Create the class *material* with the attribute *technical characteristics*.

Table B.31: MS13 - Ensure sortability (EoL view)

Information	Source	Representation in UML class diagram
Provide information about sorting techniques	-	Create an attribute *type* at the class *preprocessing*.

Table B.32: MS15 - Consider recyclability of the system in the early phase of the engineering process (EoL view)

Information	Source	Representation in UML class diagram
Provide for each component a certain recycling technique	[VDI02], [SFR02], [PBFG07], [Hub01b], [Ruh06], [RMTF99], [SPPR97], [KK14]	Create an attribute *type* at the class *preprocessing* that has a relation to the classes *production system* and *component*.
Provide for each component a recycling option	[CRM12], [PAS04], [SFR02], [Ruh06]	Create an attribute *type* at the class *preprocessing* that has a relation to the classes *production system* and *component*.
How much waste is produced during recycling process?	[Ruh06], [Hub01a]	Create the class *preprocessing* with the attribute *waste generation*.
What is input and output of the process?	[Ruh06]	Create the attribute *order* at the classes *production system*, *component*, and *material*.
What is process sequence?	[CRM12], [SFR02], [Ruh06], [Hub01a]	The class *preprocessing* can consist of other *preprocessings*.

Table B.33: MS16 - Specify recycling friendly structures (EoL view)

Information	Source	Representation in UML class diagram
Provide degree of recycling	[Ruh06]	Create the class *production system*, *component*, and *material* with the attribute *degree of recovery*.
Provide information about recycling rate	[Ruh06]	Create the class *production system*, *component*, and *material* with the attribute *degree of recovery*. Create an attribute *recovery goal* at the class *material*.
Consider material upgrading possibilities	[PBFG07]	Create the class *preprocessing* with the attribute *type*, *cost*, and *revenue*.
Give recycling classes/categories	[CRM12], [PAS04], [SFR02], [Ruh06]	Create the attribute *type* at the class *preprocessing*.

Table B.34: MS17 - Ensure deformation resistance (EoL view)

Information	Source	Representation in UML class diagram
Provide information about materials used	-	Create the class *material* with the attribute *technical characteristics*.

Table B.35: MS18 - Ensure aging resistance of materials and connections (EoL view)

Information	Source	Representation in UML class diagram
Provide information about materials used	-	Create the class *material* with the attribute *technical characteristics*.

Table B.36: MS19 - No impairment of the operability of recycling plants (EoL view)

Information	Source	Representation in UML class diagram
Identification of extraneous and problematic material	[CRM12], [PAS04], [VDI02], [RMTF99]	Create the class *material* with the attribute *harmfulness* that has a relation to the class *preprocessing*. Create an attribute *problematic material* at the class *preprocessing*.
Specify classes of hazardous substances	[SFR02]	Create the class *material* with the attribute *harmfulness* that has a relation to the class *preprocessing*. Create an attribute *problematic material* at the class *preprocessing*.
Hazardous substances according to which directive/law etc.?	[PAS04]	Create an attribute *recovery goal* at class *material*. Create an attribute *regulation* at class *material*.
Provide information about capacity constraints (capacity restrictions and costs)	[SPPR97]	Create the class *equipment* with the attribute *capacity*. Create the class *preprocessing* with the attribute *cost*.
Identification of material that is critical for environmental and recycling	[VDI02]	Create the class *material* with the attribute *harmfulness* that has a relation to the class *preprocessing*. Create an attribute *problematic material* at the class *preprocessing*.

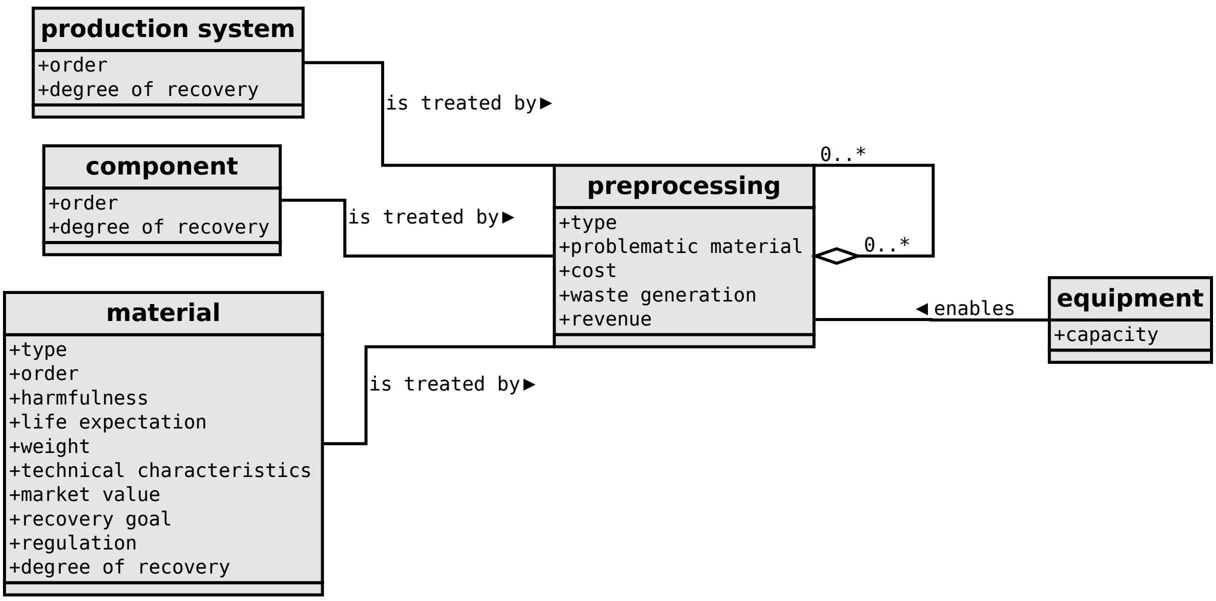

Figure B.3: Resulting UML class diagram based on requirements and information on material structure focused on End-of-Life

B.4 System design and characterization with EoL process view

This section provides the requirements and information about the design and characterization of a production system that are recovery process relevant. In the following tables only those requirements and information are listed that are recovery process relevant. Information is, then, modeled in a UML class diagram. The result for the system design and characterization is depicted in Figure B.4.

Table B.37: SD1 - System must be digitally cleanable (protection of know-how) (EoL view)

Information	Source	Representation in UML class diagram
Provide information about how the system can be digitally cleaned	-	Create a class *memory* with an attribute *type* that has a relation to the classes classes *production system* and *component*. Create a class *cleaning* with an attribute *type* that has a relation to the classes *production system*, *component*, and *memory*.

Table B.38: SD3 - Enable depollution and material recovery (EoL view)

Information	Source	Representation in UML class diagram
Assign recycling activities to components or the system	[CRM12], [SFR02], [Ruh06]	Create a class *preprocessing* with the attribute *type*. This has a relation to the classes *production system*, *component*, and *material*. Several classes of *preprocessing* can be associated to the classes *production system* and *component*.
Assign recycling techniques to components or the system	[FTM99], [DSK+08], [Ruh06]	Create a class *preprocessing* with the attribute *type*. This has a relation to the classes *production system*, *component*, and *material*. Several classes of *preprocessing* can be associated to the classes *production system*, *component*, and *material*.
Give information about the total amount/weight of a certain material or percentage share	[CRM12], [PAS04], [VDI02], [Ruh06], [RMTF99], [HS94], [KK14], [Hub01a]	Create the class *material* with the attribute *weight* that has a relation to the class *component*. Create an attribute *weight* at the classes *production system* and *component*.
Give information about the quantitative relations of single components	[SFR02]	Create the class *material* with the attribute *weight* that has a relation to the class *component*. Create an attribute *weight* at the classes *production system* and *component*.
Provide information about the size	[CRM12], [PAS04], [Ruh06],[RMTF99]	Create an attribute *size* at the classes *production system* and *component*.
Provide information about the costs for the final disposal	[DSK+08]	Create a class *disposal* with the attribute *type* and *cost* that has a relation to the classes *production system*, *component*, and *material*.
Provide information about special tools necessary to use	[PAS04], [VDI02], [Ruh06]	Create a class *disassembly* with the attribute *type* that has a relation to the class *connection*. Create a class *equipment* with the attribute *type* that has a relation to the class *disassembly*.
Provide information about disassembly technologies	[RMTF99], [Hub01a]	Create a class *disassembly* with the attribute *type* that has a relation to the class *connection*.
Which statutory provisions need to be met (prohibited materials, obligatory disassembly activities, limitation of energetic recovery)?	[Ruh06]	Create a class *production system*, *component*, and *material* with the attribute *regulation*. Create the class *equipment* with the attribute *regulation*.

Table B.39: SD9 - Operating liquids within the system must be drained easy and fast (EoL view).

Information	Source	Representation in UML class diagram
Provide information about relevant connections/interfaces	-	Create the class *connection* with the attribute *technical characteristics*.

Table B.40: SD12 - Evaluation of suitability for further use phases (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create the attribute *market value* at the classes *production system*, *component*, and *material*. This decision-making process relevant and will be considered later.

Table B.41: SD13 - Components must be capable of being reused in the next generation of the system. This requires compatibility of the components (EoL view).

Information	Source	Representation in UML class diagram
Provide information about recovery options for the system or components for identical or different functions	[PBFG07]	Create an abstract class *recovery* for the classes *cleaning*, *disposal*, *disassembly*, and *preprocessing*. In the following, this will be neglected.

Table B.42: SD15 - Enable reconditioning (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create a class *decommissioning* with an attribute *type* that has a relation to the classes *production system* and *component*. Create a class *reconditioning* with an attribute *type* that has a relation to the classes *production system* and *component*. Create a class *commissioning* with an attribute *type* that has a relation to the classes *production system* and *component*.

Table B.43: SD16 - Enable repair-ability (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create a class *repair* with an attribute *type* that has a relation to the classes *production system* and *component*.

Table B.44: SD17 - Ensure clean-ability, e.g. by avoiding to design gaps (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create a class *cleaning* with an attribute *type* that has a relation to the classes *production system* and *component*.

Table B.45: SD18 - Ensure testability, e.g. check deterioration state (EoL view)

Information	Source	Representation in UML class diagram
Provide information about the system and components	-	Create a class *testing* with an attribute *type* that has a relation to the classes *production system* and *component*.

Table B.46: SD20 - Enable the decommissioning (EoL view)

Information	Source	Representation in UML class diagram
Provide information about decommissioning costs	[VDM06]	Create a class *decommissioning* with an attribute *type* that has a relation to the classes *production system* and *component*.

Table B.47: SD21 - Avoid to have costly process step afterwards (EoL view)

Information	**Source**	**Representation in UML class diagram**
Provide certain instructions to depollute fast and easy	-	Create an attribute *instructions* at the classes *reconditioning*, *preprocessing*, and *disassembly*.
Provide information about the working time required	[CRM12], [VDI02]	Create the classes *reconditioning*, *preprocessing*, and *disassembly* with an attribute *time*.
Provide information about the capacities of the waste disposal contractor	[SPPR97], [Hub01a]	Create the class *equipment* with an attribute *capacity* that has a relation to the class *disposal*.
Economic assessment of the recycling processes including disassembly, separation, preprocessing, and logistics	[VDI02], [SBK01]	Create a class *transportation* with an attribute *type* that has a relation to *production system*, *component*, and *material*. This is relevant for characterization and will be considered later.
Provide information about status of operating capacity of the disassembly facilities	[SO13]	Create the class *equipment* with an attribute *capacity* that has a relation to the class *disassembly*.
Give information about commodity market fluctuations	[SO13]	Create the classes *production system*, *component*, and *material* with an attribute *market value*.
Identify target fraction of the recycler	[CRM12]	Create the class *equipment* with the attributes *revenue* and *cost*.
Give information about the target fraction for each component	[VDI02], [Ruh06]	Create the classes *component* and *material* each with an attribute *recovery goal*.
Give information about process parameters (e.g. degree of purity of fractions)	[VDI02], [Ruh06]	Create the class *preprocessing* with an attribute *technical characteristics*.
Give information about indivisible materials	[Ruh06]	Create the class *material* with the attribute *type* that has a relation to the class *preprocessing*. Create an attribute *problematic material* at the class *preprocessing*.
Identify target valuable material and non-target material	[CT13]	Create the class *material* with an attribute *recovery goal*.
Provide information about reconstruction costs	[VDM06]	Create an class *commissioning* with an attribute *type* that has a relation to the classes *production system* and *component*. Create the class *commissioning* with an attribute *cost*.
Provide guided benchmarking including clearly defined KPIs (e.g. energy efficiency)	[SBSW12]	This decision-making process relevant and will be considered later.
Provide information about costs and revenues	[PAS04], [Ruh06], [RMTF99], [DSK+08], [HS94], [SPPR97], [VDM06], [Hub01a]	This is relevant for characterization and will be considered later.
Give information about market	[Hub01a]	This decision-making process relevant and will be considered later.
Determine residual value of materials, components, or production system	[VDM06], [Hub01a]	Create the attribute *market value* at the classes *production system*, *component*, *material*. This decision-making process relevant and will be considered later.
Assess environmental impact	[HS94]	This is relevant for characterization and will be considered later.
Consider service agreements (disposal commitments)	[Hub01a]	Create a class *equipment* with an attribute *capacity*.
Give information about disposal costs and options	[VDI02], cite17Spengler1997, cite18VDMA341602006, cite21Huber2001	Create a class *disposal* with the attributes *type* and *cost*. Several classes of *disposal* can be associated to the classes *production system*, *component*, and *material*.

Table B.48: SD23 - Design a proper reverse logistics network (EoL view)

Information	**Source**	**Representation in UML class diagram**
Provide information about transportation costs and route	[SPPR97], [VDM06], [Hub01a]	Create an attribute *type* and *cost* at the class *transportation*. Create an attribute *location designation* at the class *equipment*.
Provide information about the size of the transportable system, components, or material	-	Create the classes *production system*, *component*, and *material* each with an attribute *size*. Create the class *equipment* with an attribute *size*.
Provide information about warehouse capacities	[SPPR97], [Hub01a]	Create a class *storage* with an attribute *type* that has a relation to the classes *production system* and *component*. Create a class *equipment* with an attribute *capacity* that has a relation to the class *storage*.

Table B.49: SD26 - Components must meet the safety regulations that are applied for new components (EoL view).

Information	**Source**	**Representation in UML class diagram**
Provide information about regulations for the components	[SFR02], [SPPR97], [Hub01a]	Create the attribute *regulation* at the classes *production system* and *component*.

Table B.50: SD27 - Changes of the system during operation in operation phase need to be included in the documentation (which components were added, deterioration state, work load) (EoL view)

Information	**Source**	**Representation in UML class diagram**
Give information about quality properties of the system	[PBFG07], [SBK01]	Create the classes *production system* and *component* each with an attribute *market value*.
Assessment of the load of the used systems includes properties like mobility, functionality, degree of contamination, harmfulness, deterioration state, deformation state, corrosion state	[Hub01a]	This can be derived by the attributes *deterioration state* and *condition* of the classes *production system*, *component*, and *connection*.
Give information about probability that the system is used again	[Hub01a]	This decision-making process relevant and will be considered later.
Provide information about the estimated life time or useful life	[PAS04], [PBFG07], [Hub01b], [DSK+08], [OHB+13], [HS94], [KK14], [Pla16a]	Create the classes *production system* and *component* each with an attribute *life expectation*.
Give information about the storability	[SFR02]	Create a class *storage* with an attribute *type* that has a relation to the classes *production system* and *component*.
Provide information about condition and life time dependent quality properties	[SFR02], [SBK01]	This can be derived by the attribute *deterioration state* and *condition* of the classes *production system*, *component*, and *connection*. And this can be derived by the attribute *life expectation* of the classes *production system*, *component*, and *material*.

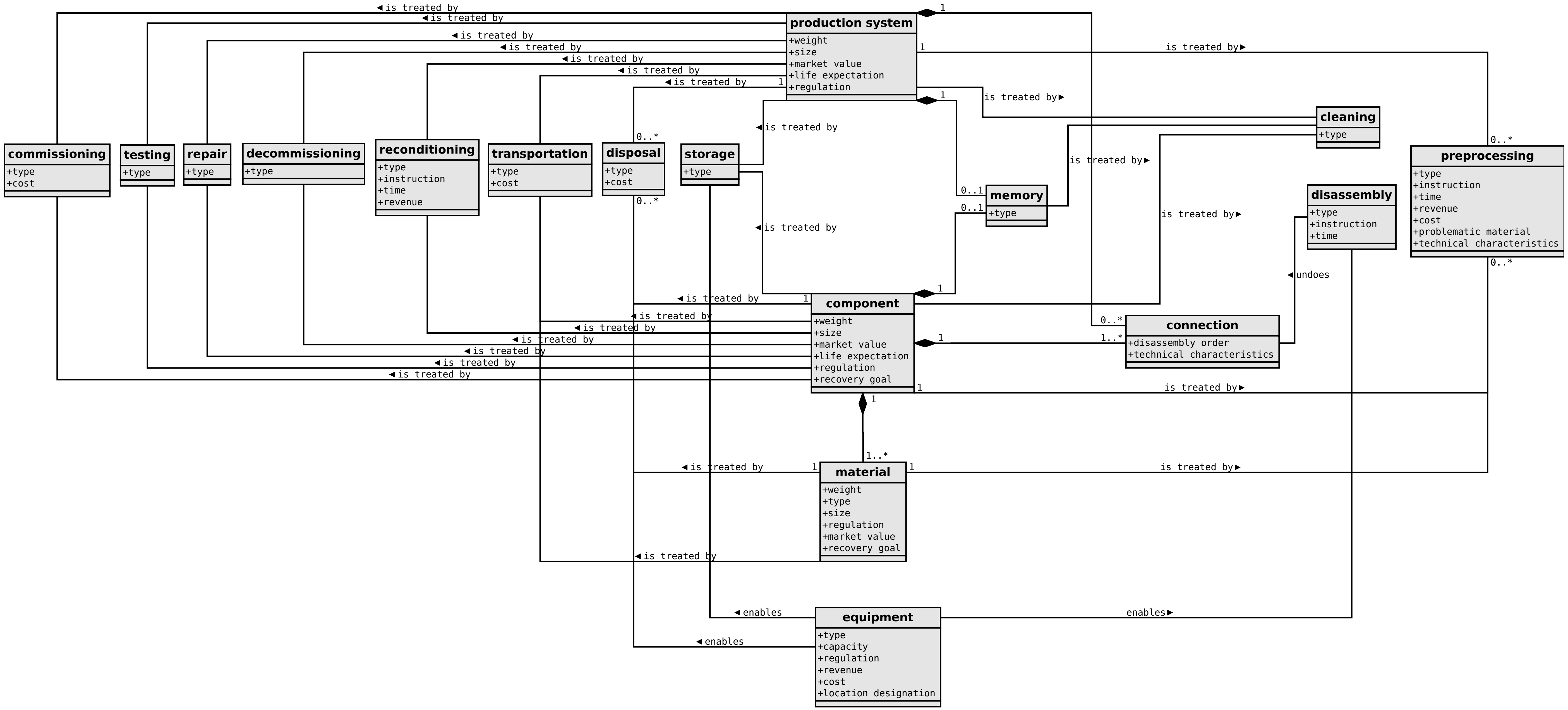

Figure B.4: Resulting UML class diagram based on requirements and information on system design and characterization focused on End-of-Life

C Requirements on software tool and data exchange

The literature review mentioned in Chapter 3 resulted, besides the EoL data model, also in a list of requirements on software support. These are listed in the following table, but this is not an exhaustive overview. It is also indicated whether a requirement applies to the software tool functionality or to the data exchange between software tools.

Table C.1: Requirements on software (x: applies to, - not applies to) and how AutomationML meets the requirements

No.	Requirements	Source	Software tool	Data exchange	AutomationML's capabilities
RS1	Protection of product and company secrets and know-how	[PAS04]	x	x	XML Encryption and Signature.
RS2	Integrability into the company's internal and external IT landscape	[PAS04]	x	x	Standardized in IEC 62714.
RS3	Create a simple and consistent data structure	[PAS04]	x	x	Syntax is standardized in IEC 62714-1.
RS4	Easy and fast provision of the requested data	[PAS04], [Ruh06]	x	-	-
RS5	Consider relevant security aspects and access authorization	[PAS04]	x	x	XML Encryption and Signature.
RS6	Provide change history	[PAS04]	x	x	Meta information on file (WriterHeader) and object level (ChangeMode, Version, etc.) available.
RS7	User friendly design of the documentation for the usage by service and recycling	[VDI02], [PBFG07], [Ruh06]	x	-	-
RS8	Automatic data generation	[SFR02]	x	-	-
RS9	Mapping of the model to the target format	[SFR02]	x	x	Done in this thesis.
RS10	Storage of all necessary data in a consistent manner	[SFR02], [Ruh06]	x	x	Syntax is set. But semantics is flexible. Proper semantics will be developed in this thesis.
RS11	Enable the modeling of modules	[OHB$^+$13]	x	x	Object oriented modeling approach allows the modeling of scalable objects.
RS12	Extensible	[VDM06], [KK14], [SBSW12]	x	x	It is extensible in terms of semantics an in terms of integrating other external documents.
RS13	Scalable	[VDM06]	x	x	Object oriented modeling approach allows the modeling of scalable objects. The flexible semantics defines the meaning of an object.
RS14	Domain model has to carry information in form of various data pattern	[KK14], [SBK01]	x	x	Syntax is standardized in IEC 62714-1. Semantics is flexible and will be developed in this thesis.
RS15	Handle data from different and distributed sources	[KK14], [SBK01]	x	x	Standardized in IEC 62714.
RS16	Handle data from an abstract until a concrete level	[KK14]	x	x	Object oriented modeling approach allows the modeling of scalable objects. The flexible semantics defines the meaning of an object.
RS17	Make information available in a unified format	[KK14]	x	x	Standardized in IEC 62714.
RS18	Liability of data exchange	[SBSW12]	x	-	-
RS19	Standardized description of products and components	[SBK01]	x	-	Object oriented modeling approach allows the modeling of scalable objects. The flexible semantics defines the meaning of an object.
RS20	Automation of plausibility checks	[SBSW12]	x	-	-
RS21	Comprehensive support of data collection processes as well as automation and support of data preparation	[SBSW12]	x	-	-

D Description of case studies

In this annex the different case studies are described. It contains a short description of the company itself and what the company is doing in the scope of this thesis (namely the recovery of production systems). But all is anonymized and generalized.

D.1 Case study I: plant engineering company

This case study analyzes a company that is specialized in plant engineering as a service. This company engineers, installs, commissions production systems for product producing companies (mainly for the discrete manufacturing industry), hereafter, called customers. The company also moves the production systems, which they have planned, from one location to another location on behalf of the customer. The relocation of the production system can be done in the same factory, because of layout changes. Or it can be done from one country to another, because of strategical or economic reasons. While relocating, the production system can also be enhanced, adapted, or modernized according to customer's requirements. Thus, the company has access to the engineering data from the engineering phase. During the operation phase, the company has no access to the operational data (like wear and tear, repair, or maintenance data). This data needs to be gathered before relocation.
Usually, the company does the relocation for its regular customers. However, in some cases, the company does a relocation also for customers, whose production systems the company has not planned. In that case, the company needs to gather also the engineering data besides the operational data.

In Figure D.1 the recovery process for production systems is visualized. The interviewee was using his experience of several years to construct a general process comprising 12 activities. Each activity is then further described from Table D.1 to Table D.12. The times are highly project dependent so that they are left blank.

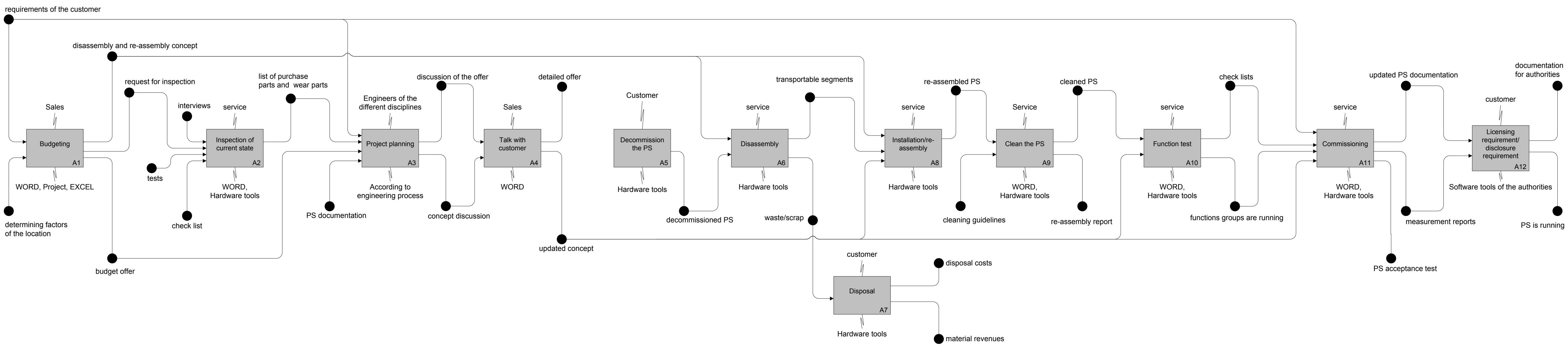

Figure D.1: Process description of case study I: production system relocation / recovery

Table D.1: A1) Budgeting

activity	A1) Budgeting
description	Customer gets in touch with the company (request) because he wants to relocate a PS (relocating the PS in one and the factory or across countries. Together with the relocation, the PS can also get modernized, e.g. to increase throughput.). Customer wants to know what this relocation would cost. Usually, the estimated costs are up to +/-10% exact. Disassembly concept is developed.
time	It starts 1-2 years before. Budgeting takes two days to two weeks - depending on the project scope.
predecessor activity	-
successor activity	A2)
involved experts	sales
involved tools (software and hardware)	MS WORD, MS Project, MS EXCEL
required artifacts	1) requirements of the customer
	2) determining factors of the location
description of artifacts	1) What does the customer want to modernize/adapt: new requirements/PS parameters are discussed and documented together (MS WORD).
	2) How do the determining factors of both locations look like: Which circumstances/conditions need to be met?
created artifacts	1) disassembly and re-assembly concept
	2) budget offer
	3) request for inspection
description of artifacts	1) Disassembly and re-assembly concept is developed. The effect for doing this is 50% higher than for a new PS because every part needs to be handled two times (disassembly and re-assembly). For new PS there is only the assembly. (MS Project, Gantt charts, EXCEL bar diagrams)
	2) Budget offer is up to +/-10% accurate. (MS WORD)
	3) -

Table D.2: A2) Inspection of current state

activity	A2) Inspection of current state
description	Purchase parts and wear parts are recommended.
time	After four weeks the customer is whether accepting the budget offer or rejecting it. Service needs four weeks only for the inspection. Service needs eight weeks for inspection and the document/service writes a report. The current state (as-is state) is inspected and a recommendation for purchase and wear parts is given. Service gives this to spare parts delivery and gets an offer.
predecessor activity	A1)
successor activity	A3)
involved experts	service
involved tools (software and hardware)	MS WORD, hardware tools
required artifacts	1) request for inspection
	2) interviews
	3) tests
	4) check list
description of artifacts	1) -
	2) Interviews: The service discusses the current state of the PS with the maintenance personnel of the PS owner.
	3) Tests: The parts are inspected when the PS is operating and when the PS is paused.
	4) Check lists: Either the service needs to inspect each and every part or only certain parts. For this, a check list exists. However, the company has the rule, that each and every part is inspected.
created artifacts	1) list of purchase parts and wear parts
description of artifacts	1) List that contains the recommended purchase and wear parts. (MS WORD)

Table D.3: A3) Project planning

activity	A3) Project planning
description	After identifying the purchase and wear parts the new concepts for the PS are developed. The aim is to replace as few as possible and as much as necessary. It is decided which parts are replaced, which are remanufactured, which are send back to the manufacturer and get manufactured, and which parts are used again as they are.
time	It takes four weeks to develop the concept. Afterwards, the final approval of the customer is needed.
predecessor activity	A2)
successor activity	A4)
involved experts	Engineers of the different disciplines
involved tools (software and hardware)	According to engineering process
required artifacts	1) list of purchase parts and wear parts
	2) budget offer
	3) requirements of the customer
	4) PS documentation
description of artifacts	1) List that contains the recommended purchase and wear parts. (MS WORD)
	2) Budget offer is up to +/-10% accurate. (MS WORD)
	3) What does the customer want to modernize/adapt: new requirements/PS parameters are discussed and documented together (MS WORD).
	4) PS documentation: This is old stock of CAD data (Microstation) - when it is a regular customer. The company needs to store the engineering data in a data base for 10 years - usually PDF. PS documentation comes are paper or digital as PDF on a data storage device.
created artifacts	1) discussion of the offer
	2) concept discussion
description of artifacts	1) The offer is discussed together with the customer.
	2) Position plans, process design, hall layout plans (CAD), foundation plans are adapted to costumer's needs/requirements as well as the actions the customer needs to do on site (The company defines where a pit must be and the customer needs to realize this.)

Table D.4: A4) Talk with customer

activity	A4) Talk with customer
description	The detailed offer is discussed. At that point, sales do not know whether parts are broken or not. That's why the offer is per se a bit higher. New requirements on the PS are discussed. In case of process changes, a new concept is developed. Customer approves spare parts.
time	It starts 3-4 months before the disassembly. It takes 1-2 days.
predecessor activity	A3)
successor activity	A5)
involved experts	sales
involved tools (software and hardware)	MS WORD
required artifacts	1) discussion of the offer
	2) concept discussion
description of artifacts	1) The offer is discussed together with the customer.
	2) Position plans, process design, hall layout plans (CAD), foundation plans are adapted to costumer's needs/requirements as well as the actions the customer needs to do on site (The company defines where a pit must be and the customer needs to realize this.)
created artifacts	1) detailed offer (MS WORD).
	2) updated concept
description of artifacts	1) -
	2) -

Table D.5: A5) Decommission the PS

activity	A5) Decommission the PS
description	Decommissioning of the PS is done by the customer.
time	-
predecessor activity	A4)
successor activity	A6)
involved experts	customer
involved tools (software and hardware)	hardware tools
required artifacts	1) -
description of artifacts	1) -
created artifacts	1) decommissioned PS
description of artifacts	1) -

Table D.6: A6) Disassembly

activity	**A6) Disassembly**
description	Position numbers are used to label the segments. The order needs to be considered, so that the transportation of the segments can be schedule and organized. In case the segments are transported through sea freight (e.g. to China), all the segments are transported at once with one container. It is tried to fit the segments into the standard containers. In case the segments are transported by trucks (e.g. within Europe), a continuous truck traffic ('truck convey') is organized. Disassembly order does not correspond with assembly order. Cleaning during disassembly is possible - This depends on the current state of the PS and would be ordered by the customer.
time	Ratio regarding to necessary time: 1/3 disassembly, 2/3 re-assembly + test. This depends on the PS size and the determining factors of the location.
predecessor activity	A5)
successor activity	A7) and A8)
involved experts	service
involved tools (software and hardware)	hardware tools
required artifacts	1) disassembly and re-assembly concept
	2) decommissioned PS
description of artifacts	1) Disassembly and re-assembly concept. The size of the building/factory needs to be considered since the segments need to fit through doors etc. Additionally, the load size needs to be considered (ship/truck).
	2) -
created artifacts	1) transportable segments
	2) waste/scrap
description of artifacts	1) -
	2) Waste is everything that stays in the factory and is not removed to the new location.

Table D.7: A7) Disposal

activity	**A7) Disposal**
description	Waste is everything that stays in the factory and is not removed to the new location. The customer takes care of the disposal of the waste due to monetary reasons. Usually, the customer knows the location/area/city/region/country better than the company. So, knows the best appropriate recycling companies. So, the customer is responsible for the disposal.
time	-
predecessor activity	A7)
successor activity	-
involved experts	customer
involved tools (software and hardware)	hardware tools
required artifacts	1) waste/scrap
description of artifacts	1) Waste is everything that stays in the factory and is not removed to the new location.
created artifacts	1) Disposal costs
	2) Material revenues
description of artifacts	1) -
	2) -

Table D.8: A8) Installation/re-assembly

activity	A8) Installation/re-assembly
description	Parts are re-assembly on site.
time	Ratio regarding to necessary time: 1/3 disassembly, 2/3 re-assembly + test. This depends on the PS size and the determining factors of the location.
predecessor activity	A6)
successor activity	A9)
involved experts	service
involved tools (software and hardware)	hardware tools
required artifacts	1) transportable segments
	2) disassembly and re-assembly concept
	3) updated concept
description of artifacts	1) -
	2) According to the updated plant layout. Disassembly and re-assembly concept. The size of the building/factory needs to be considered since the segments need to fit through doors etc. Additionally, the load size needs to be considered (ship/truck).
	3) -
created artifacts	1) re-assembled PS
description of artifacts	1) -

Table D.9: A9) Clean the PS

activity	A9) Clean the PS
description	The PS is cleaned by using cleansers. There are the following cleaning options available: primary cleaning (clean swept) and precision cleaning. It is defined by the company what is cleaned when. The company, however, only executed the primary cleaning. Cleaning by the customer during the disassembly is possible. It depends on the current state of the PS.
time	Ratio regarding to necessary time: 1/3 disassembly, 2/3 re-assembly + test. This depends on the PS size and the determining factors of the location.
predecessor activity	A8)
successor activity	A10)
involved experts	service (plans only the primary cleaning)
involved tools (software and hardware)	MS WORD, hardware tools
required artifacts	1) cleaning guidelines
	2) re-assembled PS
description of artifacts	1) There are cleaning guidelines. There are the following cleaning options available: primary cleaning (clean swept) and precision cleaning. It is defined by the company what is cleaned when. The company, however, only executed the primary cleaning.
	2) -
created artifacts	1) cleaned PS
	2) assembly report
description of artifacts	1) -
	2) The assembly report is filled out during the visual test/check. (MS WORD)

Table D.10: A10) Function test

activity	**A10) Function test**
description	Each part, each unit, each function group needs to operate properly, e.g. each fan must operate. This is tested at the control cabinet. This is crucial for the quality. A safety test is done.
time	Ratio regarding to necessary time: 1/3 disassembly, 2/3 re-assembly + test. This depends on the PS size and the determining factors of the location.
predecessor activity	A9)
successor activity	A11)
involved experts	service
involved tools (software and hardware)	MS WORD, hardware tools
required artifacts	1) cleaned PS
	2) updated concept
description of artifacts	1) -
	2) -
created artifacts	1) check lists (MS WORD)
	2) functions groups are running
description of artifacts	1) -
	2) -

Table D.11: A11) Commissioning

activity	A11) Commissioning
description	Defined PS functions are checked. The different modes of the PS are tested and checked (different routes (logistics), turn-off times). Process parameter are measured and documented. Measurements like effluent limit of sewage, exhaust air limit, are checked. Those parameters are controlled by local authorities. In the end, the PS is operating.
time	Ratio regarding to necessary time: 1/3 disassembly, 2/3 re-assembly + test. This depends on the PS size and the determining factors of the location.
predecessor activity	A10)
successor activity	A12)
involved experts	service
involved tools (software and hardware)	MS WORD, hardware tools
required artifacts	1) check lists
	2) functions groups are running
	3) requirements of the customer
	4) updated concept
description of artifacts	1) -
	2) -
	3) Requirements of the customer are verified.
	4) -
created artifacts	1) updated PS documentation
	2) measurement reports
	3) PS acceptance test
description of artifacts	1) Updated PS documentation is handed over to the customer (this was initially created during project planning and was kept up to date during the entire project - until now.)
	2) -
	3) This is a document that is signed by the company and the customer in order to confirm the correctness, functionality, and sold scope of delivery as well as to finish the project.

Table D.12: A12) Licensing requirement/ disclosure requirement

activity	**A12) Licensing requirement/ disclosure requirement**
description	Process parameters/values are measured and documented. Measurements like effluent limit of sewage, exhaust air limit, are checked. Those parameters are controlled by local authorities. In the end, the PS is operating.
time	-
predecessor activity	A11)
successor activity	-
involved experts	customer
involved tools (software and hardware)	Software tools of the authorities
required artifacts	1) measurement reports
	2) updated PS documentation
description of artifacts	1) -
	2) From the updated PS documentation the plant layout is in particular needed.
created artifacts	1) documentation for authorities
	2) PS is running
description of artifacts	1) Fill out documents for authorities.
	2) -

D.2 Case study II: demolition company

This case study analyzes a company that is specialized in decommissioning, disassembling, demolition, recycling, and disposal of buildings (and all what is left in the building , e.g. production systems) as a service. The company offers its services to municipalities but also to product producing companies, hereafter, called customers. The company is capable to remove parts of a building or remove the building until there is nothing left but grassland. The company recycles some of the materials by itself, e.g. concrete, that is shredded to gravel and sand. What remains is sold to external recycling companies and disposal contractors. Before the demolition and recycling the company needs to gather data from the engineering as well as from the operation phase.
An important fact is that the building is decommissioned by the building owner himself and not by the company.

In Figure D.2 the recovery process for the material the production systems are made of is visualized. The interviewee was using his experience of several years to construct a general process comprising seven activities. Each activity is then further described from Table D.13 to Table D.19.

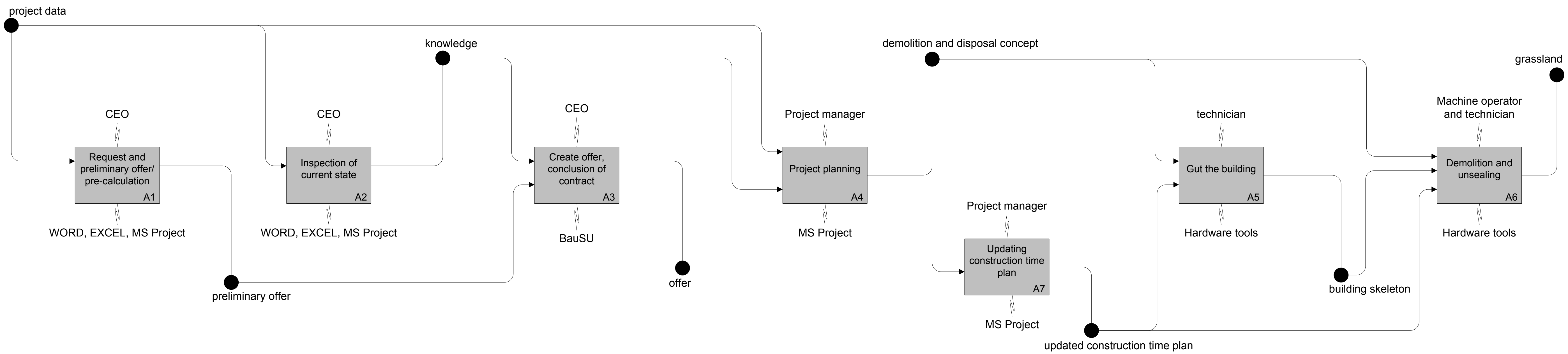

Figure D.2: Process description of case study II: material recovery

Table D.13: A1) Request and preliminary offer/pre-calculation

activity	**A1) Request and preliminary offer/pre-calculation**
description	-
time	Two months before the project starts. It takes one hour.
predecessor activity	-
successor activity	A2)
involved experts	CEO
involved tools (software and hardware)	MS WORD, MS ECXEL, MS project
required artifacts	1) Project data
description of artifacts	1) Project data (as PDF) contains the documentation and also the harmful substance list. The building owner provides this to the company. This provision is of monetary relevance of the building owner. Because without it the company would need to assume that asbestos is everywhere in the building. This would increase the costs for demolition tremendously.
created artifacts	1) Preliminary offer
description of artifacts	1) The preliminary offer is a package deal. Costs and material revenues are already considered.

Table D.14: A2) Inspection of current state

activity	**A2) Inspection of current state**
description	It is a visual inspection on site of the project data. Does the send project data match? Usually, the project data is quite complete.
time	After preliminary offer. Before project start. It takes two months.
predecessor activity	A1)
successor activity	A3)
involved experts	CEO
involved tools (software and hardware)	MS Office
required artifacts	1) Project data
description of artifacts	1) Project data (as PDF) contains the documentation and also the harmful substance list. The building owner provides this to the company. This provision is of monetary relevance of the building owner. Because without it the company would need to assume that asbestos is everywhere in the building. This would increase the costs for demolition tremendously.
created artifacts	1) knowledge
description of artifacts	1) Is the project data correct? First ideas are gathered regarding demolition and disposal concept.

Table D.15: A3) Create offer, conclusion of contract

activity	A3) Create offer, conclusion of contract
description	A contract is concluded between building owner and the company.
time	This is the project start. One month after the conclusion of the contract the construction site starts.
predecessor activity	A2)
successor activity	A4)
involved experts	CEO
involved tools (software and hardware)	BauSU (for creating offer)
required artifacts	1) preliminary offer
	2) knowledge
description of artifacts	1) The preliminary offer is a package deal. Costs and material revenues are already considered.
	2) Is the project data correct? First ideas are gathered regarding demolition and disposal concept.
created artifacts	1) offer
description of artifacts	1) offer (as PDF with necessary signatures)

Table D.16: A4) Project planning

activity	A4) Project planning
description	The demolition and disposal concept is developed during the project planning: planning of the personnel placement and development of the construction time plan. Therefore, the project data is evaluated. In a discussion of the CEO with the project manager, the knowledge of the CEO is shared with the project manager.
time	It starts with project start. It takes two weeks.
predecessor activity	A3)
successor activity	A5) and A7)
involved experts	Project manager
involved tools (software and hardware)	MS project
required artifacts	1) knowledge
	2) Project data
description of artifacts	1) Is the project data correct? First ideas are gathered regarding demolition and disposal concept.
	2) Project data (as PDF) contains the documentation and also the harmful substance list. The building owner provides this to the company. This provision is of monetary relevance of the building owner. Because without it the company would need to assume that asbestos is everywhere in the building. This would increase the costs for demolition tremendously.
created artifacts	1) demolition and disposal concept
description of artifacts	1) It comprises the planning of the personnel placement and development of the construction time plan (When is which construction waste disposed?).

Table D.17: A5) Gut the building

activity	A5) Gut the building
description	To gut the building everything needs to be removed (floorings, electrical systems etc.), until there is nothing left but the building skeleton.
time	To gut the building, 30% of the overall time is needed (referred to the duration: construction site start until construction site end)
predecessor activity	A4)
successor activity	A6)
involved experts	Technician
involved tools (software and hardware)	hardware tools
required artifacts	1) demolition and disposal concept
description of artifacts	1) It comprises the planning of the personnel placement and development of the construction time plan (When is which construction waste disposed?).
created artifacts	1) Building skeleton
description of artifacts	1) The building skeleton is only the building shell and the stonework.

Table D.18: A6) Demolition and unsealing

activity	A6) Demolition and unsealing
description	The demolition comprises activities of the pure demolition as well as the sorting of materials and the transportation of the construction waste. Additionally, an unsealing is done. Thereby, the foundation slab is removed (the 'seal' of the foundation). This activity is assigned to 'sorting of materials', because the soil is also sieved. Diggers tear down and sort the metal. Company's own trucks driving to the construction site and transport everything either to the company's place or to the landfill.
time	30% of the overall time is needed for the demolition (referred to the duration: construction site start until construction site end). 10% of the overall time is needed for the sorting of materials (referred to the duration: construction site start until construction site end). 20% of the overall time is needed for the transportation of the construction waste (referred to the duration: construction site start until construction site end).
predecessor activity	A5)
successor activity	-
involved experts	Machine operator and technician.Machine operators are doing the demolition. The technicians gather the extraneous material by hand which occurs during the demolition.
involved tools (software and hardware)	hardware tools
required artifacts	1) demolition and disposal concept
	2) Building skeleton
description of artifacts	1) It comprises the planning of the personnel placement and development of the construction time plan (When is which construction waste disposed?).
	2) The building skeleton is only the building shell and the stonework.
created artifacts	1) grassland
description of artifacts	1) Building is torn down.

Table D.19: A7) Updating construction time plan

activity	A7) Updating construction time plan
description	The construction time plan, which was initially developed during the project planning, is updated permanently.
time	It is done in parallel.
predecessor activity	A4)
successor activity	-
involved experts	Project manager
involved tools (software and hardware)	MS Project
required artifacts	1) demolition and disposal concept
description of artifacts	1) It comprises the planning of the personnel placement and development of the construction time plan (When is which construction waste disposed?).
created artifacts	1) updated construction time plan
description of artifacts	1) -

D.3 Case study IIIa and IIIb: component/equipment manufacturer

This case study analyzes a company that is specialized in component/equipment manufacturing as well as in systems and solutions for electrical engineering, electronics, and automation. The company itself engineers, installs, commissions their own production systems for the production of the company's products. The company also moves the production systems from one location to another location. The relocation of the production system can be done in the same factory, because of layout changes. Or it can be done from one country to another, because of strategical or economic reasons. While relocating, the production system can also be enhanced, adapted, or modernized due to changed requirements. Thus, the company has access to the engineering data from the engineering phase. Since the company also operates the production systems it has access to the operational data (like wear and tear, repair, or maintenance data).
Besides the relocation of production systems within the company (case study CS **IIIa**), the company also recycles its own production systems together with an external recycling company (case study CS **IIIb**). The company is doing the decommissioning and disassembling, so that the production system fits onto the truck. The external recycling company transports and does the actual disassembling and recycling of the production systems (also together with scrap dealers and disposal contractors). It is not indented to resell the production system again because of know-how protection.
The company is doing case study IIIb (material recovery) more often than IIIa (production system recovery), since the production systems are operating so many years (more than 10 years) that it is not profitable to reuse them again. However, the wear and tear state is hard to define.

In Figure D.3 the recovery process for production systems is visualized. Each activity is then further described from Table D.20 to Table D.38. The times are highly project dependent so that they are left blank.

In Figure D.4 the recovery process for the material the production systems is made of is visualized. Each activity is then further described from Table D.39 to Table D.50. The times are highly project dependent so that they are left blank.

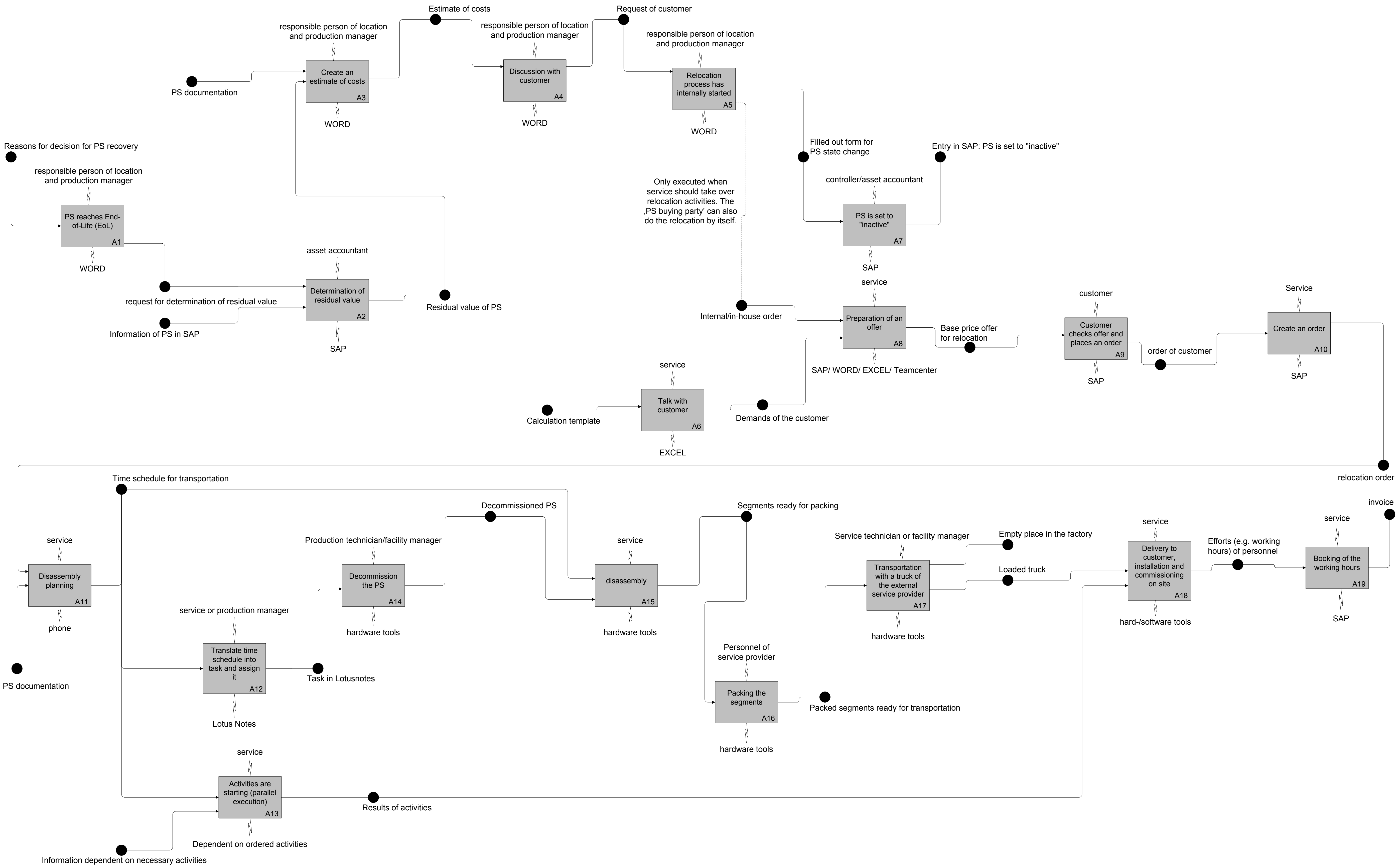

Figure D.3: Process description of case study IIIa: production system relocation / recovery

Table D.20: A1) PS reaches End-of-Life (EoL)

activity	A1) PS reaches End-of-Life (EoL)
description	Production manager, together with the responsible person of location, decide that the PS is going to be set to 'inactive' because the PS has reached its end of life for a reason, e.g. production will be relocated, and needs to be relocated.
time	-
predecessor activity	-
successor activity	A2)
involved experts	responsible person of location and production manager
involved tools (software and hardware)	MS WORD (to create PDF)
required artifacts	1) Reasons for decision for PS recovery
description of artifacts	1) -
created artifacts	1) request for determination of residual value
description of artifacts	1) Request is done through phone/email.

Table D.21: A2) Determination of residual value

activity	A2) Determination of residual value
description	Asset accountant determines the residual value of the PS by taking the fair value into account. Profit assessment: In case the PS is completely written off, then it is all profit. In case the PS has a residual value, the 'buying party/customer' needs to pay that residual value.
time	-
predecessor activity	A1)
successor activity	A3)
involved experts	asset accountant
involved tools (software and hardware)	SAP (indirectly): all information are stored in SAP. Personnel need to aggregate and interpret them appropriately.
required artifacts	1) request for determination of residual value
	2) Information of PS in SAP
description of artifacts	1) Request is done through phone/email.
	2) When was the PS built? How much did it cost (purchase price)? What is the cost center? Which equipment number has the PS? To which workplace was the PS assigned to? Which product was produced by this PS?
created artifacts	1) Residual value of PS
description of artifacts	1) The purchase price is considered and if the PS is already written off, i.e. the fair value is considered.

Table D.22: A3) Create an estimate of costs

activity	A3) Create an estimate of costs
description	Here the PS is considered and evaluated based on an inspection, which activities are reasonable, which spare parts are not in stock anymore etc. Optimization potential is identified. Therefore, expert's knowledge/experience is used - what are the weak points of the PS. Service does bigger repairs. Production technicians do the smaller repairs by themselves. That's why the service does not know for sure the actual as-is-state of the PS.
time	Inspection takes a week.
predecessor activity	A2)
successor activity	A4)
involved experts	responsible person of location and production manager
involved tools (software and hardware)	MS WORD
required artifacts	1) Residual value of PS
	2) PS documentation
description of artifacts	1) The purchase price is considered and if the PS is already written off, i.e. the fair value is considered.
	2) It contains the workplace designation incl. purchase parts label (PDF).
created artifacts	1) Estimate of costs
description of artifacts	1) Estimate of costs is send through email.

Table D.23: A4) Discussion with customer

activity	A4) Discussion with customer
description	Does the estimate of costs matches the expectations of the customer? What are the conditions the customer would buy the PS?
time	-
predecessor activity	A3)
successor activity	A5)
involved experts	responsible person of location and production manager
involved tools (software and hardware)	MS WORD
required artifacts	1) Estimate of costs
description of artifacts	1) Estimate of costs is send through email.
created artifacts	1) Request of customer
description of artifacts	1) Noncommittal request: what would a relocation of the PS mean?

Table D.24: A5) Relocation process has internally started

activity	A5) Relocation process has internally started
description	Internal/in-house order is in this case optional. This is only executed when service should take over relocation activities. The 'PS buying party/customer' can also do the relocation by itself. Chronologically, 'Filled out form for PS state change' is created before 'Internal/in-house order'.
time	-
predecessor activity	A4)
successor activity	A7) and A8) (A8) is optional, dependent on the customer's requirements)
involved experts	responsible person of location and production manager
involved tools (software and hardware)	MS WORD
required artifacts	1) Request of customer
description of artifacts	1) Noncommittal request: what would a relocation of the PS mean?
created artifacts	1) Filled out form for PS state change (containing information about relocating, selling, recycling, decommissioning a PS)
	2) Internal/in-house order
description of artifacts	1) Documents the decision which EoL scenario was chosen for the PS.
	2) With this, the service gets the order to do all necessary steps for relocating the PS.

Table D.25: A6) Talk with customer

activity	A6) Talk with customer
description	Service meets up with customer to discuss details. Develop optimization options: Depending on the budget and requirements of the customer, the costs are calculated, e.g. Installation incl. commissioning: yes/no (1 week), Decommissioning and disassembly on site: yes/no (duration 1-2 days), Training of personnel: yes/no, Inspection wanted? Recommendations and the necessary rework can be developed better if an inspection took place.
time	-
predecessor activity	-
successor activity	A8)
involved experts	service
involved tools (software and hardware)	MS EXCEL
required artifacts	1) Calculation template: What can the service offer?: Calculation template (filled out EXCEL)
description of artifacts	1) What can the service offer?: calculation template (filled out EXCEL)
created artifacts	1) Demands of the customer
description of artifacts	1) Demands/requirements of the customer are written down by using the calculation template (filled out EXCEL - from that the customer can choose (check or not check an option)).

Table D.26: A7) PS is set to 'inactive'

activity	A7) PS is set to 'inactive'
description	PS is set to 'inactive' within the asset accounting.
time	-
predecessor activity	A5)
successor activity	-
involved experts	controller/asset accountant
involved tools (software and hardware)	SAP (all PS are stored and maintained in SAP.)
required artifacts	1) Filled out form for PS state change
description of artifacts	1) Documents the decision which EoL scenario was chosen for the PS.
created artifacts	1) Entry in SAP: PS is set to 'inactive'
description of artifacts	1) It is a kind of trigger. No production orders will be assigned to this resource (PS). With this resource no costs can be generated. Resource gets inactive.

Table D.27: A8) Preparation of an offer

activity	A8) Preparation of an offer
description	An offer is prepared based on the options the customer had chosen.
time	-
predecessor activity	A5) and A6)
successor activity	A9)
involved experts	service
involved tools (software and hardware)	SAP, Teamcenter, MS WORD, MS EXCEL
required artifacts	1) Internal/in-house order
	2) Demands of the customer
description of artifacts	1) With this, the service gets the order to do all necessary steps for relocating the PS.
	2) Demands/requirements of the customer are written down by using the calculation template (filled out EXCEL - from that the customer can choose (check or not check an option)).
created artifacts	1) Base price offer for relocation
description of artifacts	1) What is defined in the scope of delivery/scope of service? The price is only a base price. In the end the efforts are summed up and an invoice is send.

Table D.28: A9) Customer checks offer and places an order

activity	A9) Customer checks offer and places an order
description	Does the offer match the expectations of the customer?
time	-
predecessor activity	A8)
successor activity	A10)
involved experts	customer
involved tools (software and hardware)	SAP
required artifacts	1) Base price offer for relocation
description of artifacts	1) What is defined in the scope of delivery/scope of service? The price is only a base price. In the end the efforts are summed up and an invoice is send.
created artifacts	1) order of customer
description of artifacts	1) Order is send through email.

Table D.29: A10) Create an order

activity	A10) Create an order
description	The location sends an official letter - with order number.
time	-
predecessor activity	A9)
successor activity	A11)
involved experts	service
involved tools (software and hardware)	SAP
required artifacts	1) order of customer
description of artifacts	1) Order is send through email.
created artifacts	1) relocation order
description of artifacts	1) It documents the current information about the PS: Who is responsible for the PS, responsible cost center, responsible technician, who initiated the relocation, which products were produced on the PS.

Table D.30: A11) Disassembly planning

activity	A11) Disassembly planning
description	Service and responsible person of location coordinate this. Transportation is planned either by service or by the customer. Necessary PS documentation and workflows are dependent on the project. Time schedule is also always coordinated with company's personnel.
time	-
predecessor activity	A10)
successor activity	A12) and A13)
involved experts	service
involved tools (software and hardware)	phone
required artifacts	1) PS documentation
	2) relocation order
description of artifacts	1) PS documentation is dependent on the project.
	2) It documents the current information about the PS: Who is responsible for the PS, responsible cost center, responsible technician, who initiated the relocation, which products were produced on the PS.
created artifacts	1) Time schedule for transportation
description of artifacts	1) What needs to be done when?

Table D.31: A12) Translate time schedule into task and assign it

activity	A12) Translate time schedule into task and assign it
description	Service or production manager informs the technicians about when the PS needs to be decommissioned (via phone or email).
time	-
predecessor activity	A11)
successor activity	A14)
involved experts	service or production manager
involved tools (software and hardware)	Lotus Notes
required artifacts	1) Time schedule for transportation
description of artifacts	1) What needs to be done when?
created artifacts	1) Task in Lotusnotes
description of artifacts	1) Energy, air and water needs to be disconnected.

Table D.32: A13) Activities are starting (parallel execution)

activity	A13) Activities are starting (parallel execution)
description	Dependent of the options the customer had ordered (e.g. training of the personnel, translation of the PS documentation) the activities are executed.
time	-
predecessor activity	A11)
successor activity	A18)
involved experts	service
involved tools (software and hardware)	Dependent on ordered activities
required artifacts	1) Time schedule for transportation
	2) Information dependent on necessary activities
description of artifacts	1) What needs to be done when?
	2) These are dependent on customer's order.
created artifacts	1) Results of activities
description of artifacts	1) It represents that, what was ordered by the customer.

Table D.33: A14) Decommission the PS

activity	A14) Decommission the PS
description	Together with the technicians the PS is decommissioned. Only technicians can disconnect water. Air and energy do also need to be disconnected.
time	It takes one day.
predecessor activity	A12)
successor activity	A15)
involved experts	Production technician/facility manager
involved tools (software and hardware)	hardware tools
required artifacts	1) Task in Lotusnotes
description of artifacts	1) Energy, air and water needs to be disconnected.
created artifacts	1) Decommissioned PS
description of artifacts	1) PS cannot produce anymore.

Table D.34: A15) disassembly

activity	A15) disassembly
description	Disassemble the PS into transportable segments. Usually the PS can be decomposed into machine tables by undoing screws. All of those connections are screw connections which increases the disassembly friendliness. Extensions of the PS will be also disassembled, like hand-work place, place for insertion, conveyor technique, packaging table - everything which 'stick out' of the PS. Service technicians know the PS - it is likely that they have maintained it. Disassembly graph is not necessary, therefore. Also an inspection for capturing the as-is-state is not necessary.
time	-
predecessor activity	A14)
successor activity	A16)
involved experts	service technician (mechanician, electrician, programmer)
involved tools (software and hardware)	hardware tools
required artifacts	1) Time schedule for transportation
	2) Decommissioned PS
description of artifacts	1) What needs to be done when?
	2) PS cannot produce anymore.
created artifacts	1) Segments ready for packing
description of artifacts	1) The size of the to be packed segments depends on how the PS is transported (sea or air) Packing is done by an external service provider.

Table D.35: A16) Packing the segments

activity	A16) Packing the segments
description	The packing of the segments are done by an external service provider.
time	It takes one day.
predecessor activity	A15)
successor activity	A17)
involved experts	Personnel of service provider
involved tools (software and hardware)	hardware tools
required artifacts	1) Segments ready for packing
description of artifacts	1) The size of the to be packed segments depends on how the PS is transported (sea or air) Packing is done by an external service provider.
created artifacts	1) Packed segments ready for transportation
description of artifacts	1) PS is disassembled into transportable segments - so that they fit onto the truck.

Table D.36: A17) Transportation with a truck of the external service provider

activity	A17) Transportation with a truck of the external service provider
description	Segments are transported by a truck of another external service provider. Sea freight takes 6 weeks but is cheaper then air freight, which is faster.
time	It takes one day.
predecessor activity	A16)
successor activity	A18)
involved experts	Service technician or facility manager. Service technicians load the segments onto the external service provider's truck. Except the segments are too huge or heavy and a forklift truck is needed, then the production technicians need to take over.
involved tools (software and hardware)	hardware tools
required artifacts	1) Packed segments ready for transportation
description of artifacts	1) PS is disassembled into transportable segments - so that they fit onto the truck.
created artifacts	1) Empty place in the factory
	2) Loaded truck
description of artifacts	1) It was made room for a now PS.
	2) Truck was loaded.

Table D.37: A18) Delivery to customer, installation and commissioning on site

activity	A18) Delivery to customer, installation and commissioning on site
description	Service personnel travels with the PS, if it was ordered by the customer.
time	-
predecessor activity	A17)
successor activity	A19)
involved experts	service
involved tools (software and hardware)	Dependent on ordered activities. hard-/software tools.
required artifacts	1) Results of activities
	2) Loaded truck
description of artifacts	1) It represents that, what was ordered by the customer.
	2) Truck was loaded.
created artifacts	1) Efforts (e.g. working hours) of personnel
description of artifacts	1) Efforts of the personnel on site.

Table D.38: A19) Booking of the working hours

activity	**A19) Booking of the working hours**
description	Service personnel document their effects/working hours. Based on this, the final invoice can be created afterwards.
time	-
predecessor activity	A18)
successor activity	-
involved experts	service
involved tools (software and hardware)	SAP
required artifacts	1) Efforts (e.g. working hours) of personnel
description of artifacts	1) Efforts of the personnel on site.
created artifacts	1) invoice
description of artifacts	1) Customer gets the final invoice.

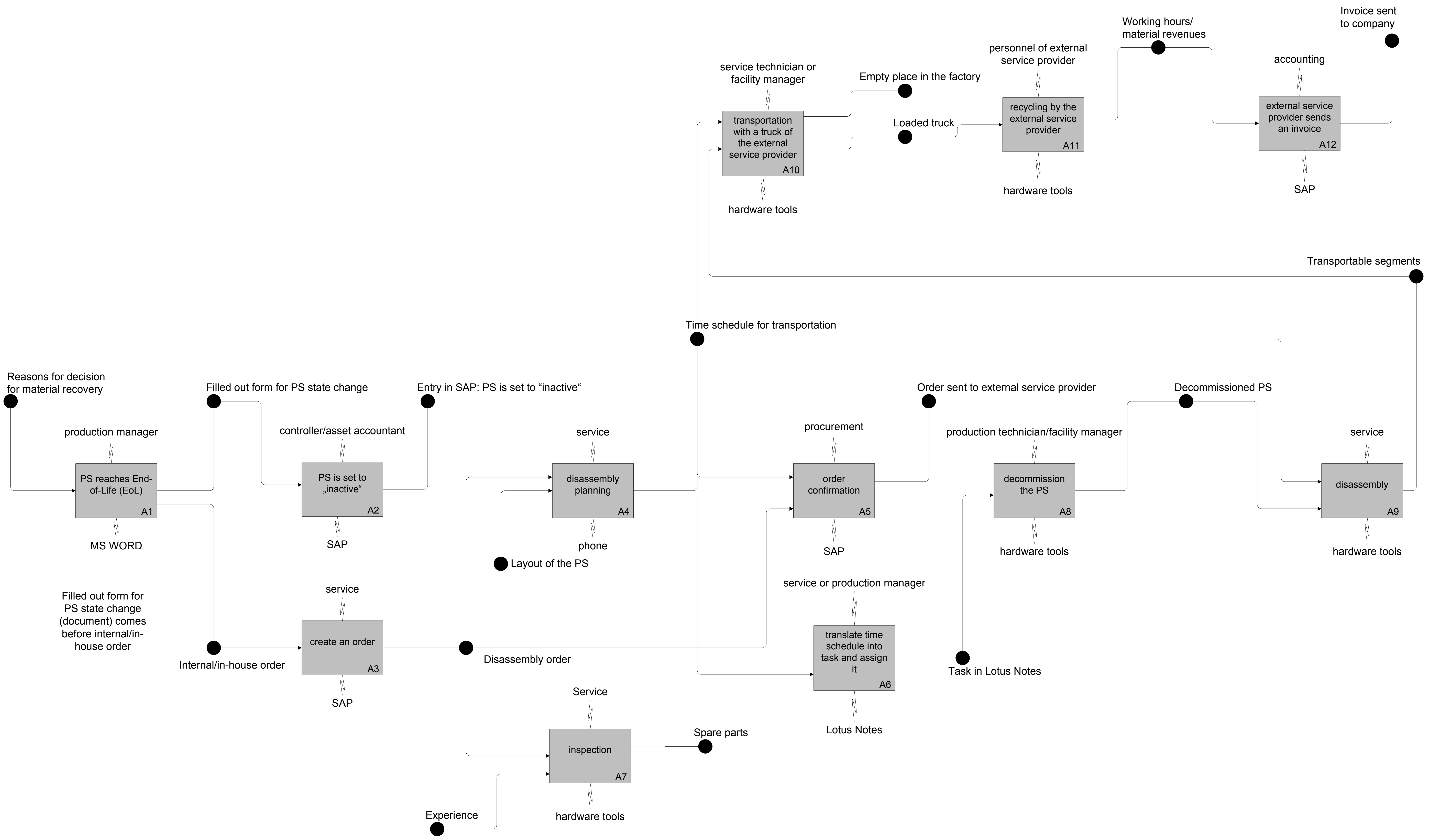

Figure D.4: Process description of case study IIIb: material recovery

Table D.39: A1) PS reaches End-of-Life (EoL)

activity	A1) PS reaches End-of-Life (EoL)
description	Production manager decides that the PS is going to be set to 'inactive' because the PS has reached its end of life for a reason, e.g. production will be relocated, and needs to be recycled.
time	-
predecessor activity	-
successor activity	A2) and A3)
involved experts	production manager
involved tools (software and hardware)	MS WORD (to create PDF)
required artifacts	1) reasons for decision for material recovery
description of artifacts	1) -
created artifacts	1) Filled out form for PS state change (containing information about relocating, selling, recycling, decommissioning a PS)
	2) Internal/in-house order
description of artifacts	1) Documents the decision which EoL scenario was chosen for the PS.
	2) With this, the service gets the order to do all necessary steps for material recovery.

Table D.40: A2) PS is set to inactive

activity	A2) PS is set to 'inactive'
description	PS is set to 'inactive' within the asset accounting.
time	-
predecessor activity	A1)
successor activity	-
involved experts	controller/asset accountant
involved tools (software and hardware)	SAP (all PS are listed in SAP)
required artifacts	1) Filled out form for PS state change (containing information about relocating, selling, recycling, decommissioning a PS)
description of artifacts	1) Documents the decision which EoL scenario was chosen for the PS.
created artifacts	1) Entry in SAP: PS is set to 'inactive'
description of artifacts	1) It is a kind of trigger. No production orders will be assigned to this resource (PS). With this resource no costs can be generated. Resource gets inactive.

Table D.41: A3) create an order

activity	A3) create an order
description	Service gets the information to disassemble the PS. They are asking for an offer to recycle the PS.
time	-
predecessor activity	A1)
successor activity	A4) and A7)
involved experts	service
involved tools (software and hardware)	SAP
required artifacts	1) Internal/in-house order
description of artifacts	1) With this, the service gets the order to do all necessary steps for material recovery.
created artifacts	1) Disassembly order (is sent via email)
description of artifacts	1) It documents the current information about the PS: Who is responsible for the PS, responsible cost center, responsible technician, who initiated the recycling, which products were produced on the PS.

Table D.42: A4) disassembly planning

activity	A4) disassembly planning
description	Service coordinates the transportation with the external service provider (when is what transported). External service provider schedules the transportation. Time schedule is also always coordinated with company's personnel.
time	Organization is dependent on the company's capacity. External service provider is at the company every week.
predecessor activity	A3)
successor activity	A5) and A6)
involved experts	service
involved tools (software and hardware)	phone
required artifacts	1) Layout of the PS
	2) Disassembly order
description of artifacts	1) Layout of the PS is used. It is a MCAD drawing printed out as sheet of paper or PDF. With this, the size can be determined. Service knows, how the PS needs to be disassembled: Into machine tables with their extension. Which size does the machine have - Does it fit onto the truck? The truck belongs to the external service provider.
	2) It documents the current information about the PS: Who is responsible for the PS, responsible cost center, responsible technician, who initiated the recycling, which products were produced on the PS.
created artifacts	1) Time schedule for transportation
description of artifacts	1) Decision is made whether the PS can be transported this week, in case there is cargo area left on the truck (weekly operating truck). Extra tours are possible, but are avoided. Moving the PS out of the factory and onto the truck is done together with the company. Truck size in general and what cargo area is available at a certain day are taking into account.

Table D.43: A5) order confirmation

activity	A5) order confirmation
description	Order is created by service. Order is places by procurement for the external service provider.
time	-
predecessor activity	A3)
successor activity	-
involved experts	procurement
involved tools (software and hardware)	Order is confirmed via SAP. (Then, the external service provider gets the order via email.)
required artifacts	1) Time schedule for transportation
	2) Disassembly order
description of artifacts	1) Decision is made whether the PS can be transported this week, in case there is cargo area left on the truck (weekly operating truck). Extra tours are possible, but are avoided. Moving the PS out of the factory and onto the truck is done together with the company. Truck size in general and what cargo area is available at a certain day are taking into account.
	2) It documents the current information about the PS: Who is responsible for the PS, responsible cost center, responsible technician, who initiated the recycling, which products were produced on the PS.
created artifacts	1) Order sent to external service provider
description of artifacts	1) It contains information about who is the coordinator at company's side. The service is the man in the middle. Service is between the external service provider and the company's production. Service talks to the production and evaluated a good disassembly and pick-up time for the production system.

Table D.44: A6) translate time schedule into task and assign it

activity	A6) translate time schedule into task and assign it
description	Service or production manager informs the technicians about when the PS needs to be decommissioned (via phone or email).
time	-
predecessor activity	A4)
successor activity	A8)
involved experts	service or production manager
involved tools (software and hardware)	Lotus Notes
required artifacts	1) Time schedule for transportation
description of artifacts	1) Decision is made whether the PS can be transported this week, in case there is cargo area left on the truck (weekly operating truck). Extra tours are possible, but are avoided. Moving the PS out of the factory and onto the truck is done together with the company. Truck size in general and what cargo area is available at a certain day are taking into account.
created artifacts	1) Task in Lotus Notes: When needs what to be disconnected.
description of artifacts	1) Energy, air and water needs to be disconnected.

Table D.45: A7) inspection

activity	A7) inspection
description	Inspection whether parts of the PS can be used again. Are these parts of which the company ran out of spare parts? Otherwise new parts are preferred.
time	As soon as the PS was set to 'inactive' the inspection can begin. PS does not necessarily need to be decommissioned.
predecessor activity	A3)
successor activity	-
involved experts	production technician (mechanician, electrician, programmer)
involved tools (software and hardware)	Hardware tools
required artifacts	1) Disassembly order
	2) Experience
description of artifacts	1) Layout of the PS is used. It is a MCAD drawing printed out as sheet of paper or PDF. With this, the size can be determined. Service knows, how the PS needs to be disassembled: Into machine tables with their extension. Which size does the machine have - Does it fit onto the truck? The truck belongs to the external service provider.
	2) It is inspected what parts the PS have. PS are operating for a long time. Spare parts are possible not available anymore. Technician knows what PS with which parts are still operating and which could possible benefit from some extra spare parts from this decommissioned PS.
created artifacts	1) Spare parts
description of artifacts	1) Spare parts, which are not produced anymore. But double stockkeeping is avoided.

Table D.46: A8) decommission the PS

activity	A8) decommission the PS
description	Together with the technicians the PS is decommissioned. Only technicians can disconnect water. Air and energy do also need to be disconnected.
time	Duration: 1 day
predecessor activity	A6)
successor activity	A9)
involved experts	production technician/facility manager
involved tools (software and hardware)	Hardware tools
required artifacts	1) Task in Lotus Notes
description of artifacts	1) Energy, air and water needs to be disconnected.
created artifacts	1) Decommissioned PS
description of artifacts	1) PS cannot produce anymore.

Table D.47: A9) disassembly

activity	A9) disassembly
description	Disassemble the PS into transportable segments. Usually the PS can be decomposed into machine tables by undoing screws. All of those connections are screw connections which increases the disassembly friendliness. Extensions of the PS will be also disassembled, like hand-work place, place for insertion, conveyor technique, packaging table - everything which 'stick out' of the PS. Service technicians know the PS - it is likely that they have maintained it. Disassembly graph is not necessary, therefore. Also an inspection for capturing the as-is-state is not necessary.
time	-
predecessor activity	A8)
successor activity	A10)
involved experts	service technician (mechanician, electrician, programmer)
involved tools (software and hardware)	Hardware tools
required artifacts	1) Time schedule for transportation
	2) Decommissioned PS
description of artifacts	1) Decision is made whether the PS can be transported this week, in case there is cargo area left on the truck (weekly operating truck). Extra tours are possible, but are avoided. Moving the PS out of the factory and onto the truck is done together with the company. Truck size in general and what cargo area is available at a certain day are taking into account.
	2) PS cannot produce anymore.
created artifacts	1) Transportable segments
description of artifacts	1) PS is disassembled into transportable segments - so that they fit onto the truck.

Table D.48: A10) transportation with a truck of the external service provider

activity	A10) transportation with a truck of the external service provider
description	Segments are transported by a truck of the external service provider.
time	Duration max. 1 day
predecessor activity	A9)
successor activity	A11)
involved experts	service technician or facility manager (Service technicians load the segments onto the external service provider's truck. Except the segments are too huge or heavy and a forklift truck is needed, then the production technicians need to take over.)
involved tools (software and hardware)	Hardware tools
required artifacts	1) Transportable segments
	2) Time schedule for transportation
description of artifacts	1) PS is disassembled into transportable segments - so that they fit onto the truck.
	2) Decision is made whether the PS can be transported this week, in case there is cargo area left on the truck (weekly operating truck). Extra tours are possible, but are avoided. Moving the PS out of the factory and onto the truck is done together with the company. Truck size in general and what cargo area is available at a certain day are taking into account.
created artifacts	1) Empty place in the factory
	2) Loaded truck
description of artifacts	1) It was made room for a now PS.
	2) Truck was loaded.

Table D.49: A11) scrapping by the external service provider

activity	A11) scrapping by the external service provider
description	The external service provider disassembles the segments for recycling. The recovered materials are brought to a nearby scrap dealer.
time	-
predecessor activity	A10)
successor activity	A12)
involved experts	personnel of external service provider
involved tools (software and hardware)	Hardware tools
required artifacts	1) Loaded truck
description of artifacts	1) Truck was loaded.
created artifacts	1) Working hours/material revenues
description of artifacts	1) How long did it take the external service provider to disassembly the segments for recycling? And how much revenue did they get for the recovered material.

Table D.50: A12) external service provider sends an invoice

activity	A12) external service provider sends an invoice
description	External service provider creates an EXCEL sheet in which all the segments/components are listed as well as the material they have brought to the scrap dealer.
time	After the recycling, the external service provider sends an invoice.
predecessor activity	A11)
successor activity	-
involved experts	accounting
involved tools (software and hardware)	SAP
required artifacts	1) Working hours/material revenues
description of artifacts	1) How long did it take the external service provider to disassembly the segments for recycling? And how much revenue did they get for the recovered material.
created artifacts	1) Invoice sent to company
description of artifacts	1) The invoice is sent via letter or email to the company. Invoice is checked and accepted.

D.4 Case study IV: manufacturer of products and consumables

This case study analyzes a company that is specialized in the production and distribution of (medical) products and consumables for those products. They company is capable of engineering, installing, and commissioning their own production systems for the production of the company's products and consumables. On the other hand the company involves external plant engineering companies. When the production system type is out of its scope of work and requires special knowledge. In case of the decommissioning of a production system of the company, components/equipment from that production systems are used again for planning another, new production system of the company. While relocating the components/equipment, the components/equipment can also be enhanced, adapted, or modernized due to changed requirements of the new production system. Thus, the company has access to the engineering data of the components/equipment from the engineering phase. Since the company also operates the production systems it has access to the operational data (like wear and tear, repair, or maintenance data).

In Figure D.5 the recovery process for a machine from a production system that has reached its end of life is visualized. The interviewee had a specific project in mind when he was interviewed. The process description for this specific project contains eight activities. Each activity is then further described from Table D.51 to Table D.58. He was able to give details about times and duration having this specific project in mind.

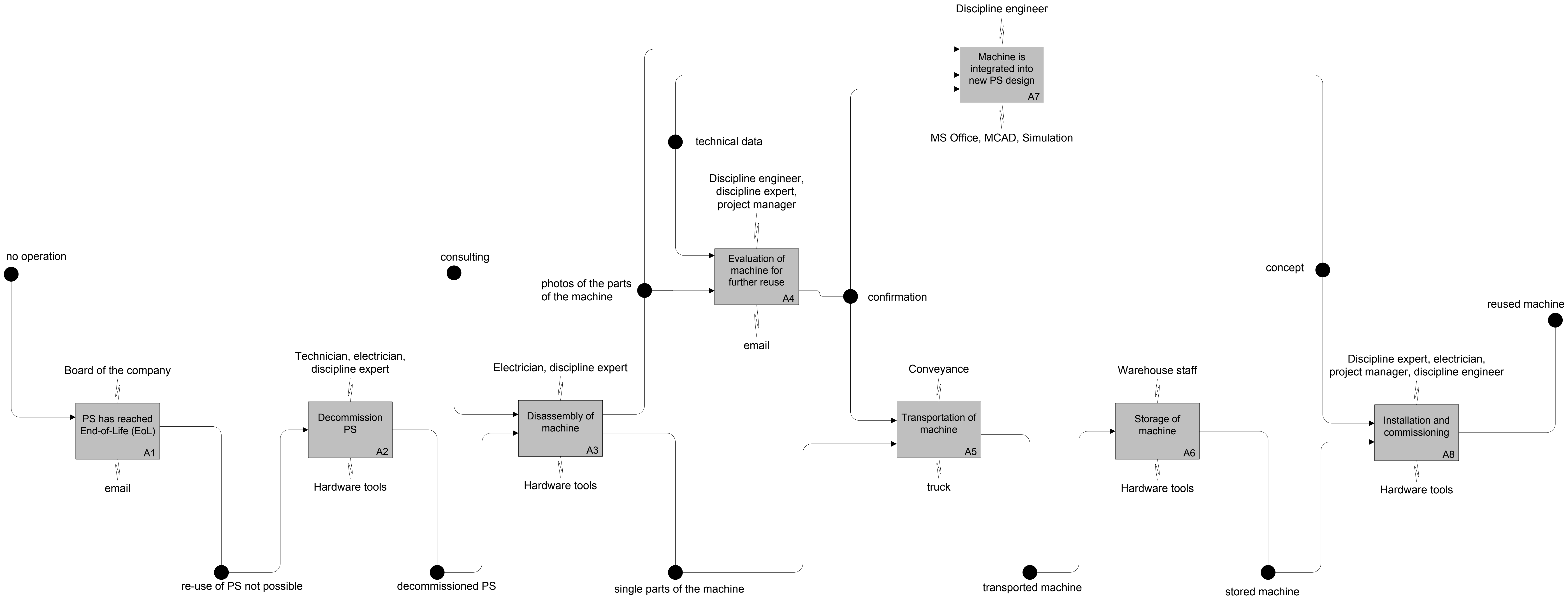

Figure D.5: Process description of case study IV: component recovery

Table D.51: A1) PS has reached End-of-Life (EoL)

activity	**A1) PS has reached End-of-Life (EoL)**
description	PS has reached its EoL because the production was relocated and the new tenant of the building had no use for the PS. That's why the PS needed to get removed.
time	It started in August 2016. It took one week.
predecessor activity	-
successor activity	A2))
involved experts	Board of the company
involved tools (software and hardware)	email
required artifacts	1) No operation
description of artifacts	1) Decision: The operation of the PS is discontinued.
created artifacts	1) re-use of PS not possible
description of artifacts	1) The re-use of the new tenant was not possible.

Table D.52: A2) Decommission PS

activity	**A2) Decommission PS**
description	PS was decommissioned. Company's personnel (UK) cleaned up the room. External service providers (UK) (electrician and discipline expert) were decommissioning the PS.
time	It started in September 2016. It took one week.
predecessor activity	A1)
successor activity	A3)
involved experts	technician (company's personnel (UK)), electrician (external service provider (UK)), discipline expert (external service provider (UK))
involved tools (software and hardware)	hardware tools
required artifacts	1) re-use of PS not possible
description of artifacts	1) The re-use of the new tenant was not possible.
created artifacts	1) decommissioned PS
description of artifacts	1) That means the PS with all its machines.

Table D.53: A3) Disassembly of machine

activity	A3) Disassembly of machine
description	The machine is disassembled into its parts. Photos of the parts are taken to capture and document the current state of the machine. Beforehand, it was voted against the reuse of the entire PS in Germany. Because the PS would not have fit into the new building (different layout, different standard for the interfaces (UK vs. Germany)). It would not have been profitable.
time	It started in September 2016. It took two weeks.
predecessor activity	A2)
successor activity	A4)
involved experts	electrician (external service provider (UK)), discipline experts (external service provider (UK))
involved tools (software and hardware)	hardware tools
required artifacts	1) decommissioned PS
	2) consulting
description of artifacts	1) That means the PS with all its machines.
	2) Company was advised that it not profitable to reuse the entire PS.
created artifacts	1) single parts of the machine
	2) photos of the parts of the machine
description of artifacts	1) Machine is disassembled into its parts.
	2) For documentation purposes, which serves as a basis for the evaluation of the current state of the machine, photos are taken of each and every part of the machine.

Table D.54: A4) Evaluation of machine for further reuse

activity	A4) Evaluation of machine for further reuse
description	Based on the photos of the parts of the machine, the company and the corresponding external service providers evaluated the machine.
time	It started in October 2016. It took one week.
predecessor activity	A3)
successor activity	A5) and A7)
involved experts	Discipline engineer (external service provider (UK), that has engineered the decommissioned PS and will also engineer the new PS), Discipline expert (German external service provider, that will realize the new PS in Germany (planned by discipline engineer)), Project manager (German personnel of the company), (Finally the board of the company needs to agree on the decision.)
involved tools (software and hardware)	email
required artifacts	1) photos of the parts of the machine
	2) technical data
description of artifacts	1) For documentation purposes, which serves as a basis for the evaluation of the current state of the machine, photos are taken of each and every part of the machine.
	2) This comprises the technical documentation of the machine incl. performance data, drawings, electrical data, and function descriptions (Important to assess whether the machine meets the requirements on the new PS in Germany.).
created artifacts	1) confirmation
description of artifacts	1) Confirmation sent through email that the machine will be reused in Germany.

Table D.55: A5) Transportation of machine

activity	A5) Transportation of machine
description	This included the packing of the parts of the machine. Those are transported to Germany.
time	It started in October 2016. It took three days.
predecessor activity	A4)
successor activity	A6)
involved experts	Conveyance (external service provider (UK))
involved tools (software and hardware)	truck
required artifacts	1) confirmation
	2) single parts of the machine
description of artifacts	1) Confirmation sent through email that the machine will be reused in Germany.
	2) Machine is disassembled into its parts.
created artifacts	1) transported machine
description of artifacts	1) Single parts of the machine are transported to Germany.

Table D.56: A6) Storage of machine

activity	A6) Storage of machine
description	The parts of the machine are stored in Germany until installation.
time	It started in October 2016. It is stored until March 2017.
predecessor activity	A5)
successor activity	A8)
involved experts	Warehouse staff (German personnel of the company)
involved tools (software and hardware)	hardware tools
required artifacts	1) transported machine
description of artifacts	1) Single parts of the machine are transported to Germany.
created artifacts	1) stored machine
description of artifacts	1) Single parts of the machine are stored to Germany at the new location.

Table D.57: A7) Machine is integrated into new PS design

activity	A7) Machine is integrated into new PS design
description	The discipline engineer develops a concept for the new PS in Germany under consideration of the machine to be reused. Due to other conditions the machine will be operating in Germany, the machine and its characteristics needed to be adapted regarding the new conditions.
time	It started in October 2016. It will be finished in January 2017.
predecessor activity	A4)
successor activity	A8)
involved experts	Discipline engineer (external service provider (UK), that has engineered the decommissioned PS and will also engineer the new PS)
involved tools (software and hardware)	MCAD (CATIA), Simulation tool specialized for the discipline, MS Office
required artifacts	1) confirmation
	2) photos of the parts of the machine
	3) technical data
description of artifacts	1) Confirmation sent through email that the machine will be reused in Germany.
	2) For documentation purposes, which serves as a basis for the evaluation of the current state of the machine, photos are taken of each and every part of the machine.
	3) This comprises the technical documentation of the machine incl. performance data, drawings, electrical data, and function descriptions (Important to assess whether the machine meets the requirements on the new PS in Germany.).
created artifacts	1) concept
description of artifacts	1) The concept comprises the dimensioning and the realization concept of the PS in which the machine is reused.

Table D.58: A8) Installation and commissioning

activity	A8) Installation and commissioning
description	The machine is installed and commissioned according to the new concept for the PS. The commissioning and ramp-up (parametrization) of the machine and of the entire PS is done by the discipline engineer.
time	It will start in March 2017. It will be finished in April 2017.
predecessor activity	A7) and A8)
successor activity	-
involved experts	Discipline expert (German external service provider), Electrician (German external service provider), Project manager (German personnel of the company), Discipline engineer (external service provider (UK), that has engineered the decommissioned PS and will also engineer the new PS)
involved tools (software and hardware)	hardware tools
required artifacts	1) stored machine
	2) concept
description of artifacts	1) Single parts of the machine are stored to Germany at the new location.
	2) The concept comprises the dimensioning and the realization concept of the PS in which the machine is reused.
created artifacts	1) reused machine
description of artifacts	1) The machine is reused.

D.5 Case study V: marketplace provider

This case study analyzes a company that offers a marketplace and auctions ranging from products to production systems of all industries as well as properties. The company acts like a broker for the selling party (private person or company) and like a marketplace for the buying party (private person or company). The company needs to estimate the value of physical assets before putting them onto their marketplace. Therefore, data from the engineering as well as from the operation phase need to be gathered by the company.

In Figure D.6 the recovery process for a machine from a production system that has reached its end of life is visualized. The interviewees had a specific project in mind when they were interviewed. The process description for this specific project contains seven activities. Each activity is then further described from Table D.59 to Table D.65.

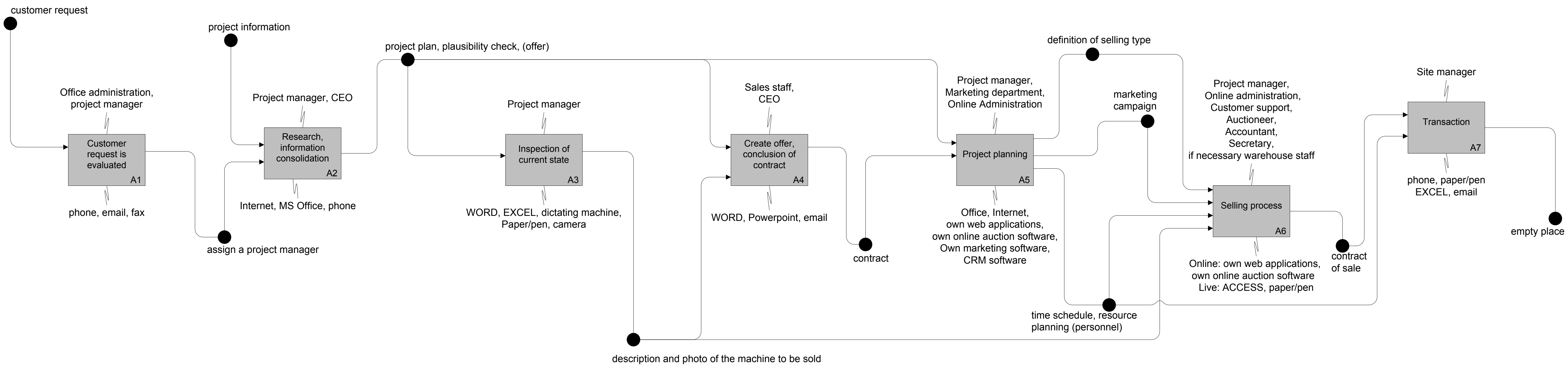

Figure D.6: Process description of case study V: component recovery

Table D.59: A1) Customer request is evaluated

activity	A1) Customer request is evaluated
description	Customer contacts the company that he wants to, e.g. sell a machine. Office administration forwards this request to the corresponding department respectively to the prospective project manager for this project.
time	It takes one day.
predecessor activity	-
successor activity	A2)
involved experts	Office administration, project manager
involved tools (software and hardware)	phone, email, fax
required artifacts	1) customer request
created artifacts	1) assign a project manager

Table D.60: A2) Research, information consolidation

activity	A2) Research, information consolidation
description	The following questions are answered: Who is the selling party/customer? What shall be sold? Where is the asset located? What amount/which number shall be sold? Sales estimates are generated and the project volume. It this project profitable for the company and for the customer?
time	It takes 1-7 days (project dependent).
predecessor activity	A1)
successor activity	A3)
involved experts	Project manager, CEO
involved tools (software and hardware)	internet, MS Office, phone
required artifacts	1) assign a project manager
	2) project information
created artifacts	1) project plan, plausibility check, (offer)

Table D.61: A3) Inspection of current state

activity	A3) Inspection of current state
description	The machine (or asset) is inspected, pictures are taken and the previously researched information is verified and corrected, if necessary.
time	It takes 1-14 days (project dependent).
predecessor activity	A2)
successor activity	A3)
involved experts	Project manager
involved tools (software and hardware)	MS WORD, MS EXCEL, dictating machine, paper/pen, camera
required artifacts	1) project plan, plausibility check, (offer)
created artifacts	1) description and photo of the machine to be sold

Table D.62: A4) Create offer, conclusion of contract

activity	A4) Create offer, conclusion of contract
description	The company estimated the value of the machine and set it off against with the accruing costs. In case the selling party/customer agrees on it, the contract can be concluded.
time	It takes one day.
predecessor activity	A3)
successor activity	A5)
involved experts	Sales staff, CEO
involved tools (software and hardware)	MS WORD, MS Powerpoint, email
required artifacts	1) project plan, plausibility check, (offer)
	2) description and photo of the machine to be sold
created artifacts	1) contract

Table D.63: A5) Project planning

activity	A5) Project planning
description	The selling type is defined: free sale, auction, or call for bids. Marketing campaign is created: How much money is spent for the marketing of which asset?
time	It starts four weeks before the sale. It takes 1-4 days (project dependent)
predecessor activity	A4)
successor activity	A6)
involved experts	Project manager, Marketing department, Online Administration
involved tools (software and hardware)	MS Office, internet, own web applications, own online auction software, own marketing software, CRM Software
required artifacts	1) project plan, plausibility check, (offer)
	2) contract
created artifacts	1) definition of selling type
	2) marketing campaign
	3) time schedule, resource planning (personnel)

Table D.64: A6) Selling process

activity	A6) Selling process
description	The activities defined during the project planning are done. The best case is that a buyer will be found and the contract of sale can be signed for the machine.
time	It starts four weeks before the online auction. Free sale or call for bids take 2-12 months.
predecessor activity	A5)
successor activity	A7)
involved experts	Project manager, Online administration, Customer support, Auctioneer, Accountant, Secretary, if necessary warehouse staff
involved tools (software and hardware)	Online: own web applications, own online auction software; Live: MS ACCESS, paper/pen
required artifacts	1) description and photo of the machine to be sold
	2) definition of selling type
	3) marketing campaign
	4) time schedule, resource planning (personnel)
created artifacts	1) contract of sale

Table D.65: A7) Transaction

activity	A7) Transaction
description	After the selling process has ended, the machine is decommissioned by the selling party/customer. The buying party takes care of the disassembly and the transportation. The company can assist here if the buying party is interested in this.
time	It starts after the complete payment. The duration is project dependent.
predecessor activity	A6)
successor activity	-
involved experts	Site manager
involved tools (software and hardware)	phone, paper/pen, MS EXCEL, email
required artifacts	1) contract of sale
	2) time schedule, resource planning (personnel)
created artifacts	1) empty place

D.6 Case study VIa and VIb: component/equipment manufacturer

This case study CS **VIa** analyzes a company that is specialized in component/equipment manufacturing as well as in systems and solutions for electrical engineering, electronics, and automation. The company itself engineers, installs, and commissions their own production systems for the production of the company's products. The company also moves the production systems from one location to another location. The relocation of the production system can be done in the same factory, because of layout changes. Or it can be done from one country to another, because of strategic or economic reasons. While relocating, the production system can also be enhanced, adapted, or modernized due to changed requirements. Thus, the company has access to the engineering data from the engineering phase. Since the company also operates the production systems it has access to the operational data (like wear and tear, repair, or maintenance data).
There is also another case study CS **VIb**. The company is not doing component recovery, i.e. it is not taking/buying back its component neither to recycle them nor to remanufacture or repair and sell them again. The problem is that the customers usually operate the component beyond the component's specifications. Additionally, the component's purchase price is relatively low as well as the amount of material, the component consists of. Since the company cannot assesses the deterioration of the component's material, it is neither profitable to take/buy them back nor possible for the company to give a warranty for the remanufactured components. Usually the customers recycle the company's components by themselves.
In case a customer puts in a warranty claim, the company is taking back the component and sending the customer a new one. In that case the component is disassembled and recycled by the company in recycling containers. Those are usually used for production waste. Disassembly before recycling is more profitable due to higher sale prices for unmixed scrap than for mixed scrap.
It is also possible that the company transports its own craps directly to, e.g. an ironworks. But to do so, the ironworks demands a constant delivery of a fixed amount of scrap.

In Figure D.7 the recovery process for production systems is visualized. The interviewee had a specific project in mind when he was interviewed. The process description for this specific project contains 13 activities. Each activity is further described from Table D.66 to Table D.78. In this case it was not possible to develop a detailed process description due to the interview structure and the provided documents. Missing points are the artifact flow and the used tool information in Figure D.7 as well as the detailed descriptions in the corresponding tables.

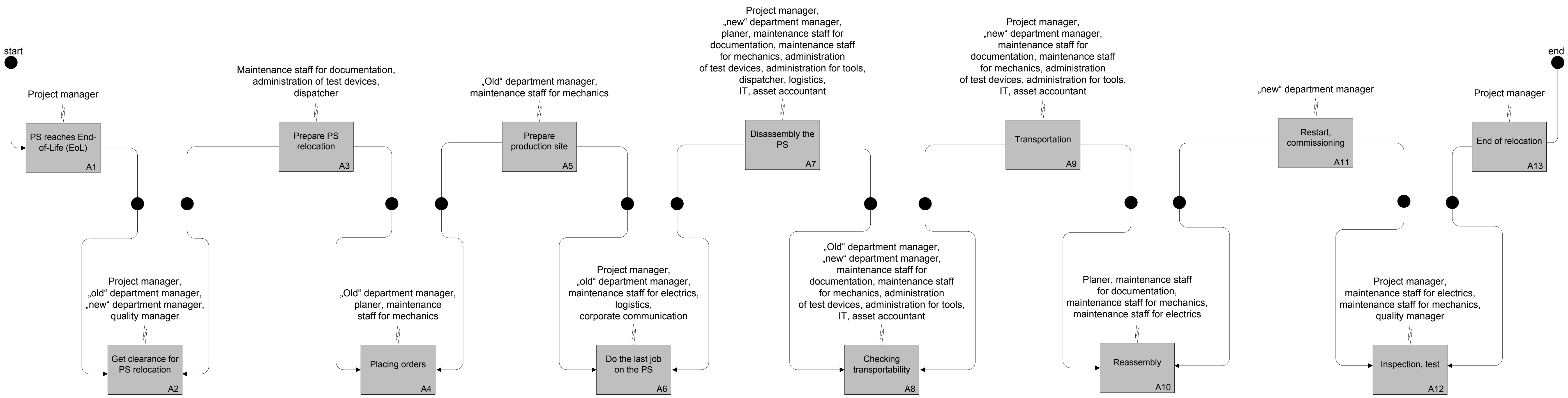

Figure D.7: Process description of case study VIa: production system relocation / recovery

Table D.66: A1) PS reaches End-of-Life (EoL)

activity	A1) PS reaches End-of-Life (EoL)
description	The project manager starts the relocation process within a workflow system.
predecessor activity	-
successor activity	A2)
involved experts	Project manager

Table D.67: A2) Get clearance for PS relocation

activity	A2) Get clearance for PS relocation
description	In case an external service provider is needed for this relocation, it takes the project manager five days to get an offer. The 'old', the 'new' department manager, and the quality manager give their clearance within one day. The 'old' department manager is person in charge of at the location from where the production system is removed. The 'new' department manager is the person in charge at the location to where the production system is moved.
predecessor activity	A1)
successor activity	A3)
involved experts	Project manager, 'old' department manager, 'new' department manager, quality manager

Table D.68: A3) Prepare PS relocation

activity	A3) Prepare PS relocation
description	The maintenance staff for documentation prepares the production system relocation, which takes one day. The maintenance schedule for the test devices gets adapted (one day). It might be also necessary to increase the stock to compensate the relocation time when the prduction system cannot produce the products (one day).
predecessor activity	A2)
successor activity	A4)
involved experts	Maintenance staff for documentation, administration of test devices, dispatcher

Table D.69: A4) Placing orders

activity	A3) Placing orders
description	It takes one day to organize the disposal of the cooling lubricant. This is done by the 'old' department manager. The planer needs to organize the building services. The maintenance staff for mechanics clarifies the loading situation and make the necessary arrangements regarding date and time and consults the disciplines. Each of them takes a day.
predecessor activity	A3)
successor activity	A5)
involved experts	'Old' department manager, planer, maintenance staff for mechanics

Table D.70: A5) Prepare production site

activity	A5) Prepare production site
description	The 'old' department manager checks the new location for the production system (one day). The maintenance staff for mechanics prepares the new location (one day).
predecessor activity	A4)
successor activity	A6)
involved experts	'Old' department manager, maintenance staff for mechanics

Table D.71: A6) Do the last job on the PS

activity	A6) Do the last job on the PS
description	The project manager checks the schedule (it takes a day) two weeks before the relocation. The 'old' department manager does the final and last job on the production system (one day) and starts with the cleaning and transportation activities, which also take a day. The maintenance staff for electrics is responsible for the electric transport lock of the production system (one day). The logistics decouples this production supply area from the remaining one, which takes a day. Corporate communication is in charge to communicate this relocation.
predecessor activity	A5)
successor activity	A7)
involved experts	Project manager, 'old' department manager, maintenance staff for electrics, logistics, corporate communication

Table D.72: A7) Disassembly the PS

activity	A7) Disassembly the PS
description	The planer removes the connections (one day). The project manager checks the labels of the connections (one day). The new' department manager organizes that the personnel data is changed by human resources (one day), that the quality workplaces are created (one day), and that the production system is put into the environment data base (one day). The maintenance staff for documentation maintains the equipment master data (one day). The maintenance staff for mechanics informs the production system engineer and does the mechanical disassembly. Each takes them one day. The administration of test devices transfers/rebook the test devices (one day). And the administration for tools does this for the tools, which also takes a day. The dispatcher transfers the stock. The logistics replans the delivery goal. The IT maintains the workplaces (one day). And the asset accountant updates the value of the asset (one day).
predecessor activity	A6)
successor activity	A8)
involved experts	Project manager, 'new' department manager, planer, maintenance staff for documentation, maintenance staff for mechanics, administration of test devices, administration for tools, dispatcher, logistics, IT, asset accountant

Table D.73: A8) Checking transportability

activity	A8) Checking transportability
description	The 'old' department manager organizes a clearance for the passing of risk (one day). The maintenance staff for mechanics checks the readiness for transportation (one day). The actions of the 'new' department manager, maintenance staff for documentation, maintenance staff for mechanics, administration of test devices, administration for tools, IT, and asset accountant mentioned in A7) can also taken place in this activity.
predecessor activity	A7)
successor activity	A9)
involved experts	'Old' department manager, 'new' department manager, maintenance staff for documentation, maintenance staff for mechanics, administration of test devices, administration for tools, IT, asset accountant

Table D.74: A9) Transportation

activity	A9) Transportation
description	The project manager supervises the transportation of the produciton system (one day). The actions of the 'new' department manager, maintenance staff for documentation, maintenance staff for mechanics, administration of test devices, administration for tools, IT, and asset accountant mentioned in A7) can also taken place in this activity.
predecessor activity	A8)
successor activity	A10)
involved experts	Project manager, 'new' department manager, maintenance staff for documentation, maintenance staff for mechanics, administration of test devices, administration for tools, IT, asset accountant

Table D.75: A10) Reassembly

activity	A10) Reassembly
description	The planer organizes the connection with the building services (one day). The maintenance staff for documentation documents the IP addresses (one day). The maintenance staff for mechanics does the reassembly of the production system (one day). The maintenance staff for electrics is responsible for establishing the data connections (one day) and for the removal of the electrical transport lock.
predecessor activity	A9)
successor activity	A11)
involved experts	Planer, maintenance staff for documentation, maintenance staff for mechanics, maintenance staff for electrics

Table D.76: A11) Restart, commissioning

activity	A11) Restart, commissioning
description	The 'new' department manager ramps up the production system and the production.
predecessor activity	A10)
successor activity	A12)
involved experts	'New' department manager

Table D.77: A12) Inspection, test

activity	A12) Inspection, test
description	The project manager let the production system be checked by the authorities. The maintenance staff for electrics, maintenance staff for mechanics, and quality manager test the operating production system.
predecessor activity	A11)
successor activity	A13)
involved experts	Project manager, maintenance staff for electrics, maintenance staff for mechanics, quality manager

Table D.78: A13) End of relocation

activity	A13) End of relocation
description	The project manager terminates the project and the relocation of the production system.
predecessor activity	A11)
successor activity	A13)
involved experts	Project manager

D.7 Case study VII: assembly and relocation company

This case study analyzes a company that is specialized in assembly and relocation of machines and production systems as a service. This company disassembles, relocates, reassembles, and re-commissions the machines and production systems of their customers. These customers are companies from every type of industry. It is also possible that the machine or production system is maintained or modernized while relocating it according to customer's requirements. The company moves the machines and production systems, which they have not planned, from one location to another location on behalf of the customer. Before the relocation the company needs to gather data from the engineering as well as from the operation phase.

E Description of the business ecosystem in the End-of-Life phase

This annex provides a description of each stakeholder of the business ecosystem in the EoL phase including its business and the availability of information. All descriptions are based on the interview results of and insights through the case studies (listed in Table 4.1.)
The numbering of each stakeholder refers to the numbering in Figure 4.2.

a) Production system operator - EoL3 The production system operator (or owner) has its own service which is capable of doing a removal with/without remanufacturing/repair. Usually this is a removal within the company just between different locations, e.g. from Germany to Poland. It can be used to build up a new “bigger” system or to integrate it into an already existing one, e.g. a factory.
Documentation “as built” is available. The production system is well known but inspection is necessary to get the “as-is” state.

b) System integrator - EoL3 The system integrator that had engineered the production system before is now in charge to do the removal of the production system for the production system operator (or owner), usually within contracting company just between different locations. It can be used to build up a new “bigger” system or to integrate it into an already existing one, e.g. a factory.
Documentation “as built” and knowledge are available, since the system integrator has engineered and built up the production system. Inspection is necessary to get the “as-is” state.

c) Third party (PS) (with broker) - EoL3 A third party service provider is in charge to do the removal of the production system either only for the production system operator (or owner) or for the future production system operator (or owner). The relocation can, therefore, be done between the different locations of one company or between two different companies. It can be used to build up a new “bigger” system or to integrate it into an already existing one, e.g. a factory.
Documentation “as built” is provided by production system operator (or owner). The third party service provider has usually no knowledge of the production system before. Inspection is necessary to get the “as-is” state.
This recovery process can be accompanied by a broker which had connected the actual production system operator or owner (selling party) with the future one (buying party) before the third party service provider starts the recovery. But also the broker could be the first company in charge. That means, that the buying party contract the broker to find or to propose a proper third party service provider to do the relocation.

d) Production system operator - EoL2 The production system has reached its end of life. EoL3 is not possible. The production system operator (or owner) has its own

service which is capable to assess the components to recover before the production system gets recycled. The extracted components are, then, used to equip still running systems of this company or they are used to build up a new production system within the company. (Or it is used to build up other systems, which would lead the components out of the production system life cycle into other life cycles, e.g. those of logistics systems.) But in this recovery process there can also be a system integrator involved at the same time. The reason could be that special knowledge is necessary for decommissioning the component at location A and commissioning the component at location B in a new of already existing production system. Therefore, a system integrator can be involved, since it had usually engineered and built up the component before.
Documentation “as built” is available. Components are well known but inspection is necessary to get the “as-is” state.

e) System integrator - EoL2 The production system has reached its end of life. The system integrator that had engineered the production system before is now in charge to assess the components to recover for the production system operator (or owner) before the production system gets recycled. Usually the system integrator is at the same time in charge to engineer a new production system at this place by using the recovered components. Another possibility is that the system integrator is in charge to relocate the production system for the production system operator (or owner) and when this also includes remanufacturing activities, it can happen that old components need to be replaced by new ones. In that case, those used components are collected and handed over to the production system operator (or owner). These components can, then, be used in another production system. Or it is used to build up other systems, which would lead the components out of the production system life cycle into other life cycles, e.g. those of logistics systems.
Documentation “as built” and knowledge are available, since the system integrator has engineered and built up the production system with its components. Inspection is necessary to get the “as-is” state.

f) Third party (PS) (with broker) - EoL2 The production system has reached its end of life. A third party service provider is now in charge to assess the components to recover for the production system operator (or owner) before the production system gets recycled. Usually the third party service provider is at the same time in charge to engineer a new production system at this place by using the recovered components. Another possibility is that the third party service provider is in charge to relocate the production system for a company and when this also includes remanufacturing activities, it can happen that old components need to be replaced by new ones. In that case, those used components can be sold directly by the third party service provider or with a broker involved. These components can, then, be used in another production system. Or it is used to build up other systems, which would lead the components out of the production system life cycle into other life cycles, e.g. those of logistics system.
Documentation “as built” is provided by production system operator (or owner). The third party service provider has usually no knowledge of the production system with its components before. Inspection is necessary to get the “as-is” state.

g) Component manufacturer - EoL2 The production system has reached its end of life. The component manufacturer is in charge to pick up the components from the production system operator's (or owner's) place. Usually the component manufacturer has sold exactly those components to the production system operator (or owner). The component manufacturer can either buy the used components from the production system operator (or owner) to resell them (directly or after remanufacturing) to another company/person or can be in charge to remanufacture the components and bring them back to the production system operator (or owner) afterwards. These components can, then, be used in another production system. Or it is used to build up other systems, which would lead the components out of the production system life cycle into other life cycles, e.g. those of logistics system.
Documentation "as built" and knowledge about the components are available. Inspection is necessary to get the "as-is" state.

h) Third party (Comp.) (with broker) - EoL2 The production system has reached its end of life. The third party service provider is in charge to pick up the components from the production system operator's (or owner's) place. The third party service provider can either buy the used components from the production system operator (or owner) to resell them (directly or after remanufacturing) to another company/person (by itself or in cooperation with a broker) or can be in charge to remanufacture the components and bring them back to the production system operator (or owner) afterwards. These components can, then, be used in another production system. Or it is used to build up other systems, which would lead the components out of the production system life cycle into other life cycles, e.g. those of logistics system.
Documentation "as built" is provided by production system operator (or owner). The third party service provider has usually no knowledge of the components before. Inspection is necessary to get the "as-is" state.

i) Production system operator with broker - EoL1 The production system has reached its end of life. EoL3 is not possible and all reusable components got extracted. The production system operator (or owner) has its own service which is doing the material recovery of the production system. The material is, then, either picked up by a third party service provider, which further sorts the material, or the production system operator (or owner) transports the material directly to a broker. A broker is here a scrap dealer. Finally, the materials can be used to produce components of a new generation for production systems. Or it is used to produce other products, which would lead the materials out of the production system life cycle into other life cycles, e.g. those of cars. Documentation "as built" and knowledge are available. Inspection is necessary to get the "as-is" state.

j) System integrator with broker - EoL1 The production system has reached its end of life. EoL3 is not possible and all reusable components got extracted. The system integrator that had engineered the production system before is now in charge to remove the production system. Usually the system integrator is at the same time in charge to engineer a new production system at this place for the production system operator (or owner). Another possibility is that the system integrator is in charge to relocate the production system for the production system operator (or owner) and when this also includes remanufacturing activities, it can happen that used materials and waste are the result. In both cases those are, then, transported to a broker. A broker is here a scrap dealer. Finally, the materials can be used to produce components of a new generation for production systems. Or it is used to produce other products, which would lead the materials out of the production system life cycle into other life cycles, e.g. those of cars. Documentation “as built” and knowledge are available, since the system integrator has engineered and built up the production system with its components and materials. Inspection is necessary to get the “as-is” state.

k) Third party (PS) with broker - EoL1 The production system has reached its end of life. EoL3 is not possible and all reusable components got extracted. The third party service provider is now in charge to remove the production system for he production system operator (or owner). Usually the third party service provider is at the same time in charge to engineer a new production system at this place. Another possibility is that the system integrator is in charge to relocate the production system for the production system operator (or owner) and when this also includes remanufacturing activities, it can happen that used materials and waste are the result. In both cases those are, then, transported to a broker. A broker is here a scrap dealer. Finally, the materials can be used to produce components of a new generation for production systems. Or it is used to produce other products, which would lead the material out of the production system life cycle into other life cycles, e.g. those of cars.
Documentation “as built” is provided by production system operator (or owner). The third party service provider has usually no knowledge of the production system with its components and materials before. Inspection is necessary to get the “as-is” state.

l) Third party (Mat.) with broker - EoL1 The production system has reached its end of life. EoL3 is not possible and all reusable components got extracted. A third party service provider, specialized in removal, disassembly, and material extraction, is now in charge to remove the production system for the production system operator (or owner). The extracted and sorted material is, then, transported to a broker. A broker is here a scrap dealer. Finally, the materials can be used to produce components of a new generation for production systems. Or it is used to produce other products, which would lead the materials out of the production system life cycle into other life cycles, e.g. those of cars.
Documentation “as built” is provided by production system operator (or owner). The third party service provider has usually no knowledge of the production system with its components and materials before. Inspection is necessary to get the “as-is” state.

F Detailed description of the End-of-Life data model

This chapter provides detailed information about the attributes of the EoL data model as well as their attribute grouping.

F.1 Base End-of-Life data model

This section provides additions and a grouping of the attributes of the classes of the base EoL data model.

Figure F.1 shows the grouping of the class *production system* of the base EoL data model (see Figure 3.3). The base EoL data model contains this information, which needs to come from the engineering phase and operation phase in order to enable a proper EoL phase. When looking at the EoL process data model (see Figure 3.4), which contains information that usually needs to come from the EoL phase itself, it is obvious that some of this information needs to be also provided by the engineering and/or operation phase, so that a proper EoL phase can be enabled. In the case of the class *production system* this is the ’life expectation’, ’weight’, and ’size’. Table F.1 correspondingly provides the attribute descriptions.

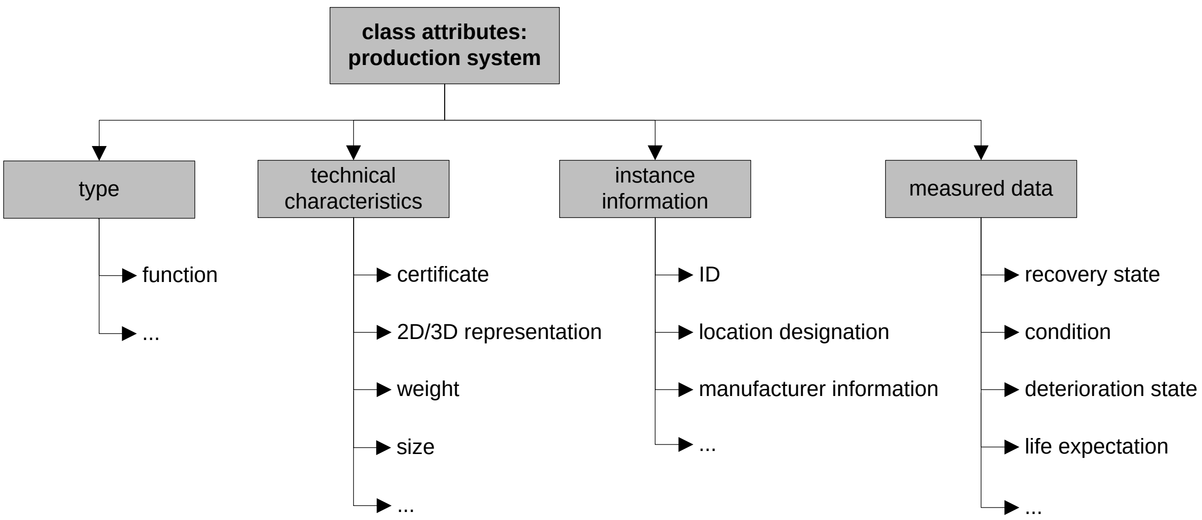

Figure F.1: Addition and grouping of the attributes of the class production system (of the base End-of-Life data model)

Table F.1: Addition and grouping of the attributes of the class production system (of the base End-of-Life data model)

attribute	description
type	
function	It provides the different functions the production system has.
technical characteristics	
certificate	It provides laws, guidelines, directives etc. that are met by the production system.
2D/3D representation	It provides a visualization of the production system.
weight	It provides information about the weight of the production system.
size	It provides information about the size of the production system.
instance information	
ID	It provides an unambiguous identifier that should not change throughout the production system life cycle.
location designation	It provides the current location of the production system.
manufacturer information	It provides vendor information.
measured data	
recovery state	It provides information if a production system was already reused, remanufactured etc.
condition	It provides information about the state of the production system (e.g. harmfulness, degree of contamination).
deterioration state	It provides information about life time and load dependent quality properties.
life expectation	It provides information about the estimated residual life time or useful life of the production system.

For the class *component* the attributes 'life expectation', 'weight', and 'size' are added to the base EoL data model (see Figure F.2). Table F.2 correspondingly provides the attribute descriptions.

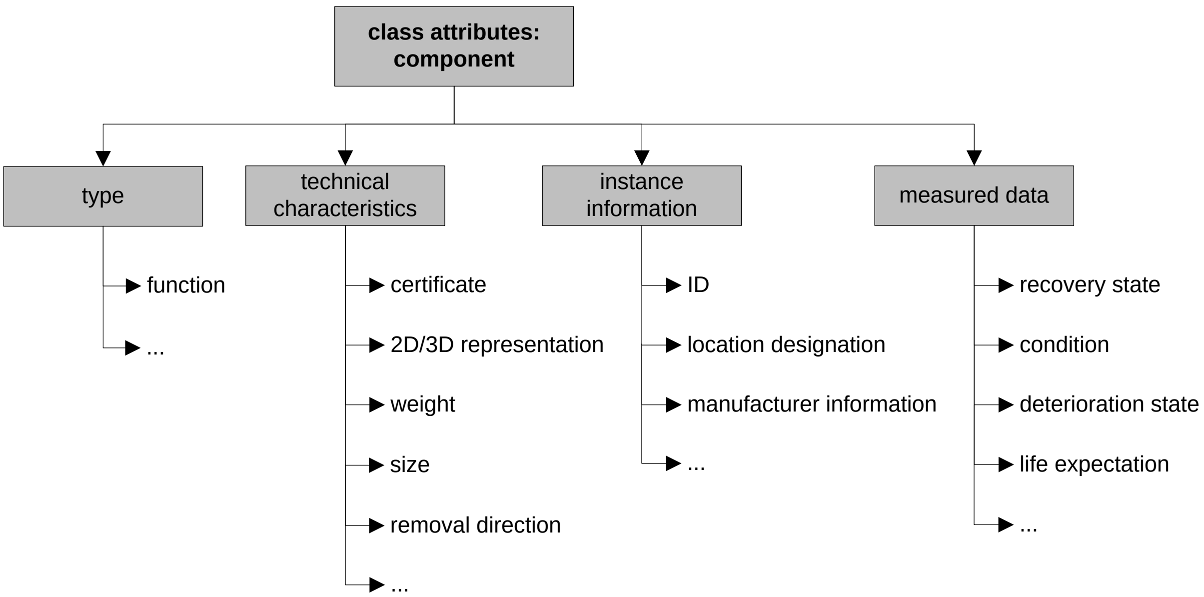

Figure F.2: Addition and grouping of the attributes of the class component (of the base End-of-Life data model)

Table F.2: Addition and grouping of the attributes of the class component (of the base End-of-Life data model)

attribute	description
type	
function	It provides the different functions the component has.
technical characteristics	
certificate	It provides laws, guidelines, directives etc. that are met by the component.
2D/3D representation	It provides a visualization of the component.
weight	It provides information about the weight of the component.
size	It provides information about the size of the component.
removal direction	It provides information about how the component can be removed out of the production system.
instance information	
ID	It provides an unambiguous identifier that should not change throughout the component life cycle.
location designation	It provides the current location of the component.
manufacturer information	It provides vendor information.
measured data	
recovery state	It provides information if a component was already reused, remanufactured etc.
condition	It provides information about the state of the component (e.g. harmfulness, degree of contamination).
deterioration state	It provides information about life time and load dependent quality properties.
life expectation	It provides information about the estimated residual life time or useful life of the component.

For the class *material* the attributes 'life expectation', 'weight', 'size', and 'technical characteristics' are added to the base EoL data model (see Figure F.3). Table F.3 correspondingly provides the attribute descriptions.

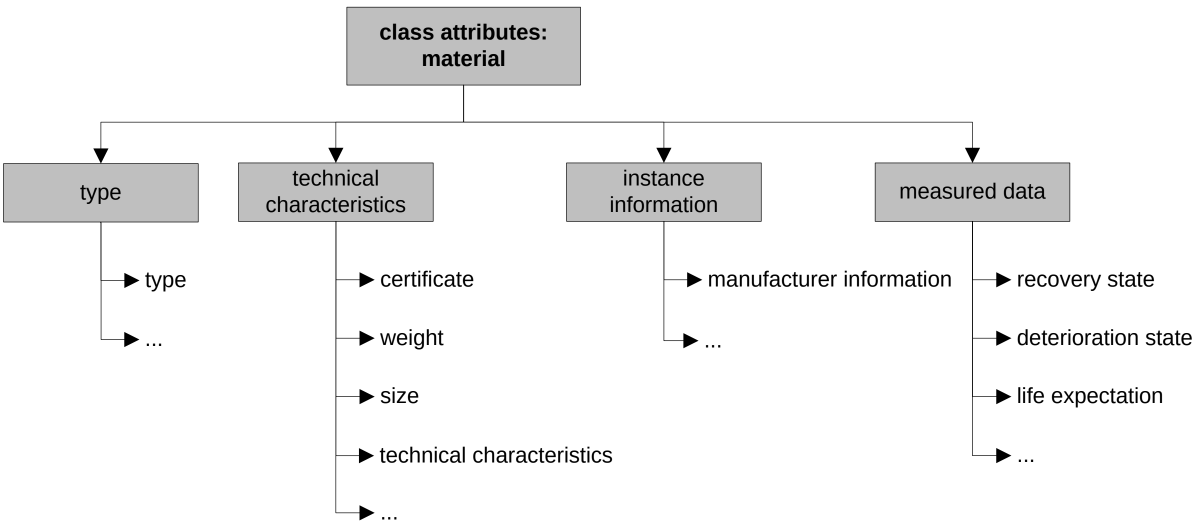

Figure F.3: Addition and grouping of the attributes of the class material (of the base End-of-Life data model)

Table F.3: Addition and grouping of the attributes of the class material (of the base End-of-Life data model)

attribute	description
type	
type	It provides an identification of the material.
technical characteristics	
certificate	It provides laws, guidelines, directives etc. that are met by the material.
weight	It provides information about the weight of the material.
size	It provides information about the size of the material.
technical characteristics	It provides technical characteristics.
instance information	
manufacturer information	It provides vendor information.
measured data	
recovery state	It provides information if a component was already reused, remanufactured etc.
deterioration state	It provides information about life time and load dependent quality properties.
life expectation	It provides information about the estimated residual life time or useful life of the component.

For the class *connection* the attribute 'technical characteristics' and 'working surface' are added to the base EoL data model (see Figure F.4). Table F.4 correspondingly provides the attribute descriptions.

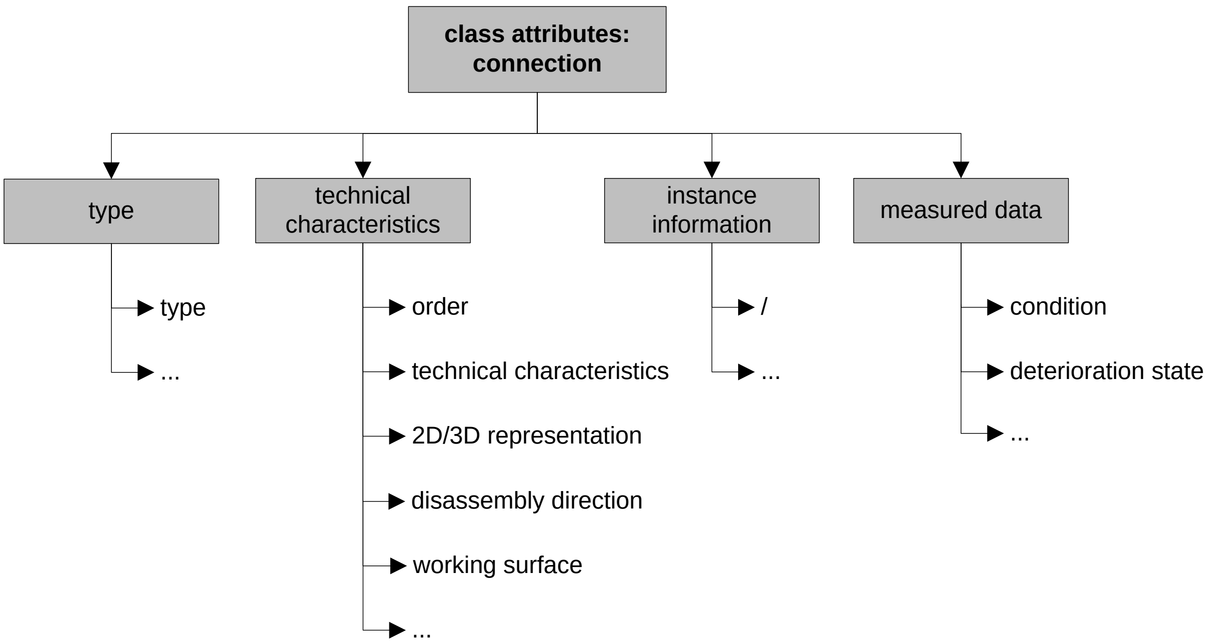

Figure F.4: Addition and grouping of the attributes of the class connection (of the base End-of-Life data model)

Table F.4: Addition and grouping of the attributes of the class connection (of the base End-of-Life data model)

attribute	description
type	
type	It provides an identification of the connection.
technical characteristics	
order	It provides information about assembly order.
technical characteristics	It provides technical characteristics.
2D/3D representation	It provides a visualization of the connection.
disassembly direction	It provides the direction how a connections can be accessed.
working surface	It provides information about where a tool can be used at the connection to undo the it.
instance information	
/	/
measured data	
condition	It provides information about the state of the connection (e.g. force to be applied to undo the connection).
deterioration state	It provides information about life time and load dependent quality properties.

For the class *memory* the only attribute 'type' is grouped into type. Due to the simplicity of this class, an own figure is not provided. Table F.5 correspondingly provides the attribute descriptions.

Table F.5: Grouping of the attribute of the class memory (of the base End-of-Life data model)

attribute	description
type	
type	It provides an identification of the memory.
technical characteristics	
/	/
instance information	
/	/
measured data	
/	/

F.2 End-of-Life process data model

This section provides a grouping of the attributes of the classes of the EoL process data model.

Figure F.5 shows that grouping for the classes *production system* and *component*. Both are identical. Table F.6 correspondingly provides the attribute descriptions.

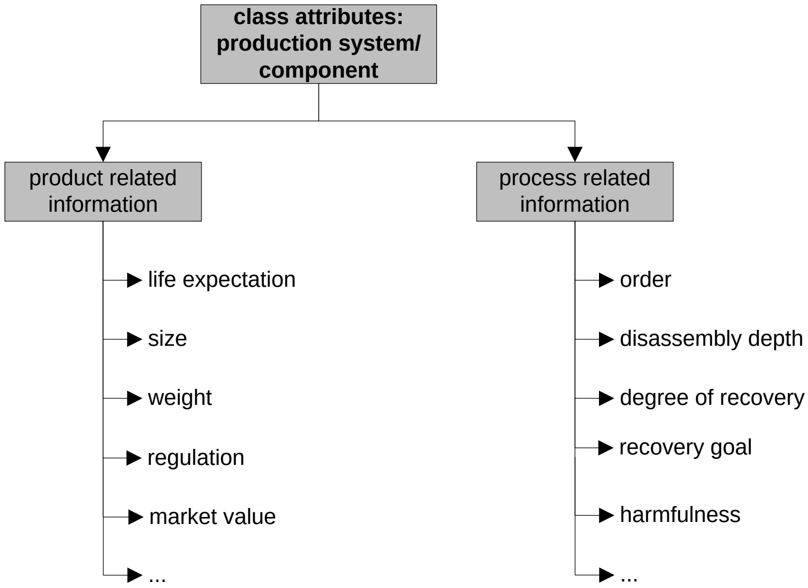

Figure F.5: Grouping of the attributes of the classes production system and component (of the End-of-Life process data model)

Table F.6: Grouping of the attributes of the classes production system and component (of the End-of-Life process data model)

attribute	description
product related information	
life expectation	It provides information about the estimated residual life time or useful life of the production system or component.
weight	It provides information about the weight of the production system or component.
size	It provides information about the size of the production system or component.
regulation	It provides laws, guidelines, directives etc. that need to be met by the production system or component.
market value	It provides information about the current value on the market for this specific production system or component.
process related information	
order	It provides information about what is input for the recovery process and what is output.
disassembly depth	It provides the depth of disassembly for the production system or component.
degree of recovery	It provides the rate of recovery after the recovery itself.
recovery goal	It provides information if a production system or component is reusable, recyclable etc.
harmfulness	It provides the classification if a certain production system or component is considered as harmful or not.

For the class *material* the grouping is as given in Figure F.6. Table F.7 correspondingly provides the attribute descriptions.

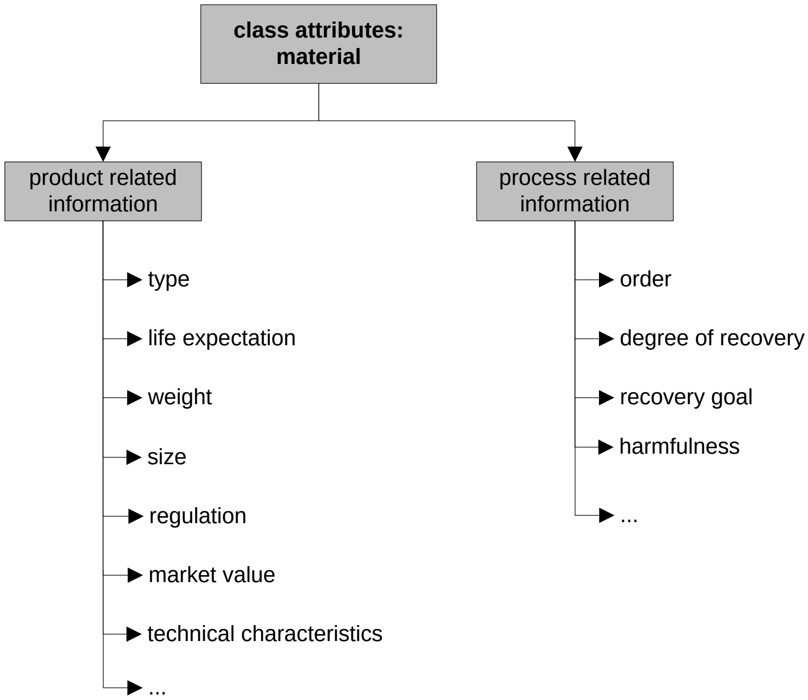

Figure F.6: Grouping of the attributes of the class material (of the End-of-Life process data model)

Table F.7: Grouping of the attributes of the class material (of the End-of-Life process data model)

attribute	description
product related information	
type	It provides an identification of the material.
life expectation	It provides information about the estimated residual life time or useful life of the material.
weight	It provides information about the weight of the material.
size	It provides information about the size of the material.
regulation	It provides laws, guidelines, directives etc. that need to be met by the material.
market value	It provides information about the current value on the market for this specific material.
technical characteristics	It provides physical and/or chemical characteristics of the material.
process related information	
order	It provides information about what is input for the recovery process and what is output.
degree of recovery	It provides the rate of recovery after the recovery itself.
recovery goal	It provides information if a production system or component is reusable, recyclable etc.
harmfulness	It provides the classification if a certain material is considered as harmful or not.

For the class *connection* the grouping is as given in Figure F.7. Table F.8 correspondingly provides the attribute descriptions.

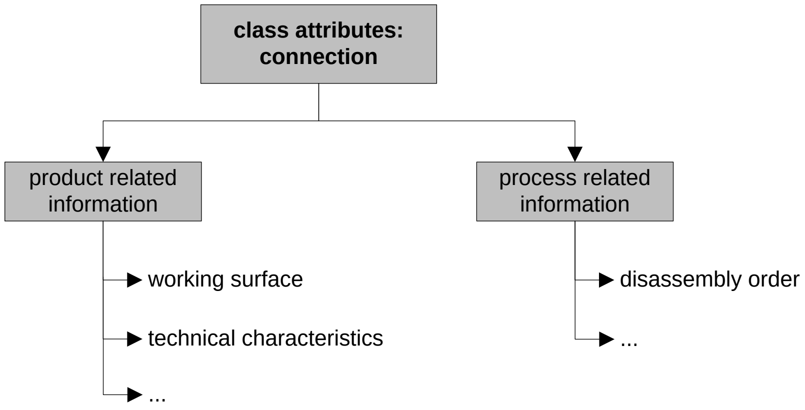

Figure F.7: Grouping of the attributes of the class connection (of the End-of-Life process data model)

Table F.8: Grouping of the attributes of the class connection (of the End-of-Life process data model)

attribute	description
product related information	
working surface	It provides information about where a tool can be used at the connection to undo the it.
technical characteristics	It provides technical characteristics.
process related information	
disassembly order	It indicates the overall sequence of disassembly.

For the class *memory* the only attribute ’type’ is grouped into the product related information. Due to the simplicity of this class, an own figure is not provided. Table F.9 correspondingly provides the attribute descriptions.

Table F.9: Grouping of the attributes of the class memory (of the End-of-Life process data model)

attribute	description
product related information	
type	It provides an identification of the memory.
process related information	
/	/

For process classes a grouping into product, process, and resource related information seems reasonable - see Figure F.8. Table F.10 correspondingly provides the attribute descriptions.

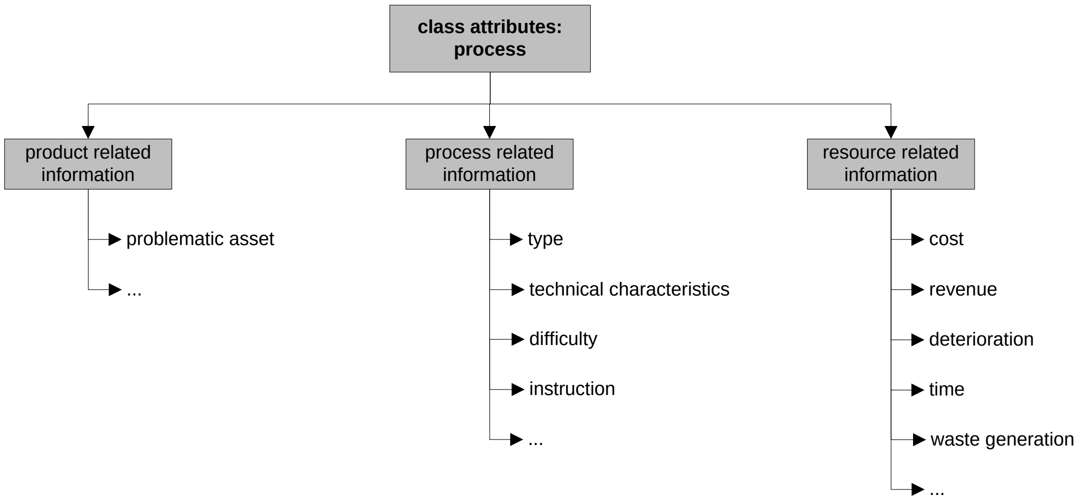

Figure F.8: Grouping of the unified attributes of the process classes (of the End-of-Life process data model)

Table F.10: Grouping of the common attributes of the process classes (of the End-of-Life process data model)

attribute	description
product related information	
problematic asset	It provides information about objects or substances that are problematic for the process.
process related information	
type	It provides an identification of the process.
technical characteristics	It provides technical parameters of the process.
difficulty	It provides a classification how difficult a process is.
instruction	It provides visualization or guidance for the workers that executes the process.
resource related information	
cost	It provides the costs of the process.
revenue	It provides information whether it can make money with this process.
deterioration	It provides information about the quality loss of the physical asset through having the process executed.
time	It provides the time a certain process needs.
waste generation	It provides information how much waste is generated because of this process.

The attributes correspond with the process attributes in *cost* and *revenue*, since only the physical realization of a process can generate costs and revenues. Therefore, the attributes are grouped into process related information and resource related information - see Figure F.9. Table F.11 correspondingly provides the attribute descriptions.

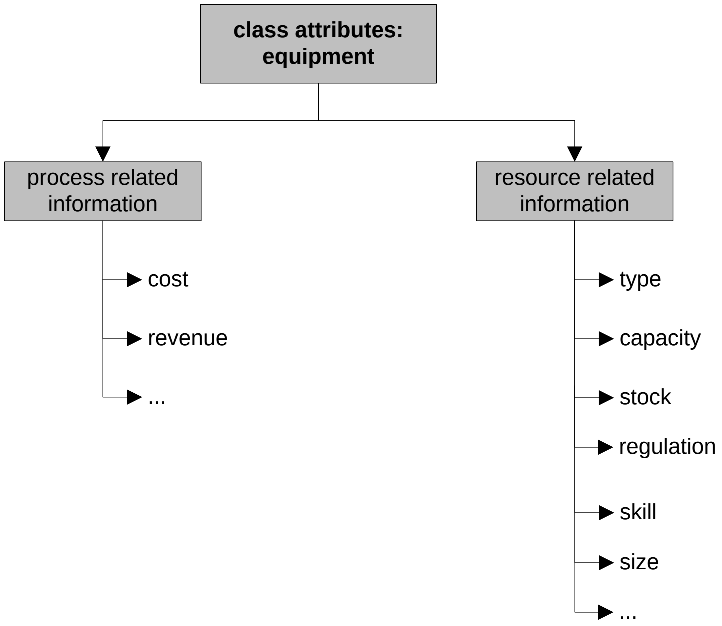

Figure F.9: Grouping of the attributes of the class equipment (of the End-of-Life process data model)

Table F.11: Grouping of the attributes of the class equipment (of the End-of-Life process data model)

attribute	description
process related information	
cost	It provides the costs of executing the process.
revenue	It provides information whether it can make money with executing this process.
resource related information	
type	It provides an identification of the memory.
capacity	It provides capacity constraints of the equipment regarding the work plan or schedule.
stock	It provides capacity constraints of the equipment regarding storage.
regulation	It provides laws, guidelines, directives etc. that need to be met by the equipment.
skill	It provides the necessary skills of a human to operate an equipment or to execute the process.
size	It provides the dimensions of an equipment.

The multiplicities of the association relations of the EoL process data model could be considered as 0..* but are not explicitly, quantitatively given in Figure F.10.

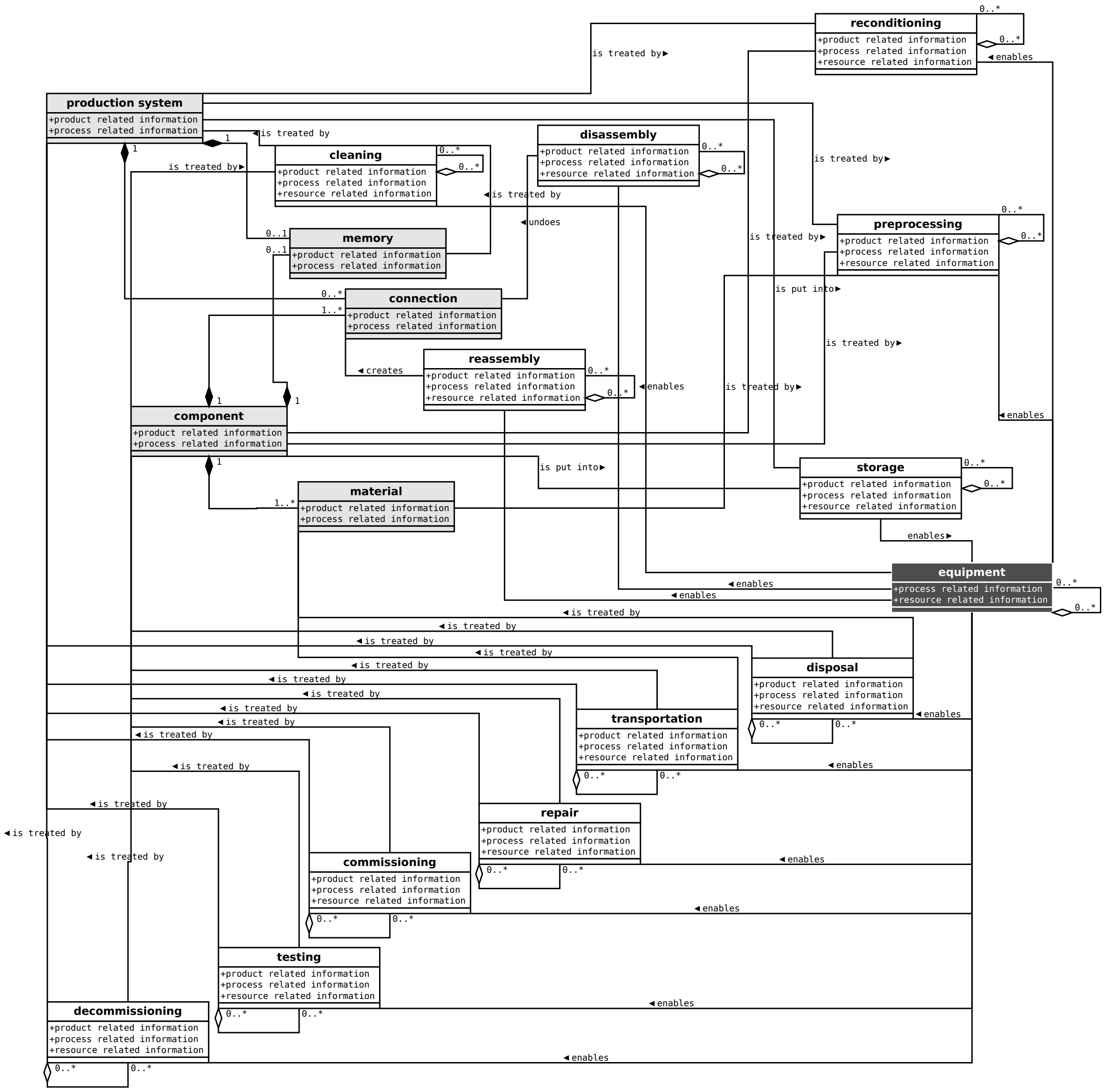

Figure F.10: End-of-Life process data model (unified and generalized)

F.3 End-of-Life data model

The EoL data model is the result, when the base EoL data model and the EoL process data model are aggregated (see Figure F.11). The multiplicities of the association relations of the EoL data model could be considered as 0..* but are not explicitly, quantitatively given in this figure.

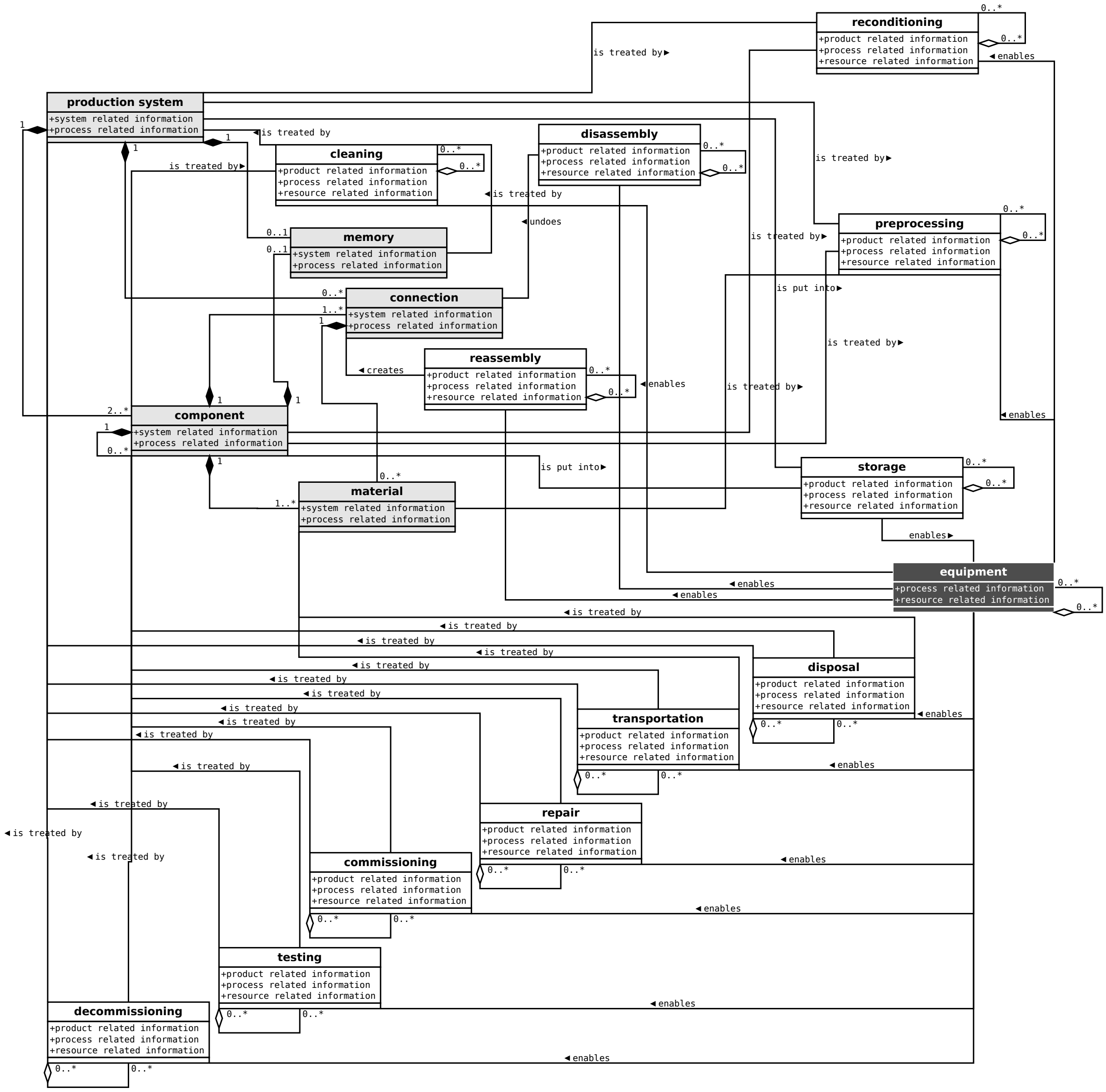

Figure F.11: End-of-Life data model (unified and generalized)

G Library concept for the recovery planning method

This chapter shows the library concept of the RPM.

Figure G.1 shows the workflow on using and creating recovery goal templates as library entries. It also visualizes interrelations to other libraries.

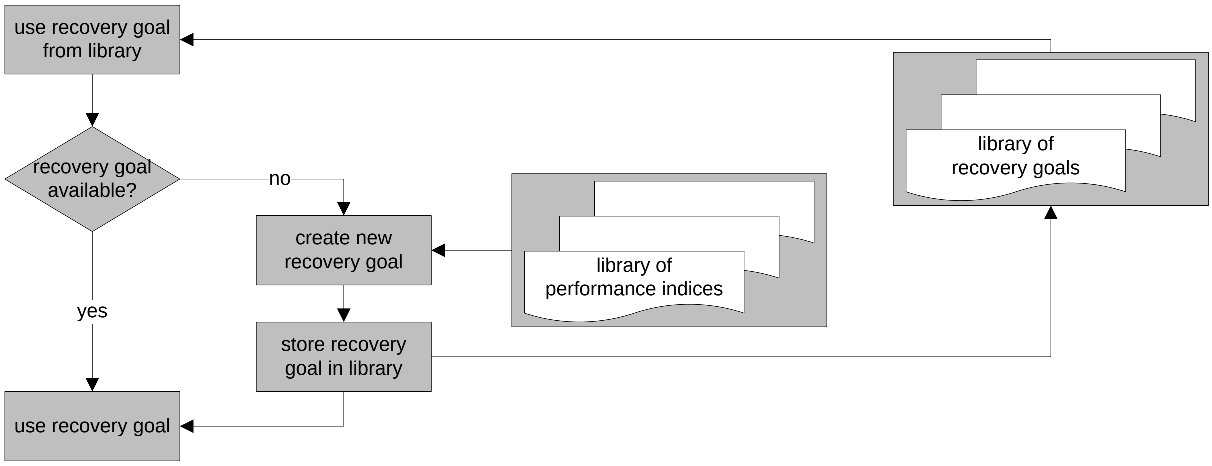

Figure G.1: Workflow for recovery goals as libraries entries

Figure G.2 shows the workflow on using and creating EoL strategy templates as library entries. It also visualizes interrelations to other libraries.

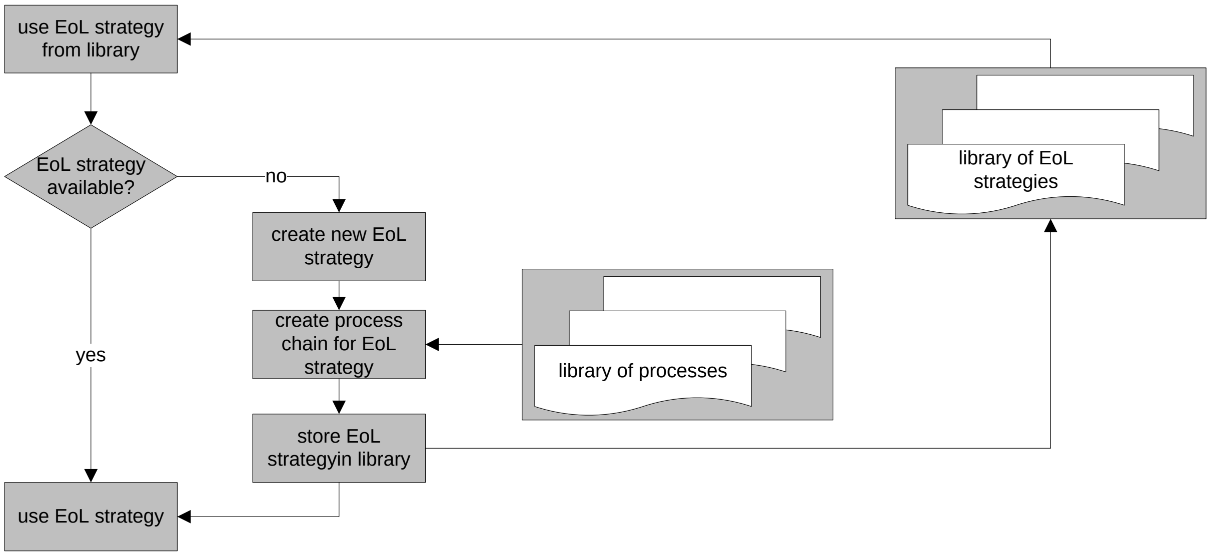

Figure G.2: Workflow for EoL strategies as libraries entries

Figure G.3 shows the workflow on using and creating process templates as library entries. It also visualizes interrelations to other libraries.

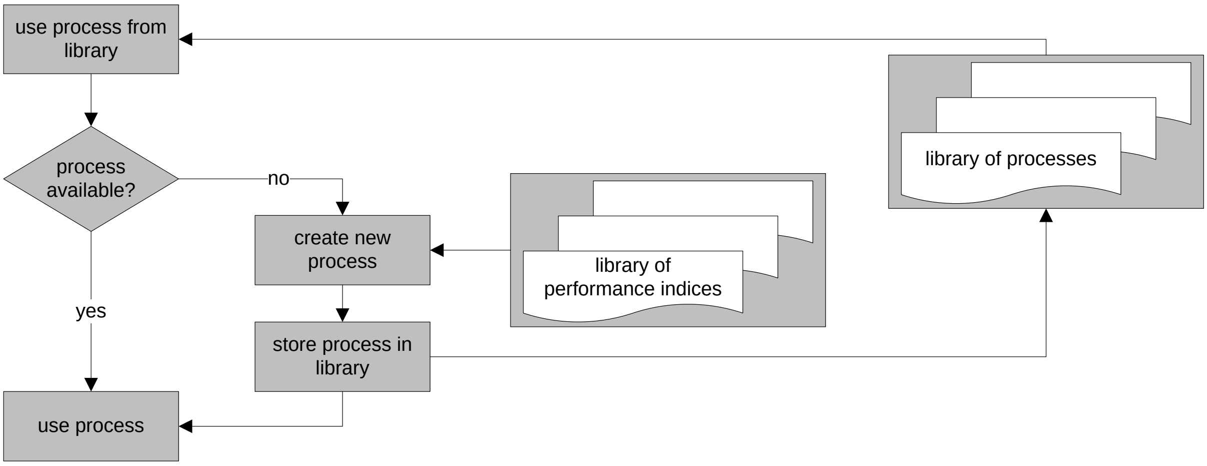

Figure G.3: Workflow for processes as libraries entries

Figure G.4 shows the workflow on using and creating performance index templates as library entries.

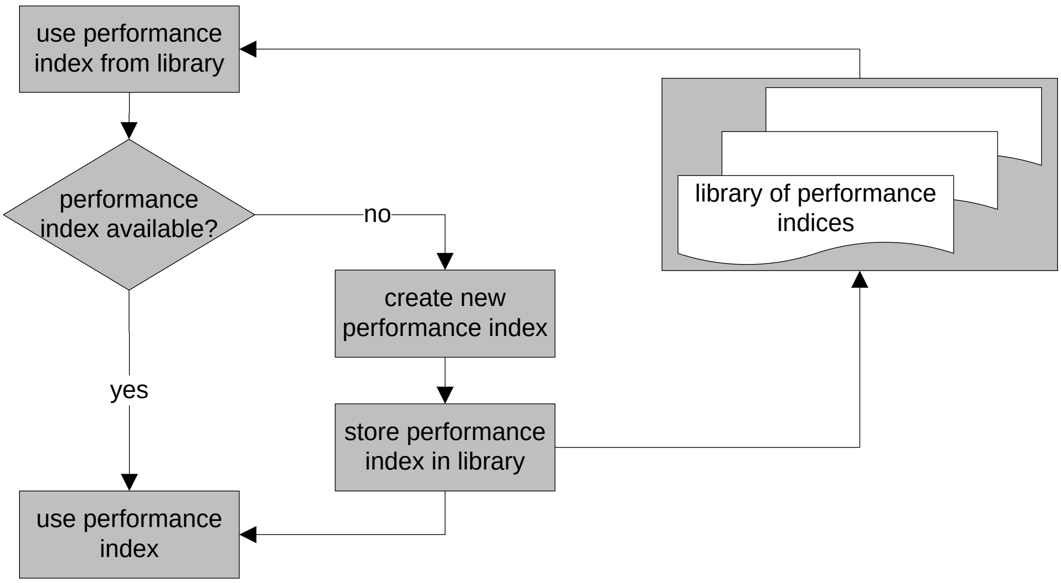

Figure G.4: Workflow for performance indices as library entries

Figure G.5 visualizes the library concept in its entirety.

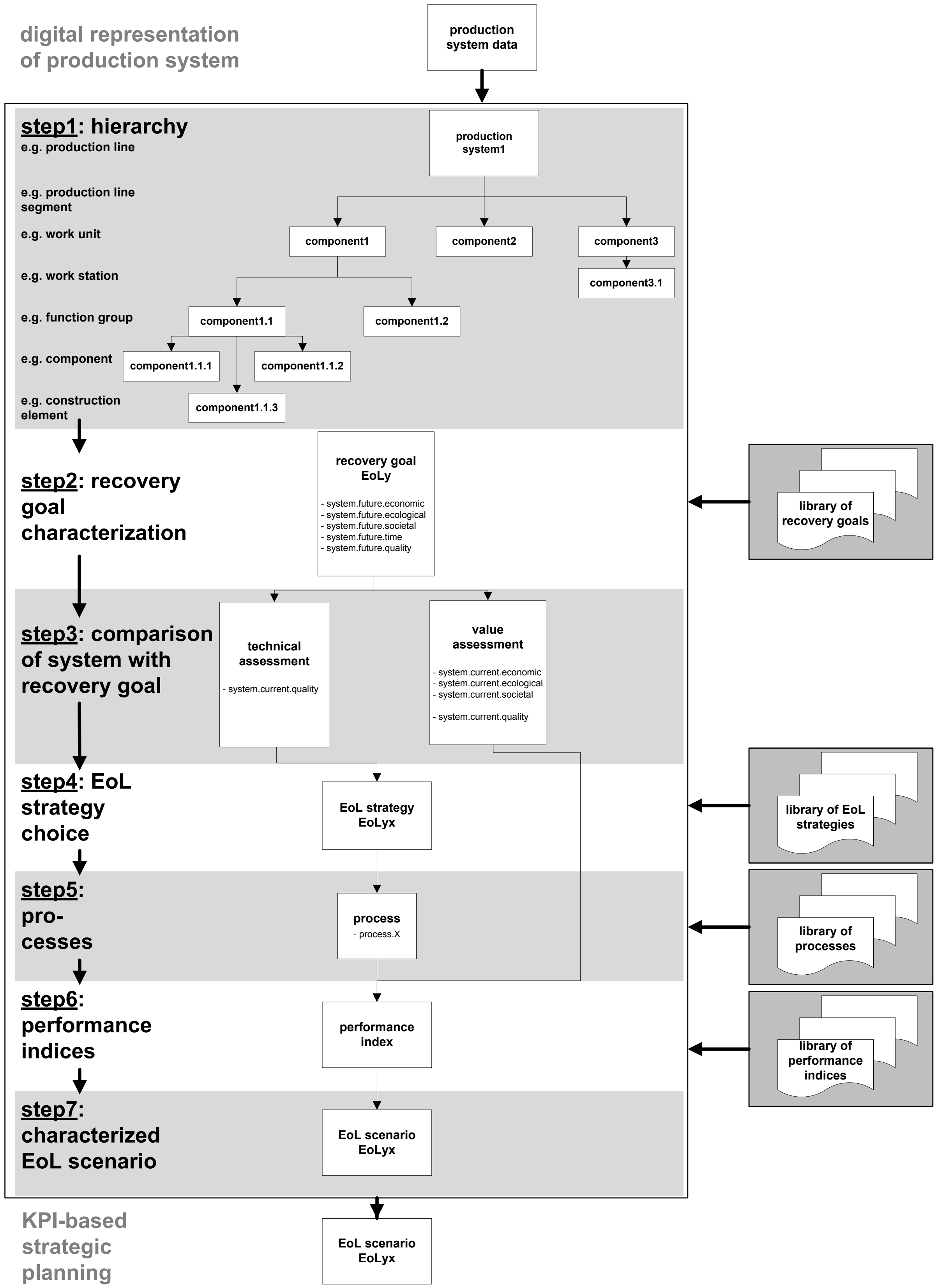

Figure G.5: Library concept for the recovery planning method

H XML representation of RoleClassLib of DIN 8591 disassembly processes

```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="DIN8580_Disassembly.aml" SchemaVersion="2.15"
    xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.
    org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.3.11.0</WriterVersion>
      <WriterRelease>4.3.11.0</WriterRelease>
      <LastWritingDateTime>2017-11-02T14:35:23.145+01:00</LastWritingDateTime>
      <WriterProjectTitle>Dissertation Nicole Schmidt OvGU</WriterProjectTitle>
      <WriterProjectID>Dissertation Nicole Schmidt OvGU</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0" />
  <RoleClassLib Name="ManufacturingProcesses_DIN8580_RoleClassLib">
    <Version>1.0.0</Version>
    <RoleClass Name="3-Separating" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole">
      <RoleClass Name="3-1-Severing" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating" />
      <RoleClass Name="3-2-MachiningWithGeometricallyDefinedCuttingEdges"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating
          " />
      <RoleClass Name="3-3-MachiningWithGeometricallyUndefinedCuttingEdges"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating
          " />
      <RoleClass Name="3-4-ChiplessMachining" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating" />
      <RoleClass Name="3-5-Diassembling" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating">
        <RoleClass Name="3-5-1-Dismantling" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
            Diassembling">
          <RoleClass Name="3-5-1-1-PickingUp" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling/3-5-1-Dismantling" />
          <RoleClass Name="3-5-1-2-TakingOut" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling/3-5-1-Dismantling" />
          <RoleClass Name="3-5-1-3-SlidingOut" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling/3-5-1-Dismantling" />
          <RoleClass Name="3-5-1-4-Detaching" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling/3-5-1-Dismantling" />
        </RoleClass>
        <RoleClass Name="3-5-2-Draining" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
            Diassembling">
          <RoleClass Name="3-5-2-1-Evacuating" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling/3-5-2-Draining" />
        </RoleClass>
        <RoleClass Name="3-5-3-UndoingForceClosureConnections" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
            Diassembling">
          <RoleClass Name="3-5-3-1-Unscrewing" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
```

```
                Diassembling/3-5-3-UndoingForceClosureConnections" />
            <RoleClass Name="3-5-3-2-Unclamping" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections" />
            <RoleClass Name="3-5-3-3-UndoingClipConnections" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections" />
            <RoleClass Name="3-5-3-4-UndoingCimpConnections" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections">
              <RoleClass Name="3-5-3-4-1-PressingOut" RefBaseClassPath="
                  ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                  Diassembling/3-5-3-UndoingForceClosureConnections/3-5-3-4-
                  UndoingCimpConnections" />
              <RoleClass Name="3-5-3-4-2-UndoingByExtension" RefBaseClassPath="
                  ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                  Diassembling/3-5-3-UndoingForceClosureConnections/3-5-3-4-
                  UndoingCimpConnections" />
              <RoleClass Name="3-5-3-4-3-UndoingByShrinkage" RefBaseClassPath="
                  ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                  Diassembling/3-5-3-UndoingForceClosureConnections/3-5-3-4-
                  UndoingCimpConnections" />
            </RoleClass>
            <RoleClass Name="3-5-3-5-UndoingNailConnections" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections" />
            <RoleClass Name="3-5-3-6-UndoingWedgeConnections" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections" />
            <RoleClass Name="3-5-3-7-UndoingByReleasing" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-3-UndoingForceClosureConnections" />
          </RoleClass>
          <RoleClass Name="3-5-4-DisassemblingOfThroughPrimaryShapingJoinedParts"
              RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/3-
              Separating/3-5-Diassembling">
            <RoleClass Name="3-5-4-1-MeltingOut" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-4-DisassemblingOfThroughPrimaryShapingJoinedParts" />
            <RoleClass Name="3-5-4-2-Skinning" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-4-DisassemblingOfThroughPrimaryShapingJoinedParts" />
          </RoleClass>
          <RoleClass Name="3-5-5-DisassembingOfThroughFormingJoinedParts" RefBaseClassPath=
              "ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling">
            <RoleClass Name="3-5-5-1-DisassemblingByUntwist" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-5-DisassembingOfThroughFormingJoinedParts" />
            <RoleClass Name="3-5-5-2-Unwinding" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-5-DisassembingOfThroughFormingJoinedParts" />
            <RoleClass Name="3-5-5-3-DisassemblingbyBend" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-5-DisassembingOfThroughFormingJoinedParts" />
            <RoleClass Name="3-5-5-4-UndoingRivetConnections" RefBaseClassPath="
                ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
                Diassembling/3-5-5-DisassembingOfThroughFormingJoinedParts" />
          </RoleClass>
          <RoleClass Name="3-5-7-Unsoldering" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling" />
          <RoleClass Name="3-5-8-UndoingAdhesiveConnections" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling" />
          <RoleClass Name="3-5-9-DisassemblingTextileConnections" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/3-Separating/3-5-
              Diassembling" />
        </RoleClass>
      </RoleClass>
    </RoleClassLib>
</CAEXFile>
```

I XML representation of RoleClassLib of preprocessing processes


```xml
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="Recycling.aml" SchemaVersion="2.15" xsi:noNamespaceSchemaLocation="
    CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.3.11.0</WriterVersion>
      <WriterRelease>4.3.11.0</WriterRelease>
      <LastWritingDateTime>2017-11-02T18:05:52.7344+01:00</LastWritingDateTime>
      <WriterProjectTitle>Dissertation Nicole Schmidt OvGU</WriterProjectTitle>
      <WriterProjectID>Dissertation Nicole Schmidt OvGU</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0" />
  <RoleClassLib Name="Recycling_Preprocessing_RoleClassLib">
    <Version>1.0.0</Version>
    <RoleClass Name="Mechanical" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole">
      <RoleClass Name="Shredding" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
          Mechanical">
        <RoleClass Name="task">
          <RoleClass Name="SizeReduction" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="Crushing" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="Feeding" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
        </RoleClass>
        <RoleClass Name="size">
          <RoleClass Name="CoarseShredding" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="MediumShredding" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="FineShredding" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="UltraFineShredding" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
        </RoleClass>
        <RoleClass Name="stress">
          <RoleClass Name="Pressure" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="Shock" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
              Mechanical/Shredding" />
          <RoleClass Name="Shear" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
              Mechanical/Shredding" />
          <RoleClass Name="ImpactA" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="ImpactB" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
          <RoleClass Name="Thrust" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib
              /Mechanical/Shredding" />
          <RoleClass Name="Cut" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
              Mechanical/Shredding" />
          <RoleClass Name="Tug" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
              Mechanical/Shredding" />
          <RoleClass Name="Bend" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
              Mechanical/Shredding" />
          <RoleClass Name="Torsion" RefBaseClassPath="
              Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
```

```
      <RoleClass Name="Rip" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
          Mechanical/Shredding" />
    </RoleClass>
    <RoleClass Name="solidness">
      <RoleClass Name="Hard" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
          Mechanical/Shredding" />
      <RoleClass Name="MediumHard" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
      <RoleClass Name="Soft" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
          Mechanical/Shredding" />
    </RoleClass>
    <RoleClass Name="deformation_behavior">
      <RoleClass Name="BrittleElastic" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
      <RoleClass Name="RubberElastic" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
      <RoleClass Name="ElasticPlastic" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
      <RoleClass Name="ElasticViscous" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Shredding" />
    </RoleClass>
  </RoleClass>
  <RoleClass Name="Classing" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib/
      Mechanical">
    <RoleClass Name="SieveClassification" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Classing">
      <RoleClass Name="WetScreening" />
      <RoleClass Name="DryScreening" />
    </RoleClass>
    <RoleClass Name="CurrentClassification" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Classing" />
  </RoleClass>
  <RoleClass Name="Segregating" RefBaseClassPath="
      Recycling_Preprocessing_RoleClassLib/Mechanical">
    <RoleClass Name="DenseSegregation" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating">
      <RoleClass Name="SinkFLoatSorting" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
          DenseSegregation" />
      <RoleClass Name="SettingProcess" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
          DenseSegregation" />
      <RoleClass Name="FlowSorting" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
          DenseSegregation">
        <RoleClass Name="AeroFlowSorter" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            DenseSegregation/FlowSorting" />
        <RoleClass Name="HydroFlowSorter" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            DenseSegregation/FlowSorting" />
      </RoleClass>
    </RoleClass>
    <RoleClass Name="MagneticSegregation" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating">
      <RoleClass Name="EddyCurrentSorting" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
          MagneticSegregation" />
    </RoleClass>
    <RoleClass Name="ElectricSegregation" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating" />
    <RoleClass Name="Flotation" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating" />
    <RoleClass Name="SegregationByDifferentMechanicalProperties" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating" />
    <RoleClass Name="PickingOut" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating" />
    <RoleClass Name="SensorBasedSegregation" RefBaseClassPath="
        Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating">
      <RoleClass Name="NearInfraredAnalysis" RefBaseClassPath="
          Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
          SensorBasedSegregation" />
      <RoleClass Name="OptoelectricIdentification" RefBaseClassPath="
```

```xml
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="ElectromagneticInduction" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="X-rayTransmissionMeasurement" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="X-rayFluorescence" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="AtomicEmissionSpectroscopy" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="InfraredSpectroscopy" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="InfraredLaserPulseThermography" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="PyrolysisGasChromatography" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="LaserPulsePyrolysisGasChromatography" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
          <RoleClass Name="CMYKRecognition" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Segregating/
            SensorBasedSegregation" />
        </RoleClass>
      </RoleClass>
      <RoleClass Name="Compacting" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib
          /Mechanical">
        <RoleClass Name="task">
          <RoleClass Name="Melting" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Compacting" />
          <RoleClass Name="Dissolving" RefBaseClassPath="
            Recycling_Preprocessing_RoleClassLib/Mechanical/Compacting" />
          <RoleClass Name="Gasify" RefBaseClassPath="Recycling_Preprocessing_RoleClassLib
            /Mechanical/Compacting" />
        </RoleClass>
      </RoleClass>
    </RoleClass>
    <RoleClass Name="PhysicalChemical" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole" />
  </RoleClassLib>
</CAEXFile>
```

J XML representation of InterfaceClassLib of DIN 8593 joining processes


```xml
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="DIN8580.aml" SchemaVersion="2.15" xsi:noNamespaceSchemaLocation="
    CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.3.11.0</WriterVersion>
      <WriterRelease>4.3.11.0</WriterRelease>
      <LastWritingDateTime>2017-10-09T23:44:44.4654002+02:00</LastWritingDateTime>
      <WriterProjectTitle>Dissertation Nicole Schmidt OvGU</WriterProjectTitle>
      <WriterProjectID>Dissertation Nicole Schmidt OvGU</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0" />
  <InterfaceClassLib Name="ManufacturingProcesses_DIN8580_InterfaceClassLib">
    <Version>0.0.3</Version>
    <InterfaceClass Name="4-Joining" RefBaseClassPath="AutomationMLInterfaceClassLib/
        AutomationMLBaseInterface">
      <InterfaceClass Name="4-1-Assembling" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
        <InterfaceClass Name="4-1-1-PuttingOn" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
        <InterfaceClass Name="4-1-2-PuttingIn" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
        <InterfaceClass Name="4-1-3-SlideIn" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
        <InterfaceClass Name="4-1-4-HookIn" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
        <InterfaceClass Name="4-1-5-Inserting" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
        <InterfaceClass Name="4-1-6-ElasticSpeading" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-1-Assembling" />
      </InterfaceClass>
      <InterfaceClass Name="4-2-Filling" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
        <InterfaceClass Name="4-2-1-FillingIn" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-2-Filling" />
        <InterfaceClass Name="4-2-2-Saturating_Impregnating" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-2-Filling" />
      </InterfaceClass>
      <InterfaceClass Name="4-3-JoiningByMechanicalMeans" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
        <InterfaceClass Name="4-3-1-Screwing" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
            JoiningByMechanicalMeans" />
        <InterfaceClass Name="4-3-2-Clamping" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
            JoiningByMechanicalMeans" />
        <InterfaceClass Name="4-3-3-Clipping" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
            JoiningByMechanicalMeans" />
        <InterfaceClass Name="4-3-4-JoiningByCrimpConnection" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
            JoiningByMechanicalMeans">
          <InterfaceClass Name="4-3-4-1-AssemblingByForceFittingPinning" RefBaseClassPath
              ="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
              JoiningByMechanicalMeans/4-3-4-JoiningByCrimpConnection" />
          <InterfaceClass Name="4-3-4-2-JoiningByShrinkage" RefBaseClassPath="
              ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
```

```
            JoiningByMechanicalMeans/4-3-4-JoiningByCrimpConnection" />
        <InterfaceClass Name="4-3-4-3-JoiningByExtension" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
            JoiningByMechanicalMeans/4-3-4-JoiningByCrimpConnection" />
      </InterfaceClass>
      <InterfaceClass Name="4-3-5-Nailing_DrivingIn" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
          JoiningByMechanicalMeans" />
      <InterfaceClass Name="4-3-6-Wedging" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
          JoiningByMechanicalMeans" />
      <InterfaceClass Name="4-3-7-Bracing" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-3-
          JoiningByMechanicalMeans" />
    </InterfaceClass>
    <InterfaceClass Name="4-4-JoiningByPrimaryShaping" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
      <InterfaceClass Name="4-4-1-Grouting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping" />
      <InterfaceClass Name="4-4-2-Embedding" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping">
        <InterfaceClass Name="4-4-2-1-InjectionMoulding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
            JoiningByPrimaryShaping/4-4-2-Embedding" />
        <InterfaceClass Name="4-4-2-2-Moulding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
            JoiningByPrimaryShaping/4-4-2-Embedding" />
        <InterfaceClass Name="4-4-2-3-JoiningByVulcanizing" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
            JoiningByPrimaryShaping/4-4-2-Embedding" />
      </InterfaceClass>
      <InterfaceClass Name="4-4-3-Socketing" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping" />
      <InterfaceClass Name="4-4-4-JoiningByElectroplating" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping" />
      <InterfaceClass Name="4-4-5-Covering" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping" />
      <InterfaceClass Name="4-4-6-Puttying" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-4-
          JoiningByPrimaryShaping" />
    </InterfaceClass>
    <InterfaceClass Name="4-5-JoiningByForming" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
      <InterfaceClass Name="4-5-1-FormingOfWiryBodies" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming">
        <InterfaceClass Name="4-5-1-1-BraidingOfWires" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-2-JointlyTwisting" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-3-Stranding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-4-Splicing" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-5-Knotting" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-6-JoiningByWinding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-7-WeavingOfWires" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
            JoiningByForming/4-5-1-FormingOfWiryBodies" />
        <InterfaceClass Name="4-5-1-8-JoiningByStaples" RefBaseClassPath="
```

```
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-1-FormingOfWiryBodies" />
</InterfaceClass>
<InterfaceClass Name="4-5-2-FormingOfPlatesTubesAndShapes" RefBaseClassPath="
    ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
    JoiningByForming">
  <InterfaceClass Name="4-5-2-1-JoiningByGroovingAndCenter-punches"
      RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
      Joining/4-5-JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-2-JointlyExtrusion" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-3-JointlyExtracting_Covering" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-4-JoiningByExpanding" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes">
    <InterfaceClass Name="4-5-2-4-1-TubeAssemblingByExpansionRolling"
        RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
        Joining/4-5-JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes/4-5-2-4-
        JoiningByExpanding" />
    <InterfaceClass Name="4-5-2-4-2-JoiningByWideningWithInternalPressure"
        RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
        Joining/4-5-JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes/4-5-2-4-
        JoiningByExpanding" />
  </InterfaceClass>
  <InterfaceClass Name="4-5-2-5-JoiningByTightening" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes">
    <InterfaceClass Name="4-5-2-5-1-JoiningByRotarySwaging" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
        JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes/4-5-2-5-
        JoiningByTightening" />
    <InterfaceClass Name="4-5-2-5-2-JoiningByNecking" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
        JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes/4-5-2-5-
        JoiningByTightening" />
    <InterfaceClass Name="4-5-2-5-3-JoiningByBeading" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
        JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes/4-5-2-5-
        JoiningByTightening" />
  </InterfaceClass>
  <InterfaceClass Name="4-5-2-6-JoiningByCrimping" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-7-JoiningByHemming" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-8-Winding_Wrapping" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-9-LockForming" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-10-FormingInjecting" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-11-Clinching" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-12-JoiningByPressing" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
  <InterfaceClass Name="4-5-2-13-JoiningBySqueezing" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
      JoiningByForming/4-5-2-FormingOfPlatesTubesAndShapes" />
</InterfaceClass>
<InterfaceClass Name="4-5-3-ProceduresOfRiveting" RefBaseClassPath="
    ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
    JoiningByForming">
  <InterfaceClass Name="4-5-3-1-Riveting" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
```

```
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
      <InterfaceClass Name="4-5-3-2-HollowRiveting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
      <InterfaceClass Name="4-5-3-3-PegRiveting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
      <InterfaceClass Name="4-5-3-4-TubularPegRiveting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
      <InterfaceClass Name="4-5-3-5-IntermediatePegRiveting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
      <InterfaceClass Name="4-5-3-6-PunchRiveting" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-5-
          JoiningByForming/4-5-3-ProceduresOfRiveting" />
    </InterfaceClass>
  </InterfaceClass>
  <InterfaceClass Name="4-6-Welding" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
    <InterfaceClass Name="4-6-1-PressureWelding" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding">
      <InterfaceClass Name="4-6-1-1-PressureWeldingBySolidBodies" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-2-PressureWeldingByLiquids" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-3-PressureWeldingByGas" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-4-PressureWeldingByElectricDischarge"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
          Joining/4-6-Welding/4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-5-PressureWeldingByRadiation" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-6-PressureWeldingByMovementOfAMass"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
          Joining/4-6-Welding/4-6-1-PressureWelding" />
      <InterfaceClass Name="4-6-1-7-PressureWeldingByElectricCurrent"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
          Joining/4-6-Welding/4-6-1-PressureWelding" />
    </InterfaceClass>
    <InterfaceClass Name="4-6-2-FusionWelding" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding">
      <InterfaceClass Name="4-6-2-2-FusionWeldingByLiquids" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-2-FusionWelding" />
      <InterfaceClass Name="4-6-2-3-FusionWeldingByGas" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-2-FusionWelding" />
      <InterfaceClass Name="4-6-2-3-FusionWeldingByElectricDischarge"
          RefBaseClassPath="ManufacturingProcesses_DIN8580_InterfaceClassLib/4-
          Joining/4-6-Welding/4-6-2-FusionWelding" />
      <InterfaceClass Name="4-6-2-5-FusionWeldingByRadiation" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-2-FusionWelding" />
      <InterfaceClass Name="4-6-2-7-FusionWeldingByElectricCurrent" RefBaseClassPath=
          "ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-6-Welding
          /4-6-2-FusionWelding" />
    </InterfaceClass>
  </InterfaceClass>
  <InterfaceClass Name="4-7-JoiningBySolderingOrBrazing" RefBaseClassPath="
      ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
    <InterfaceClass Name="4-7-1-Soldering" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
        JoiningBySolderingOrBrazing">
      <InterfaceClass Name="4-7-1-1-SolderingBySolidBodies" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
          JoiningBySolderingOrBrazing/4-7-1-Soldering" />
      <InterfaceClass Name="4-7-1-2-SolderingByLiquids" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
```

```
            JoiningBySolderingOrBrazing/4-7-1-Soldering" />
          <InterfaceClass Name="4-7-1-3-SolderingByGas" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-1-Soldering" />
          <InterfaceClass Name="4-7-1-5-SolderingByRadiation" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-1-Soldering" />
          <InterfaceClass Name="4-7-1-7-SolderingByElectricCurrent" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-1-Soldering" />
          <InterfaceClass Name="4-7-1-8-FurnaceSoldering" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-1-Soldering" />
        </InterfaceClass>
        <InterfaceClass Name="4-7-2-Brazing" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
          JoiningBySolderingOrBrazing">
          <InterfaceClass Name="4-7-2-2-BrazingByLiquids" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
          <InterfaceClass Name="4-7-2-3-BrazingByGas" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
          <InterfaceClass Name="4-7-2-4-BrazingByElectricDischarge" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
          <InterfaceClass Name="4-7-2-5-BrazingByRadiation" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
          <InterfaceClass Name="4-7-2-7-BrazingByElectricCurrent" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
          <InterfaceClass Name="4-7-2-8-HightemperatureBrazing" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-7-
            JoiningBySolderingOrBrazing/4-7-2-Brazing" />
        </InterfaceClass>
      </InterfaceClass>
      <InterfaceClass Name="4-8-JoiningByMeansOfAdhesives" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining">
        <InterfaceClass Name="4-8-1-SettingByPhysicalCombination" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
          JoiningByMeansOfAdhesives">
          <InterfaceClass Name="4-8-1-1-WetBonding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
            JoiningByMeansOfAdhesives/4-8-1-SettingByPhysicalCombination" />
          <InterfaceClass Name="4-8-1-2-ContactBonding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
            JoiningByMeansOfAdhesives/4-8-1-SettingByPhysicalCombination" />
          <InterfaceClass Name="4-8-1-3-ActivatedBonding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
            JoiningByMeansOfAdhesives/4-8-1-SettingByPhysicalCombination" />
          <InterfaceClass Name="4-8-1-4-Pressure-sensitiveBonding" RefBaseClassPath="
            ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
            JoiningByMeansOfAdhesives/4-8-1-SettingByPhysicalCombination" />
        </InterfaceClass>
        <InterfaceClass Name="4-8-2-SettingByChemicalsCombination" RefBaseClassPath="
          ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining/4-8-
          JoiningByMeansOfAdhesives" />
      </InterfaceClass>
      <InterfaceClass Name="4-9-TextileJoining" RefBaseClassPath="
        ManufacturingProcesses_DIN8580_InterfaceClassLib/4-Joining" />
    </InterfaceClass>
  </InterfaceClassLib>
</CAEXFile>
```

K Complete mapping of the End-of-Life data model onto AutomationML

This chapter shows at the glance the complete mapping of the base EoL data model (see Figure K.1), the EoL process data model (see Figure K.2), and the EoL data model (see Figure K.3) onto AML.

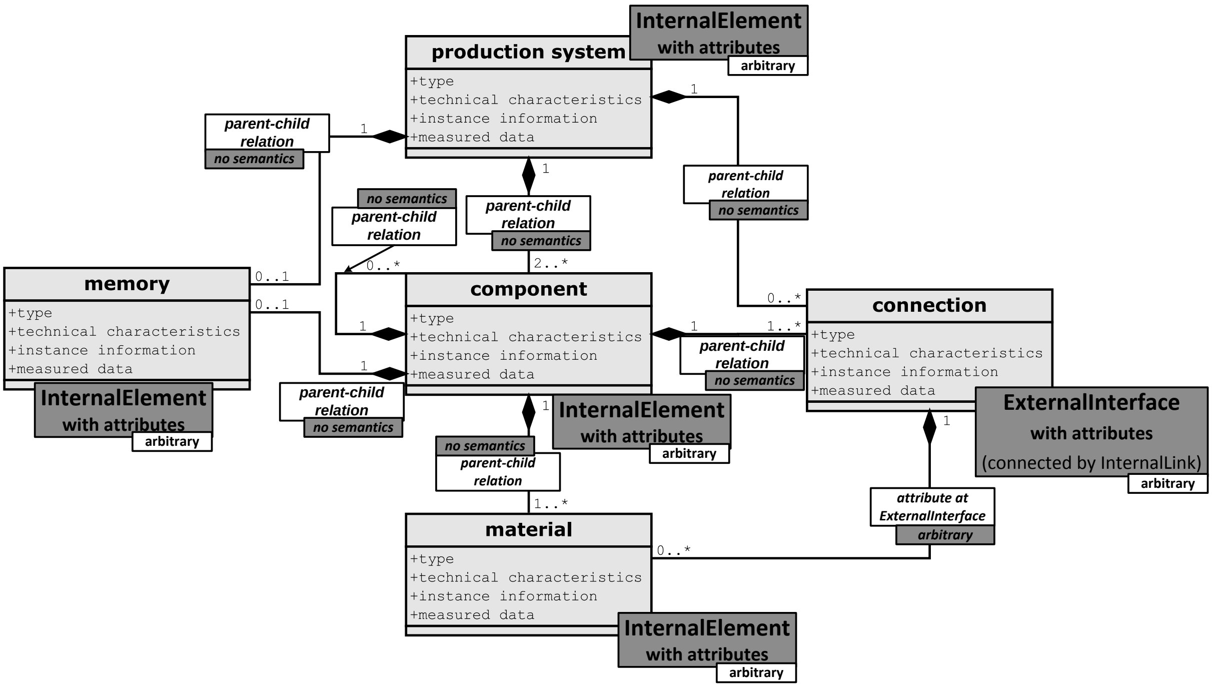

Figure K.1: Base End-of-Life data model (generalized) completely mapped onto AutomationML

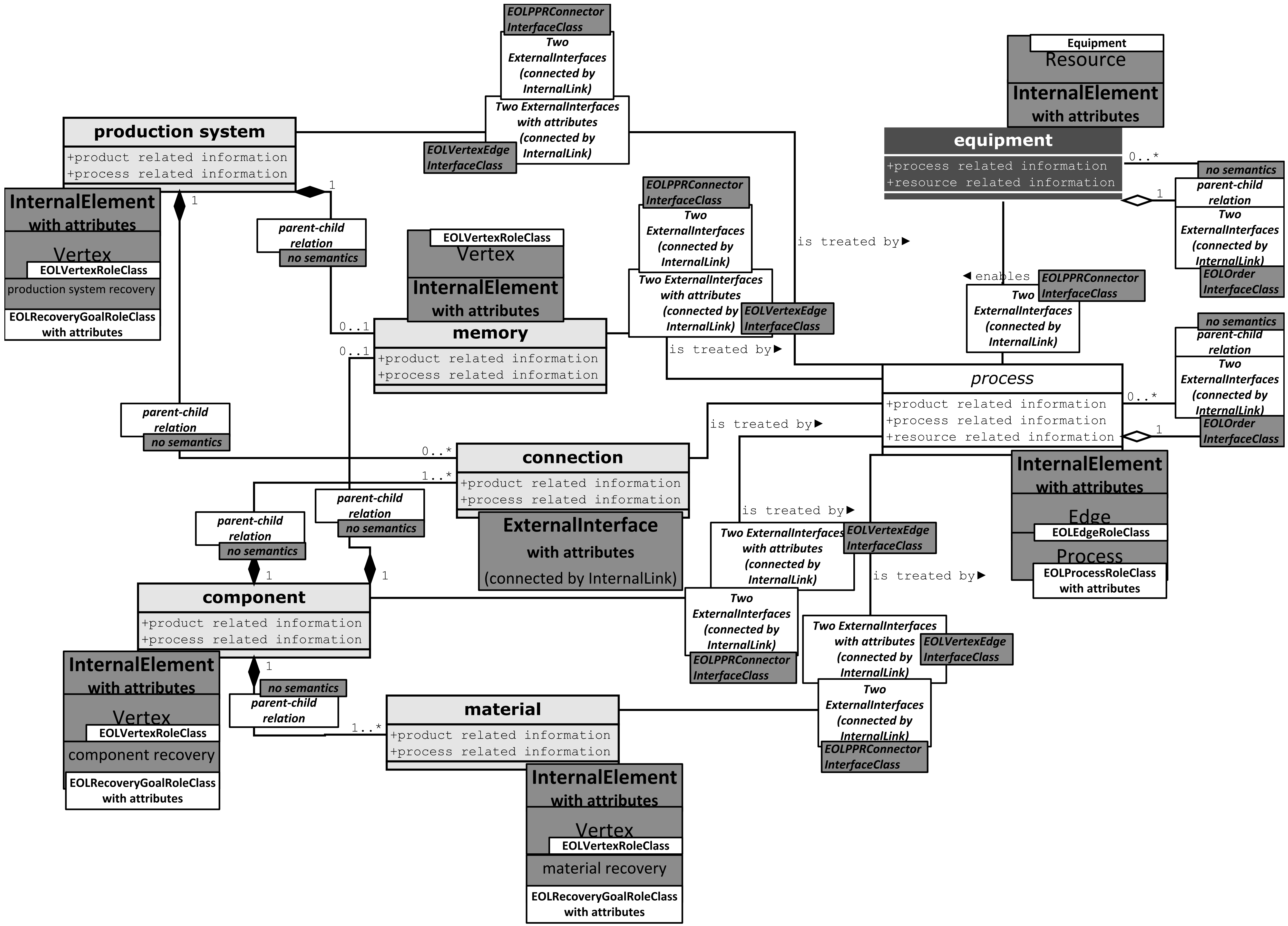

Figure K.2: End-of-Life process data model (unified and generalized) completely mapped onto AutomationML

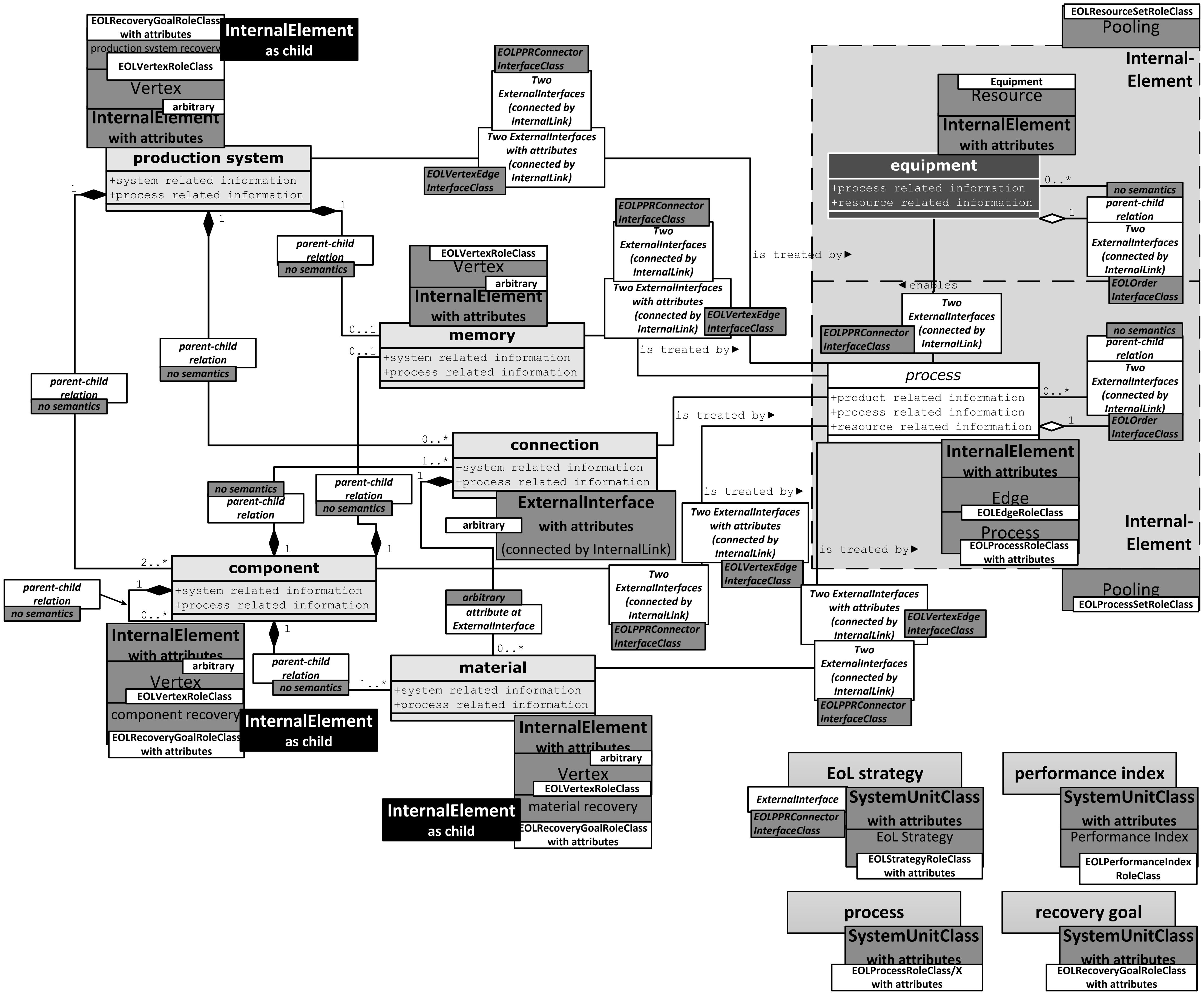

Figure K.3: End-of-Life data model (unified and generalized) completely mapped onto AutomationML

L XML representation of AutomationMLEOL libraries


```xml
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="AutomationMLEOLLibs.aml" SchemaVersion="2.15"
    xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.
    org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.3.11.0</WriterVersion>
      <WriterRelease>4.3.11.0</WriterRelease>
      <LastWritingDateTime>2017-10-15T20:21:49.765+02:00</LastWritingDateTime>
      <WriterProjectTitle>Dissertation Nicole Schmidt OvGU</WriterProjectTitle>
      <WriterProjectID>Dissertation Nicole Schmidt OvGU</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0" />
  <InterfaceClassLib Name="AutomationMLEOLInterfaceClassLib">
    <Version>1.0.0</Version>
    <InterfaceClass Name="EOLVertexEdgeInterfaceClass" RefBaseClassPath="
        AutomationMLGraphInterfaceClassLib/VertexEdgeInterface" />
    <InterfaceClass Name="EOLPPRConnectorInterfaceClass" RefBaseClassPath="
        AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PPRConnector" />
    <InterfaceClass Name="EOLOrderInterfaceClass" RefBaseClassPath="
        AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Order" />
  </InterfaceClassLib>
  <RoleClassLib Name="AutomationMLEOLRoleClassLib">
    <Version>1.0.0</Version>
    <RoleClass Name="EOLProcessRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole/Process">
      <Attribute Name="ProcessEconomic" />
      <Attribute Name="ProcessEcological" />
      <Attribute Name="ProcessSocietal" />
      <Attribute Name="ProcessTime" />
      <Attribute Name="ProcessQuality" />
    </RoleClass>
    <RoleClass Name="EOLResourceRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib
        /AutomationMLBaseRole/Resource"></RoleClass>
    <RoleClass Name="EOLVertexRoleClass" RefBaseClassPath="AutomatioMLGraphRoleClassLib/
        Vertex" />
    <RoleClass Name="EOLEdgeRoleClass" RefBaseClassPath="AutomatioMLGraphRoleClassLib/
        Edge" />
    <RoleClass Name="EOLRecoveryGoalRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
      <Attribute Name="SystemFutureEconomic" />
      <Attribute Name="SystemFutureEcological" />
      <Attribute Name="SystemFutureSocietal" />
      <Attribute Name="SystemFutureTime" />
      <Attribute Name="SystemFutureQuality" />
    </RoleClass>
    <RoleClass Name="EOLProcessSetRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure/ProcessStructure" />
    <RoleClass Name="EOLResourceSetRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure/ResourceStructure" />
    <RoleClass Name="EOLStrategyRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib
        /AutomationMLBaseRole">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
    </RoleClass>
    <RoleClass Name="EOLPerformanceIndexRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
```

```xml
</RoleClassLib>
<SystemUnitClassLib Name="AutomationMLEOLRecoveryGoalSystemUnitClassLib">
  <Version>1.0.0</Version>
  <SystemUnitClass Name="ProductionSystemRecovery">
    <Attribute Name="SystemCurrentEconomic" />
    <Attribute Name="SystemCurrentEcological" />
    <Attribute Name="SystemCurrentSocietal" />
    <Attribute Name="SystemCurrentQuality" />
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLRecoveryGoalRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="ComponentRecovery">
    <Attribute Name="SystemCurrentEconomic" />
    <Attribute Name="SystemCurrentEcological" />
    <Attribute Name="SystemCurrentSocietal" />
    <Attribute Name="SystemCurrentQuality" />
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLRecoveryGoalRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="MaterialRecovery">
    <Attribute Name="SystemCurrentEconomic" />
    <Attribute Name="SystemCurrentEcological" />
    <Attribute Name="SystemCurrentSocietal" />
    <Attribute Name="SystemCurrentQuality" />
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLRecoveryGoalRoleClass" />
  </SystemUnitClass>
</SystemUnitClassLib>
<SystemUnitClassLib Name="AutomationMLEOLProcessSystemUnitClassLib">
  <Version>1.0.0</Version>
  <SystemUnitClass Name="Disassembly">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Reconditioning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Preprocessing">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
```

```
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Storage">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Reassembly">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Cleaning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Decommissioning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
```

```
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Testing">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Commissioning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Transportation">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
```

```xml
    </SystemUnitClass>
    <SystemUnitClass Name="Disposal">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Repair">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLPerformanceIndicesSystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="EconomicImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="EcologicalImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="SocietalImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="TimeImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="QualityImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLStrategySystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="DirectReuse">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
```

```
    </SystemUnitClass>
    <SystemUnitClass Name="ReuseAfterRepair">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="ReuseAfterRemanufacturing">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Recycling">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
</CAEXFile>
```

M XML representation of simple example structure


```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="EOLDataModel5_6_7.aml" SchemaVersion="2.15"
    xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd" xmlns:xsi="http://www.w3.
    org/2001/XMLSchema-instance">
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>AutomationML Editor</WriterName>
      <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
      <WriterVendor>AutomationML e.V.</WriterVendor>
      <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
      <WriterVersion>4.3.11.0</WriterVersion>
      <WriterRelease>4.3.11.0</WriterRelease>
      <LastWritingDateTime>2017-10-15T20:33:37.287+02:00</LastWritingDateTime>
      <WriterProjectTitle>Dissertation Nicole Schmidt OvGU</WriterProjectTitle>
      <WriterProjectID>Dissertation Nicole Schmidt OvGU</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <AdditionalInformation AutomationMLVersion="2.0" />
  <InstanceHierarchy Name="Factory">
    <Version>1.0.0</Version>
    <InternalElement Name="productionsystem1" ID="8bedc2a0-2f47-44a6-b7a6-3fec54c72e64"
        RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
      <ExternalInterface Name="mechanicalInterfaceToComp1" ID="bd68ed13-62e9-46e1-b506-5
          bc6bb4c18e1" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <Attribute Name="material">
          <Value>arbitrary</Value>
        </Attribute>
      </ExternalInterface>
      <ExternalInterface Name="mechanicalInterfaceToComp2" ID="0fa5348c-7669-4952-821d-2
          c92f62e8711" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <Attribute Name="material">
          <Value>arbitrary</Value>
        </Attribute>
      </ExternalInterface>
      <ExternalInterface Name="mechanicalInterfaceToComp3" ID="fb8fcd09-1724-4f0b-989b-
          bfa37bad35b1" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <Attribute Name="material">
          <Value>arbitrary</Value>
        </Attribute>
      </ExternalInterface>
      <ExternalInterface Name="EOLVertexEdgeInterfaceClass_instance_1" RefBaseClassPath="
          AutomationMLEOLInterfaceClassLib/EOLVertexEdgeInterfaceClass" ID="1cdccaed
          -4411-4c2e-a8d8-711cc60186ff">
        <Attribute Name="Direction" AttributeDataType="xs:string">
          <Value>Out</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="component1" ID="a033b1f5-6947-4bd8-85cc-2fa939b00f01"
          RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <ExternalInterface Name="mechanicalInterfaceToPS1" ID="dc337a48-d583-4b85-abf7-24
            c13a5ef204" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
          <Attribute Name="material">
            <Value>arbitrary</Value>
          </Attribute>
        </ExternalInterface>
        <InternalElement Name="component1_1" ID="3ef5df55-6346-43d5-bd0f-9eaacd2d58b5"
            RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
          <InternalElement Name="materialA" ID="bc3cf605-4e66-423a-98de-fbf6f4e69db0"
              RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
            <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/
                arbitrary" />
          </InternalElement>
          <InternalElement Name="materialB" ID="087295b1-120c-448f-ba75-eb8c66645579"
              RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
```

```xml
        <ExternalInterface Name="EOLVertexEdgeInterfaceClass_instance_1"
            RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
            EOLVertexEdgeInterfaceClass" ID="4b95848d-d725-4682-8e49-8cf17fc701fc">
          <Attribute Name="Direction" AttributeDataType="xs:string">
            <Value>In</Value>
          </Attribute>
        </ExternalInterface>
        <InternalElement Name="MaterialRecovery_instance_1" RefBaseSystemUnitPath="
            AutomationMLEOLRecoveryGoalSystemUnitClassLib/MaterialRecovery" ID="
            a48c4194-25fe-4af8-925f-a19501b87207">
          <Attribute Name="SystemCurrentEconomic" AttributeDataType="xs:decimal">
            <Value>6</Value>
          </Attribute>
          <Attribute Name="SystemCurrentEcological" AttributeDataType="xs:decimal">
            <Value>6</Value>
          </Attribute>
          <Attribute Name="SystemCurrentSocietal" AttributeDataType="xs:decimal">
            <Value>6</Value>
          </Attribute>
          <Attribute Name="SystemCurrentQuality" AttributeDataType="xs:decimal">
            <Value>6</Value>
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
              EOLRecoveryGoalRoleClass">
            <Attribute Name="SystemFutureEconomic" AttributeDataType="xs:decimal">
              <Value>7</Value>
            </Attribute>
            <Attribute Name="SystemFutureEcological" AttributeDataType="xs:decimal">
              <Value>7</Value>
            </Attribute>
            <Attribute Name="SystemFutureSocietal" AttributeDataType="xs:decimal">
              <Value>7</Value>
            </Attribute>
            <Attribute Name="SystemFutureTime" AttributeDataType="xs:decimal">
              <Value>7</Value>
            </Attribute>
            <Attribute Name="SystemFutureQuality" AttributeDataType="xs:decimal">
              <Value>7</Value>
            </Attribute>
          </RoleRequirements>
          <MappingObject>
            <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEconomic"
                RoleAttributeName="SystemFutureEconomic" />
            <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEcological"
                RoleAttributeName="SystemFutureEcological" />
            <AttributeNameMapping SystemUnitAttributeName="SystemCurrentSocietal"
                RoleAttributeName="SystemFutureSocietal" />
            <AttributeNameMapping SystemUnitAttributeName="SystemCurrentQuality"
                RoleAttributeName="SystemFutureQuality" />
          </MappingObject>
        </InternalElement>
        <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
            EOLVertexRoleClass" />
        <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/
            arbitrary" />
      </InternalElement>
      <InternalElement Name="materialC" ID="ff73de15-8e69-42c4-b4e1-b2f3134e1f0f"
          RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/
            arbitrary" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary"
           />
    </InternalElement>
    <InternalElement Name="component1_2" ID="dcdb4ee5-fe32-40ae-9fe7-390c4db23dd7"
        RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
      <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary"
           />
    </InternalElement>
    <InternalElement Name="SpecificComponentRecovery_instance_1"
        RefBaseSystemUnitPath="AutomationMLEOLRecoveryGoalSystemUnitClassLib/
        ComponentRecovery" ID="472ee402-3749-486d-b666-bfdf5fe299cb">
      <Attribute Name="SystemCurrentEconomic" AttributeDataType="xs:decimal">
```

```
          <Value>5</Value>
        </Attribute>
        <Attribute Name="SystemCurrentEcological" AttributeDataType="xs:decimal">
          <Value>5</Value>
        </Attribute>
        <Attribute Name="SystemCurrentSocietal" AttributeDataType="xs:decimal">
          <Value>5</Value>
        </Attribute>
        <Attribute Name="SystemCurrentQuality" AttributeDataType="xs:decimal">
          <Value>5</Value>
        </Attribute>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
            EOLRecoveryGoalRoleClass">
          <Attribute Name="SystemFutureEconomic" AttributeDataType="xs:decimal">
            <Value>3</Value>
          </Attribute>
          <Attribute Name="SystemFutureEcological" AttributeDataType="xs:decimal">
            <Value>3</Value>
          </Attribute>
          <Attribute Name="SystemFutureSocietal" AttributeDataType="xs:decimal">
            <Value>3</Value>
          </Attribute>
          <Attribute Name="SystemFutureTime" AttributeDataType="xs:decimal">
            <Value>3</Value>
          </Attribute>
          <Attribute Name="SystemFutureQuality" AttributeDataType="xs:decimal">
            <Value>3</Value>
          </Attribute>
        </RoleRequirements>
        <MappingObject>
          <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEconomic"
              RoleAttributeName="SystemFutureEconomic" />
          <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEcological"
              RoleAttributeName="SystemFutureEcological" />
          <AttributeNameMapping SystemUnitAttributeName="SystemCurrentSocietal"
              RoleAttributeName="SystemFutureSocietal" />
          <AttributeNameMapping SystemUnitAttributeName="SystemCurrentQuality"
              RoleAttributeName="SystemFutureQuality" />
        </MappingObject>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary" /
          >
    </InternalElement>
    <InternalElement Name="component2" ID="026ee03e-c73f-4bd7-b350-4efd329b3006"
        RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
      <ExternalInterface Name="mechanicalInterfaceToPS1" ID="23249a12-07b9-4627-9249-8
          fdef20fa124" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <Attribute Name="material">
          <Value>arbitrary</Value>
        </Attribute>
      </ExternalInterface>
      <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary" /
          >
    </InternalElement>
    <InternalElement Name="component3" ID="1b8ee145-a365-4bfd-87a6-7154db550476"
        RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
      <ExternalInterface Name="mechanicalInterfaceToPS1" ID="f05ec37f-d953-42be-b127-5
          c6f0331dc91" RefBaseClassPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <Attribute Name="material">
          <Value>arbitrary</Value>
        </Attribute>
      </ExternalInterface>
      <InternalElement Name="materialA" ID="92616c62-3a5f-4347-a440-d13e9059576f"
          RefBaseSystemUnitPath="PreviousLifeCyclePhasesRCLib/arbitrary">
        <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary"
            />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary" /
          >
    </InternalElement>
    <InternalElement Name="SpecificProductionSystemRecovery_instance_1"
        RefBaseSystemUnitPath="AutomationMLEOLRecoveryGoalSystemUnitClassLib/
        ProductionSystemRecovery" ID="86352905-6e18-49b7-b328-f1d2fb865dc6">
```

```
      <Attribute Name="SystemCurrentEconomic" AttributeDataType="xs:decimal">
        <Value>11</Value>
      </Attribute>
      <Attribute Name="SystemCurrentEcological" AttributeDataType="xs:decimal">
        <Value>11</Value>
      </Attribute>
      <Attribute Name="SystemCurrentSocietal" AttributeDataType="xs:decimal">
        <Value>11</Value>
      </Attribute>
      <Attribute Name="SystemCurrentQuality" AttributeDataType="xs:decimal">
        <Value>11</Value>
      </Attribute>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLRecoveryGoalRoleClass">
        <Attribute Name="SystemFutureEconomic" AttributeDataType="xs:decimal">
          <Value>15</Value>
        </Attribute>
        <Attribute Name="SystemFutureEcological" AttributeDataType="xs:decimal">
          <Value>15</Value>
        </Attribute>
        <Attribute Name="SystemFutureSocietal" AttributeDataType="xs:decimal">
          <Value>15</Value>
        </Attribute>
        <Attribute Name="SystemFutureTime" AttributeDataType="xs:decimal">
          <Value>15</Value>
        </Attribute>
        <Attribute Name="SystemFutureQuality" AttributeDataType="xs:decimal">
          <Value>15</Value>
        </Attribute>
      </RoleRequirements>
      <MappingObject>
        <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEconomic"
            RoleAttributeName="SystemFutureEconomic" />
        <AttributeNameMapping SystemUnitAttributeName="SystemCurrentEcological"
            RoleAttributeName="SystemFutureEcological" />
        <AttributeNameMapping SystemUnitAttributeName="SystemCurrentSocietal"
            RoleAttributeName="SystemFutureSocietal" />
        <AttributeNameMapping SystemUnitAttributeName="SystemCurrentQuality"
            RoleAttributeName="SystemFutureQuality" />
      </MappingObject>
    </InternalElement>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLVertexRoleClass" />
    <InternalLink Name="Link_mechIFComp1ToPS1" RefPartnerSideA="8bedc2a0-2f47-44a6-b7a6
        -3fec54c72e64:mechanicalInterfaceToComp1" RefPartnerSideB="a033b1f5-6947-4bd8
        -85cc-2fa939b00f01:mechanicalInterfaceToPS1" />
    <InternalLink Name="Link_mechIFComp2ToPS1" RefPartnerSideA="8bedc2a0-2f47-44a6-b7a6
        -3fec54c72e64:mechanicalInterfaceToComp2" RefPartnerSideB="026ee03e-c73f-4bd7-
        b350-4efd329b3006:mechanicalInterfaceToPS1" />
    <InternalLink Name="Link_mechIFComp3ToPS1" RefPartnerSideA="8bedc2a0-2f47-44a6-b7a6
        -3fec54c72e64:mechanicalInterfaceToComp3" RefPartnerSideB="1b8ee145-a365-4bfd
        -87a6-7154db550476:mechanicalInterfaceToPS1" />
    <InternalLink Name="Link_VertexPS1ToEdgeRecycling" RefPartnerSideA="6e3668f6-b8d0
        -424e-bd21-fa570bfcc48f:EOLVertexEdgeInterfaceClass_instance_1" RefPartnerSideB
        ="8bedc2a0-2f47-44a6-b7a6-3fec54c72e64:EOLVertexEdgeInterfaceClass_instance_1"
        />
    <RoleRequirements RefBaseRoleClassPath="PreviousLifeCyclePhasesRCLib/arbitrary" />
  </InternalElement>
  <InternalElement Name="EOLProcesses" ID="03e62a76-dfd4-476b-8724-7706ddebfa30">
    <InternalElement Name="Recycling_instance_1" RefBaseSystemUnitPath="
        AutomationMLEOLStrategySystemUnitClassLib/Recycling" ID="6e3668f6-b8d0-424e-
        bd21-fa570bfcc48f">
      <Attribute Name="EOLStrategyEconomic" AttributeDataType="xs:decimal">
        <Value>20</Value>
      </Attribute>
      <Attribute Name="EOLStrategyEcological" AttributeDataType="xs:decimal">
        <Value>20</Value>
      </Attribute>
      <Attribute Name="EOLStrategySocietal" AttributeDataType="xs:decimal">
        <Value>20</Value>
      </Attribute>
      <Attribute Name="EOLStrategyTime" AttributeDataType="xs:decimal">
        <Value>20</Value>
```

```
</Attribute>
<Attribute Name="EOLStrategyQuality" AttributeDataType="xs:decimal">
  <Value>20</Value>
</Attribute>
<ExternalInterface Name="EOLPPRConnectorInterfaceClass_instance_1"
    RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
    EOLPPRConnectorInterfaceClass" ID="ea19071f-6309-4069-966e-58074b1d603e" />
<ExternalInterface Name="EOLVertexEdgeInterfaceClass_instance_1" RefBaseClassPath
    ="AutomationMLEOLInterfaceClassLib/EOLVertexEdgeInterfaceClass" ID="c8f71e22
    -4e83-4ed8-861d-7fefdddf10e3">
  <Attribute Name="Direction" AttributeDataType="xs:string">
    <Value>In</Value>
  </Attribute>
</ExternalInterface>
<ExternalInterface Name="EOLVertexEdgeInterfaceClass_instance_2" RefBaseClassPath
    ="AutomationMLEOLInterfaceClassLib/EOLVertexEdgeInterfaceClass" ID="2ed3bbad
    -2ed2-46c5-a179-5aa961f512c6">
  <Attribute Name="Direction" AttributeDataType="xs:string">
    <Value>Out</Value>
  </Attribute>
</ExternalInterface>
<InternalElement Name="Decommissioning_instance_1" RefBaseSystemUnitPath="
    AutomationMLEOLProcessSystemUnitClassLib/Decommissioning" ID="37dc900f-f231-4
    eda-bf5a-70415cbb4ad1">
  <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <ExternalInterface Name="InterfaceOutToDisassembly" ID="d3122736-fa02-47b4-b3f5
      -1f5028dfa35c" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
      EOLOrderInterfaceClass">
    <Attribute Name="Direction" AttributeDataType="xs:string">
      <Value>Out</Value>
    </Attribute>
  </ExternalInterface>
  <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
      EOLProcessRoleClass" />
</InternalElement>
<InternalElement Name="Disassembly_instance_1" RefBaseSystemUnitPath="
    AutomationMLEOLProcessSystemUnitClassLib/Disassembly" ID="82793046-0de8-4526-
    aef4-e89987b1ef88">
  <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
    <Value>5</Value>
  </Attribute>
  <ExternalInterface Name="InterfaceInFromDecommissioning" ID="12607dc9-c2bf-433b
      -ade1-5e9ab7ee4eb5" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
      EOLOrderInterfaceClass">
    <Attribute Name="Direction" AttributeDataType="xs:string">
      <Value>In</Value>
    </Attribute>
```

```
    </ExternalInterface>
    <ExternalInterface Name="InterfaceOutToTransportation" ID="49a4a531-d64b-455f
        -870d-bdd61d598680" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
        EOLOrderInterfaceClass">
      <Attribute Name="Direction" AttributeDataType="xs:string">
        <Value>Out</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </InternalElement>
  <InternalElement Name="Transportation_instance_1" RefBaseSystemUnitPath="
      AutomationMLEOLProcessSystemUnitClassLib/Transportation" ID="a521b5f0
      -3803-472b-b979-7e78ebb3a056">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <ExternalInterface Name="InterfaceInFromDisassembly" ID="7e190fad-34a6-4e0f-827
        b-2563e34cb52b" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
        EOLOrderInterfaceClass">
      <Attribute Name="Direction" AttributeDataType="xs:string">
        <Value>In</Value>
      </Attribute>
    </ExternalInterface>
    <ExternalInterface Name="InterfaceOutPreprocessing" ID="e898f891-8736-4130-b3f2
        -09d9939ca525" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
        EOLOrderInterfaceClass">
      <Attribute Name="Direction" AttributeDataType="xs:string">
        <Value>Out</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </InternalElement>
  <InternalElement Name="Preprocessing_instance_1" RefBaseSystemUnitPath="
      AutomationMLEOLProcessSystemUnitClassLib/Preprocessing" ID="3dc1d923-72e7-4
      c98-8218-c6db8189a6e0">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <ExternalInterface Name="InterfaceInFromTransportation" ID="31356c84-9223-477e
        -8682-f7fff6672628" RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
        EOLOrderInterfaceClass">
      <Attribute Name="Direction" AttributeDataType="xs:string">
        <Value>In</Value>
      </Attribute>
    </ExternalInterface>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
```

```
      </InternalElement>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLEdgeRoleClass" />
      <InternalLink Name="Link_RecyclingToEquipment" RefPartnerSideA="6e3668f6-b8d0-424
          e-bd21-fa570bfcc48f:EOLPPRConnectorInterfaceClass_instance_1" RefPartnerSideB
          ="cbfbc1c1-1cf1-4185-96a9-74
          ffe31242ba:EOLPPRConnectorInterfaceClass_instance_1" />
      <InternalLink Name="Link_VertexMatB1ToEdgeRecycling" RefPartnerSideA="087295b1
          -120c-448f-ba75-eb8c66645579:EOLVertexEdgeInterfaceClass_instance_1"
          RefPartnerSideB="6e3668f6-b8d0-424e-bd21-
          fa570bfcc48f:EOLVertexEdgeInterfaceClass_instance_2" />
      <InternalLink Name="Link_DecommissioningToDisassembly" RefPartnerSideA="37dc900f-
          f231-4eda-bf5a-70415cbb4ad1:InterfaceOutToDisassembly" RefPartnerSideB="
          82793046-0de8-4526-aef4-e89987b1ef88:InterfaceInFromDecommissioning" />
      <InternalLink Name="Link_DisassemblyToTransportation" RefPartnerSideA="82793046-0
          de8-4526-aef4-e89987b1ef88:InterfaceOutToTransportation" RefPartnerSideB="
          a521b5f0-3803-472b-b979-7e78ebb3a056:InterfaceInFromDisassembly" />
      <InternalLink Name="Link_TransportationToPreprocessing" RefPartnerSideA="a521b5f0
          -3803-472b-b979-7e78ebb3a056:InterfaceOutPreprocessing" RefPartnerSideB="3
          dc1d923-72e7-4c98-8218-c6db8189a6e0:InterfaceInFromTransportation" />
      <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessSetRoleClass" />
  </InternalElement>
  <InternalElement Name="EOLResources" ID="38f4fefd-31bc-4dbc-9317-9a8f6eada4b9">
    <InternalElement Name="Equipment" ID="cbfbc1c1-1cf1-4185-96a9-74ffe31242ba">
      <ExternalInterface Name="EOLPPRConnectorInterfaceClass_instance_1"
          RefBaseClassPath="AutomationMLEOLInterfaceClassLib/
          EOLPPRConnectorInterfaceClass" ID="213189ea-176e-4fb7-a6cd-9baaa27e82ff" />
      <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLResourceRoleClass" />
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLResourceSetRoleClass" />
  </InternalElement>
</InstanceHierarchy>
<InterfaceClassLib Name="AutomationMLInterfaceClassLib">
  <Description>Standard Automation Markup Language Interface Class Library</Description
      >
  <Version>2.2.0</Version>
  <InterfaceClass Name="AutomationMLBaseInterface">
    <InterfaceClass Name="Order" RefBaseClassPath="AutomationMLBaseInterface">
      <Attribute Name="Direction" AttributeDataType="xs:string" />
    </InterfaceClass>
    <InterfaceClass Name="PortConnector" RefBaseClassPath="AutomationMLBaseInterface" /
        >
    <InterfaceClass Name="InterlockingConnector" RefBaseClassPath="
        AutomationMLBaseInterface" />
    <InterfaceClass Name="PPRConnector" RefBaseClassPath="AutomationMLBaseInterface" />
    <InterfaceClass Name="ExternalDataConnector" RefBaseClassPath="
        AutomationMLBaseInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI" />
      <InterfaceClass Name="COLLADAInterface" RefBaseClassPath="ExternalDataConnector"
          />
      <InterfaceClass Name="PLCopenXMLInterface" RefBaseClassPath="
          ExternalDataConnector" />
    </InterfaceClass>
    <InterfaceClass Name="Communication" RefBaseClassPath="AutomationMLBaseInterface">
      <InterfaceClass Name="SignalInterface" RefBaseClassPath="Communication" />
    </InterfaceClass>
  </InterfaceClass>
</InterfaceClassLib>
<InterfaceClassLib Name="AutomationMLGraphInterfaceClassLib">
  <InterfaceClass Name="VertexEdgeInterface" RefBaseClassPath="
      AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Order" />
  <InterfaceClass Name="MultiEdgeConnectionPoint" RefBaseClassPath="
      AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PortConnector">
    <Attribute Name="Direction" AttributeDataType="xs:string">
```

```
        <Description>The attribute Direction shall be used in order to specify the
            direction. Permitted values are In, Out or InOut. </Description>
      </Attribute>
    </InterfaceClass>
  </InterfaceClassLib>
  <InterfaceClassLib Name="AutomationMLEOLInterfaceClassLib">
    <Version>1.0.0</Version>
    <InterfaceClass Name="EOLVertexEdgeInterfaceClass" RefBaseClassPath="
        AutomationMLGraphInterfaceClassLib/VertexEdgeInterface" />
    <InterfaceClass Name="EOLPPRConnectorInterfaceClass" RefBaseClassPath="
        AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PPRConnector" />
    <InterfaceClass Name="EOLOrderInterfaceClass" RefBaseClassPath="
        AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Order" />
  </InterfaceClassLib>
  <InterfaceClassLib Name="PreviousLifeCyclePhasesIFCLib">
    <Version>1.0.0</Version>
    <InterfaceClass Name="arbitrary" />
  </InterfaceClassLib>
  <RoleClassLib Name="AutomationMLBaseRoleClassLib">
    <Description>Automation Markup Language base role class library</Description>
    <Version>2.2.0</Version>
    <RoleClass Name="AutomationMLBaseRole">
      <RoleClass Name="Group" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole">
        <Attribute Name="AssociatedFacet" AttributeDataType="xs:string" />
      </RoleClass>
      <RoleClass Name="Facet" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole" />
      <RoleClass Name="Port" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole">
        <Attribute Name="Direction" AttributeDataType="xs:string" />
        <Attribute Name="Cardinality">
          <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
          <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
        </Attribute>
        <Attribute Name="Category" AttributeDataType="xs:string" />
        <ExternalInterface Name="ConnectionPoint" ID="9942bd9c-c19d-44e4-a197-11
            b9edf264e7" RefBaseClassPath="AutomationMLInterfaceClassLib/
            AutomationMLBaseInterface/PortConnector" />
      </RoleClass>
      <RoleClass Name="Resource" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole" />
      <RoleClass Name="Product" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole" />
      <RoleClass Name="Process" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole" />
      <RoleClass Name="Structure" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole">
        <RoleClass Name="ProductStructure" RefBaseClassPath="AutomationMLBaseRoleClassLib
            /AutomationMLBaseRole/Structure" />
        <RoleClass Name="ProcessStructure" RefBaseClassPath="AutomationMLBaseRoleClassLib
            /AutomationMLBaseRole/Structure" />
        <RoleClass Name="ResourceStructure" RefBaseClassPath="
            AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure" />
      </RoleClass>
      <RoleClass Name="PropertySet" RefBaseClassPath="AutomationMLBaseRoleClassLib/
          AutomationMLBaseRole" />
    </RoleClass>
  </RoleClassLib>
  <RoleClassLib Name="AutomatioMLGraphRoleClassLib">
    <RoleClass Name="Graph" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole" />
    <RoleClass Name="Subgraph" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole/Group" />
    <RoleClass Name="Walk" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole/Group">
      <RoleClass Name="Chain" RefBaseClassPath="AutomatioMLGraphRoleClassLib/Walk" />
      <RoleClass Name="SubgraphWalk" RefBaseClassPath="AutomatioMLGraphRoleClassLib/
          Subgraph" />
    </RoleClass>
    <RoleClass Name="Edge" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole">
```

```
      <ExternalInterface Name="VertexEdgeInterface" RefBaseClassPath="
          AutomationMLGraphInterfaceClassLib/VertexEdgeInterface" ID="b33a8603-0aca
          -4478-8a67-88e1c08ae171" />
      <RoleClass Name="MultiEdge" RefBaseClassPath="AutomatioMLGraphRoleClassLib/Edge">
        <ExternalInterface Name="MultiEdgeConnectionPoint" RefBaseClassPath="
            AutomationMLGraphInterfaceClassLib/MultiEdgeConnectionPoint" ID="63a375c0-
            a6ec-445d-a755-f95fc0c05c35" />
      </RoleClass>
      <RoleClass Name="HyperEdge" RefBaseClassPath="AutomatioMLGraphRoleClassLib/Edge">
        <ExternalInterface Name="VertexEdgeInterface" RefBaseClassPath="
            AutomationMLGraphInterfaceClassLib/VertexEdgeInterface" ID="7e7e841b
            -5328-4094-9ca5-964465d9321f" />
      </RoleClass>
    </RoleClass>
    <RoleClass Name="Vertex" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole">
      <ExternalInterface Name="VertexEdgeInterface" RefBaseClassPath="
          AutomationMLGraphInterfaceClassLib/VertexEdgeInterface" ID="73d5ff56-d635
          -4780-84e3-d19f3683fc23" />
    </RoleClass>
    <RoleClass Name="EdgeSet" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole" />
  </RoleClassLib>
  <RoleClassLib Name="AutomationMLEOLRoleClassLib">
    <Version>1.0.0</Version>
    <RoleClass Name="EOLProcessRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole/Process">
      <Attribute Name="ProcessEconomic" />
      <Attribute Name="ProcessEcological" />
      <Attribute Name="ProcessSocietal" />
      <Attribute Name="ProcessTime" />
      <Attribute Name="ProcessQuality" />
    </RoleClass>
    <RoleClass Name="EOLResourceRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib
        /AutomationMLBaseRole/Resource"></RoleClass>
    <RoleClass Name="EOLVertexRoleClass" RefBaseClassPath="AutomatioMLGraphRoleClassLib/
        Vertex" />
    <RoleClass Name="EOLEdgeRoleClass" RefBaseClassPath="AutomatioMLGraphRoleClassLib/
        Edge" />
    <RoleClass Name="EOLRecoveryGoalRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
      <Attribute Name="SystemFutureEconomic" />
      <Attribute Name="SystemFutureEcological" />
      <Attribute Name="SystemFutureSocietal" />
      <Attribute Name="SystemFutureTime" />
      <Attribute Name="SystemFutureQuality" />
    </RoleClass>
    <RoleClass Name="EOLProcessSetRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure/ProcessStructure" />
    <RoleClass Name="EOLResourceSetRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure/ResourceStructure" />
    <RoleClass Name="EOLStrategyRoleClass" RefBaseClassPath="AutomationMLBaseRoleClassLib
        /AutomationMLBaseRole">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
    </RoleClass>
    <RoleClass Name="EOLPerformanceIndexRoleClass" RefBaseClassPath="
        AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
  </RoleClassLib>
  <RoleClassLib Name="PreviousLifeCyclePhasesRCLib">
    <Version>1.0.0</Version>
    <RoleClass Name="arbitrary" RefBaseClassPath="AutomationMLBaseRoleClassLib/
        AutomationMLBaseRole" />
  </RoleClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLRecoveryGoalSystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="ProductionSystemRecovery">
      <Attribute Name="SystemCurrentEconomic" />
      <Attribute Name="SystemCurrentEcological" />
      <Attribute Name="SystemCurrentSocietal" />
```

```
      <Attribute Name="SystemCurrentQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLRecoveryGoalRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="ComponentRecovery">
      <Attribute Name="SystemCurrentEconomic" />
      <Attribute Name="SystemCurrentEcological" />
      <Attribute Name="SystemCurrentSocietal" />
      <Attribute Name="SystemCurrentQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLRecoveryGoalRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="MaterialRecovery">
      <Attribute Name="SystemCurrentEconomic" />
      <Attribute Name="SystemCurrentEcological" />
      <Attribute Name="SystemCurrentSocietal" />
      <Attribute Name="SystemCurrentQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLRecoveryGoalRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLProcessSystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="Disassembly">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Reconditioning">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Preprocessing">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
```

```
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Storage">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Reassembly">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Cleaning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
        EOLProcessRoleClass" />
  </SystemUnitClass>
  <SystemUnitClass Name="Decommissioning">
    <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
      <Value>5</Value>
    </Attribute>
    <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
      <Value>5</Value>
```

```xml
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Testing">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Commissioning">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Transportation">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Disposal">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
```

```
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Repair">
      <Attribute Name="ProcessEconomic" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessEcological" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessSocietal" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessTime" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <Attribute Name="ProcessQuality" AttributeDataType="xs:decimal">
        <Value>5</Value>
      </Attribute>
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLProcessRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLPerformanceIndicesSystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="EconomicImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="EcologicalImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="SocietalImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="TimeImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="QualityImpact">
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLPerformanceIndexRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="AutomationMLEOLStrategySystemUnitClassLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="DirectReuse">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="ReuseAfterRepair">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
```

```
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="ReuseAfterRemanufacturing">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
    <SystemUnitClass Name="Recycling">
      <Attribute Name="EOLStrategyEconomic" />
      <Attribute Name="EOLStrategyEcological" />
      <Attribute Name="EOLStrategySocietal" />
      <Attribute Name="EOLStrategyTime" />
      <Attribute Name="EOLStrategyQuality" />
      <SupportedRoleClass RefRoleClassPath="AutomationMLEOLRoleClassLib/
          EOLStrategyRoleClass" />
    </SystemUnitClass>
  </SystemUnitClassLib>
  <SystemUnitClassLib Name="PreviousLifeCyclePhasesSUCLib">
    <Version>1.0.0</Version>
    <SystemUnitClass Name="arbitrary" />
  </SystemUnitClassLib>
</CAEXFile>
```