Technical Report

Introducing Engineering Data Logistics for Production Systems Engineering

Stefan Biffl[1]
Arndt Lüder[2]
Felix Rinker[1,3]
Laura Waltersdorfer[1,3]
Dietmar Winkler[1,3]

[1]TU Wien, Information Systems Engineering, Information and Software Engineering, Quality Software Engineering, Vienna, Austria
<firstname>.<lastname>@tuwien.ac.at

[2]Otto-v.Guericke Universität Magdeburg, Magdeburg, Germany
<firstname>.<lastname>@ovgu.de

[3] Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Information and Systems Engineering, TU Wien, Vienna, Austria
<firstname>.<lastname>@tuwien.ac.at

Technical Report No. CDL-SQI 2018-10
Issued: October 2018

Citation: S. Biffl., A. Lüder, F. Rinker, L. Waltersdorfer. D. Winkler: „Introducing Engineering Data Logistics for Production Systems Engineering“,Technical Report CDL-SQI 2018-10, TU Wien, Vienna, Austria, October, 2018.

Introducing Engineering Data Logistics for Production Systems Engineering

Stefan Biffl[1], Arndt Lüder[2], Felix Rinker[3], Laura Waltersdorfer[1], Dietmar Winkler[3]

Abstract. In the parallel engineering of large and long-running automation systems, such as *Production Systems Engineering* (PSE) projects, engineering teams with different backgrounds work in a so-called *Round-Trip Engineering* (RTE) process to iteratively enrich and refine their engineering artifacts and need to exchange data efficiently to prevent the divergence of local engineering models. Unfortunately, the heterogeneity of local engineering artifacts and data coming from several engineering disciplines makes it hard to integrate the discipline-specific views on the data for efficient synchronization.

In this chapter, we introduce use cases to illustrate RTE requirements for an *Engineering Data Logistics* (EDaL) process and information system that enable the efficient integration and systematic exchange of engineering data in a PSE project. We propose the concept of an EDaL process that analyzes *Engineering Data Exchange* (EDEx) flows from data providers to a consumer. We introduce requirements and steps for an EDEx process that guides the definition and semantic mapping of engineering data elements for exchange. We discuss main requirements for and design elements of an EDaL information system for automating EDaL process capabilities. We evaluate the effectiveness and effort of the EDEx process and concepts in a feasibility case study with requirements and data from real-world use cases at a large PSE company in comparison to a traditional manual point-to-point engineering data exchange. Results from the feasibility study indicate that the EDEx process may be more effective than the traditional point-to-point engineering artifact exchange and a good foundation for EDaL in an engineering project.

Keywords: multidisciplinary engineering, production systems engineering, cyber-physical production systems, engineering process, process design, data exchange, data integration.

---

[1] Technische Universität Wien, Institute of Information Systems Engineering, Vienna, Austria, <firstname>.<lastname>@tuwien.ac.at

[2] Otto-v.-Guericke University / IAF, Magdeburg, Germany, arndt.lueder@ovgu.de

[3] Christian Doppler Research Laboratory Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), <firstname>.<lastname>@tuwien.ac.at

7r.1 Introduction

Engineering industrial, recently also cyber-physical, production systems, e.g., long-running and safety-critical systems for assembling automotive parts or for producing metal, is the business of multi-disciplinary *production system engineering* (PSE) companies (Biffl *et al*., 2017; Vogel-Heuser *et al*., 2017). To shorten the PSE project duration, the engineering disciplines in the PSE process often work in parallel, in a so-called *Round-Trip Engineering* (RTE) process to iteratively enrich and refine their engineering artifacts.
In parallel engineering, the disciplines, such as mechanical, electrical, and simulation engineering, develop their engineering and artifacts, such as plans, models, software code, or machine configurations, independently, but have to consider dependencies between the engineering disciplines in order to build a common system. A key success factor is the capability for *Engineering Data Logistics* (EDaL) to enrich and improve engineering data iteratively along the engineering process and with information coming back from testing, simulation, and operation, following the so-called *round-trip engineering* (RTE) process pattern (Biffl *et al*., 2018b). EDaL depends on the capability to exchange selected data in the engineering artifacts with related domain experts efficiently and in a timely manner in order to reduce rework due to inconsistencies in diverging local data views. EDaL is the foundation for agile PSE that can adapt changes communicated from backflows in the PSE process due to design changes or errors. Unfortunately, the heterogeneity of local engineering artifacts and data coming from several engineering disciplines complicate the integration of discipline-specific views on the data for efficient synchronization and lead to late communication of changes and ineffective version management of engineering data.
The RTE process provides the foundation for consistent distributed data management and requires (a) a process for negotiating data elements requested by data consumers and matching to data elements coming from data providers and (b) an efficient engineering data exchange method and mechanism for conducting the agreed data exchanges between data sources and sinks of domain experts.

We illustrate the EDaL and *Engineering Data Exchange* (EDEx) processes with use cases from simulation in PSE, as simulation is a major consumer of engineering data for assessing the safety and business risks of a production system before system construction. Goal of the simulation engineer is to design simulation systems that allow exploring dynamic properties of the designed production system, such as throughput or the physical feasibility of production steps. Therefore, the simulation engineer requires input data from several engineering data providers on key parameters of system parts, such as the rotation speed, torque, control signals, or power consumption of a motor as foundation for calculating and analyzing the movement of work pieces and robots over time. In addition, the simulation engineer provides feedback on issues with the engineering data and on design issues that need to be addressed by the domain experts, e.g., changing the placement of a robot to improve the material throughput in a work cell.

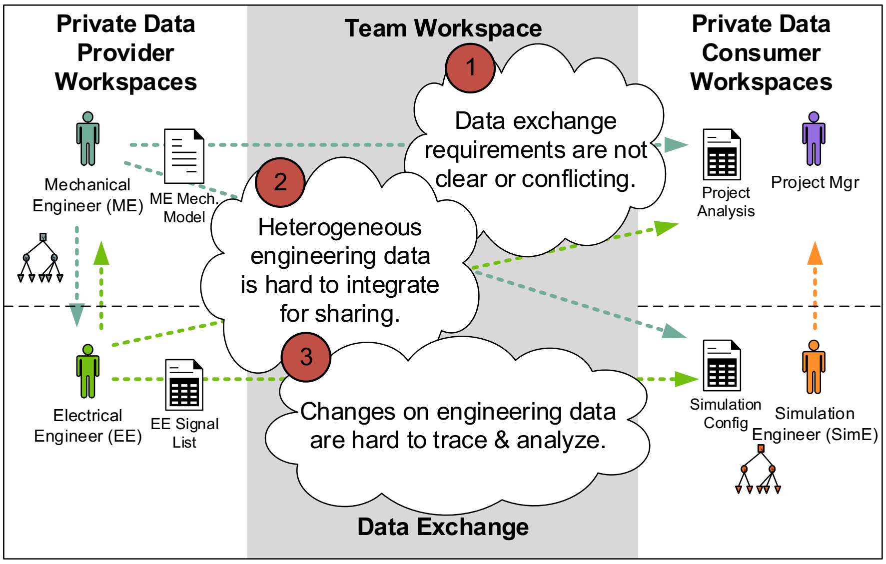


Figure 7r.1: Challenges in data exchange in parallel systems engineering.

In the *traditional EDEx process* (Biffl *et al.*, 2014a; Biffl *et al.*, 2018a), domain experts communicate their engineering artifacts point-to-point (P2P), typically in the form of spreadsheet tables, pdf or XML files. Unfortunately, in the traditional EDEx process, Lüder *et al.* (2018) identified the following major challenges that also make EdaL, which builds on EDEx, more difficult (see Figure 7r.1).

C1. Data exchange requirements are not clear or conflicting. While the domain experts know their partners in the engineering process, there is surprisingly little concern for the data exchange requirements of data consumers and the impact of ineffective or inefficient EDaL on the project team performance. As EDaL is not a formal engineering activity but a necessary cost factor, comparable to the transport activity between production tasks, every engineer tries to minimize locally, overall at the expense of the cost to the engineering team. In many cases, data consumers bear the cost and risk of EDaL due to missing awareness for and support by an EDaL process and infrastructure. For potential data providers, it is often not clearly defined which project participants require what kind of data at what point in time in the project. Even if general dependencies between stakeholders are known, the specific relations between engineering artifacts and their content within an engineering project can change during the project execution. Insufficient overview and conflicting interests may prevent even willing stakeholders from sharing their data.

C2. Heterogeneous engineering data is hard to integrate for sharing. Due to strongly diverging scientific and practical histories, engineering tools and data are typically specific for a discipline and not designed for the use with other disciplines or with the project they contribute to. While the disciplines share some common concepts (Sabou *et al*., 2017; Winkler et al, 2017), such as the concept of a machine, a device, or a signal, these concepts are not consistently modeled, making data integration for sharing error prone and hard to automate. Consequently, data providers tend to share engineering artifacts that take high effort for consuming domain experts to find and interpret, and, thus, hinder comprehensive automated processing of the engineering data buried within the engineering artifacts. While the reusable representation of explicit semantic relationships between similar concepts of data providers and consumers may be costly for an informal P2P data exchange, EDaL support for the EDEx in an engineering team can build on the explicit representation of common concepts in as semantic links between heterogeneous engineering data sets to enable automation of EDEx and analyses.

C3. RTE changes on engineering data are hard to trace and analyze. A data consumer in the RTE process has to be keep track on the changes in the data versions s/he receives to enable analyses of the received data and meta data, e.g., for identifying missing or inconsistent data. Unfortunately, using point-to-point (P2P) data exchange makes it very hard for a consumer to trace and analyze the set of data versions exchanged that may come from several providers as there is no EDaL support to keep track of EDEx flows, including roles and rules for process conduct.

In this chapter, we introduce a process for efficient *Engineering Data Logistics* (EDaL) to address these challenges and to automate data logistics in order to improve the value and reduce the risks of EDEx. We investigate the following research questions (RQs) based on *Design Science* research methodology (Wieringa, 2014) and the use cases in (Biffl *et al*., 2018a; Biffl *et al*., 2018b).

RQ1. What are main elements of an Engineering Data Logistics (EDaL) approach in round-trip Engineering? To address this research question, we define the term *Engineering Data Logistics* (EDaL) and analyze key requirements for effective and efficient EDaL, such as support for clarifying data consumer and provider win conditions that may conflict or patterns for EDaL for the enrichment and backflow of engineering data in a *round-trip engineering* process. A key EDaL capability is the effective organization of exchanging engineering data. Therefore, we derive requirements for defining and negotiating the required individual data flows between data providers and consumers.

RQ2. What are main elements of an effective and efficient engineering data exchange (EDEx) process in Multi-Disciplinary System Engineering? To address this research question, we discuss a consumer-driven process for defining, prioritizing, and designing EDEx data flows in a project team. A key capability of EDEx is to support the data integration of heterogeneous engineering data by representing the implicit relationships between engineering data coming from different domains as foundation for a common view on and efficient sharing of data.

As the manual conduct of EDEx is inefficient, we derive requirements for an EDEx information system that automates functions in EDEx process steps.

RQ3. What are main information system mechanisms that enable engineering data logistics for Multi-Disciplinary System Engineering? To address this research question, we discuss the design of an EDaL information system (EDaLIS) that supports efficient tracing of data flows in an engineering team as foundation for analyzing the exchanged data and the EDEx process.

From the research we expect the following contributions for the *information systems engineering* (ISE) community. The use cases and EDaL process give ISE researchers insight into the PSE domain, the foundation for Industry 4.0 applications. The EDEx process contributes capabilities for designing and investigating agile processes and information systems in PSE, a foundation for conducting engineering projects for cyber-physical production systems economically.

The remainder of this chapter is structured as follows. Section 2 introduces use cases, collected in workshops with stakeholders at a large PSE company, to illustrate RTE requirements for an *Engineering Data Logistics* (EDaL) process and information system that enable the efficient integration and systematic exchange of engineering data in a PSE project. Section 3 summarizes related work on approaches for data logistics in multi-disciplinary *production systems engineering* (PSE), information systems and software engineering. Section 4 motivates the research questions and the research approach. Section 5 introduces steps for an *EDaL process* to address the requirements identified in Section 2. Section 6 discusses main design elements for effective and efficient *EDaL information system* (EDaLIS) mechanisms to address the requirements identified in Section 2. Section 7 reports on an evaluation of the effectiveness and effort of the proposed EDaL process with EDaLIS mechanisms in a feasibility case study with requirements and data from real-world use cases with domain experts at a large PSE company. Section 8 discusses the research findings and limitations. Section 9 concludes and proposes future research work.

7r.2 Engineering Data Logistics Use Cases

Section 2 introduces use cases, collected in workshops with stakeholders at a large PSE company (Biffl *et al*., 2018s), to illustrate RTE requirements for an *Engineering Data Logistics* (EDaL) process and information system that enable the efficient integration and systematic exchange of engineering data in a PSE project. This section introduces a use case with illustrative data for the data exchange of the data consumer simulation with several data providers.

7r.2.1 Engineering Data Logistics Use Cases

For *Engineering Data Logistics* (EDaL), we consider the following use cases, starting from the traditional basic collection/provision of engineering artifacts in a point-to-point (P2P) network of domain experts (UC1), progressing to stepwise enrichment of engineering artifacts (U2), to the parallel iterative enrichment of engineering artifacts (U3), and finally to consider backflows in the engineering network (U4) as foundation for true round-trip engineering.

In the use cases, we assume a team of domain experts involved in designing a work cell as part of a larger production system.

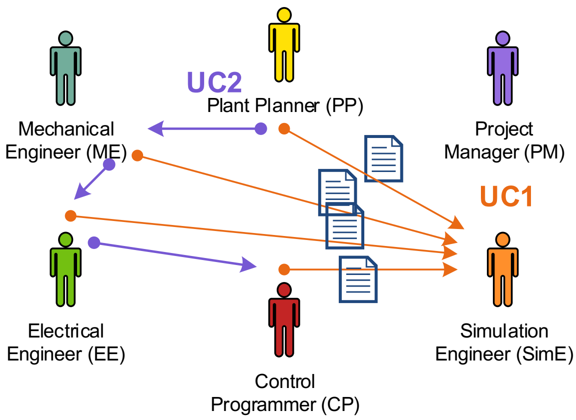


Figure 7r.2a: ***Engineering Data Logistics*** **(EDaL) Use Cases in the** ***round-trip engineering*** **(RTE) process.**

UC1 Artifact provision. Figure 7r.2a shows a set of domain experts in an engineering project, the plant planner (PP), the machine engineer (ME), the electrical engineer (EE), and the control programmer (CP) as providers of engineering artifacts, and the simulation engineer (SimE) and the project manager (PM) as consumers of engineering data. The orange arrows in Figure 7r.2a illustrate the provision of engineering artifacts by the PP, ME, EE, and CP to the SimE, who has to extract the data from the engineering artifacts, to integrate the data from heterogeneous sources, and to clarify issues with each data provider. We describe the artifact provision as data exchange in the notation (PP, ME, EE, CP) -> SimE, i.e., the SimE requires a data set from the other four domain experts.

UC2 Sequential enrichment of artifacts. In a typical sequential engineering process, the PP starts with providing the structure of the production system, then the ME selects and designs the mechanical parts, then the EE designs the electrical mechanisms for providing the system with energy and information connections, then the control programmer designs the software and configurations to automate system parts. The violet arrows in Figure 7r.2a illustrate the same sequence of engineering artifact exchanges between the domain experts.

UC2a. Simple sequential enrichment. In the simplest case, the domain experts conduct one sequence of engineering artifact exchanges for enrichment and then deliver to the SimE: PP -> ME, ME -> EE, EE -> CP; (PP, ME, EE, CP) -> SimE.

UC2b Sequential enrichment with updates. In an advanced case, each domain expert may improve her engineering artifacts and propagate the updated engineering artifact version along the engineering chain, resulting in a sequence of follow-up updates. In this case, the domain engineers require a mechanism to efficiently identify changes between artifact versions. PP (PP.A) -> ME, PP (PP.A') -> ME denotes that the PP sends the ME first his artifact A and then an updated version A'.

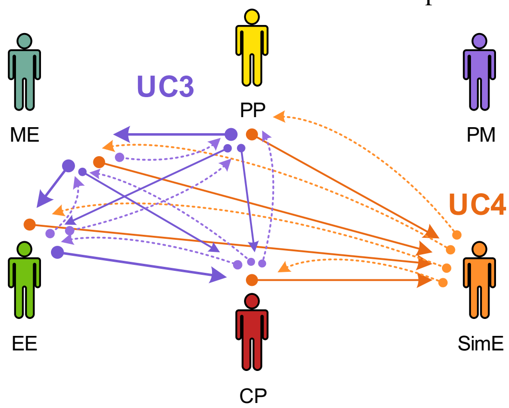


Figure 7r.2b: Engineering Data Logistics (EDaL) Use Cases in the round-trip engineering (RTE) process.

UC3 Parallel enrichment of engineering artifacts. In a parallel engineering process, the domain experts start in parallel with a rough design the refine their designs in parallel and exchange updates as needed. The violet arrows in Figure 7r.2b illustrate the multitude of point-to-point exchanges in parallel engineering making it hard to keep an overview on the artifact versions and their dependencies, as the sequence of updates and their time of communication in the team is not known. SimE -> (PP, ME, EE) denotes a backflow from simulation to earlier engineering activities.

UC4 Backflows of artifacts. Changes to engineering artifacts may come from backflows in the engineering process due to changed requirements, errors found in the engineering design, or feedback from tests, simulations, and operation. The dashed arrows in Figure 7r.2b illustrate the multitude of potential backflows in the engineering team. Unfortunately, there is, in general, no effective and efficient process for systematic backflows making it uncertain to what extent backflow information is considered or lost.

The data exchange requirements, specified by these use cases in the data logistics network, result in a set of engineering data exchange (EDEx) flows, e.g., PP -> ME. In the following, we focus on an individual EDEx flow to identify EDEx requirements and solution options. The *round-trip engineering* (RTE) process (Biffl *et al.*, 2018b) provides the foundation for consistent distributed data management and requires (a) a process for negotiating data elements requested by data consumers and matching to data elements coming from data providers and (b) a data exchange method and mechanism for executing the agreed data exchanges between domain experts and their data sources and sinks.

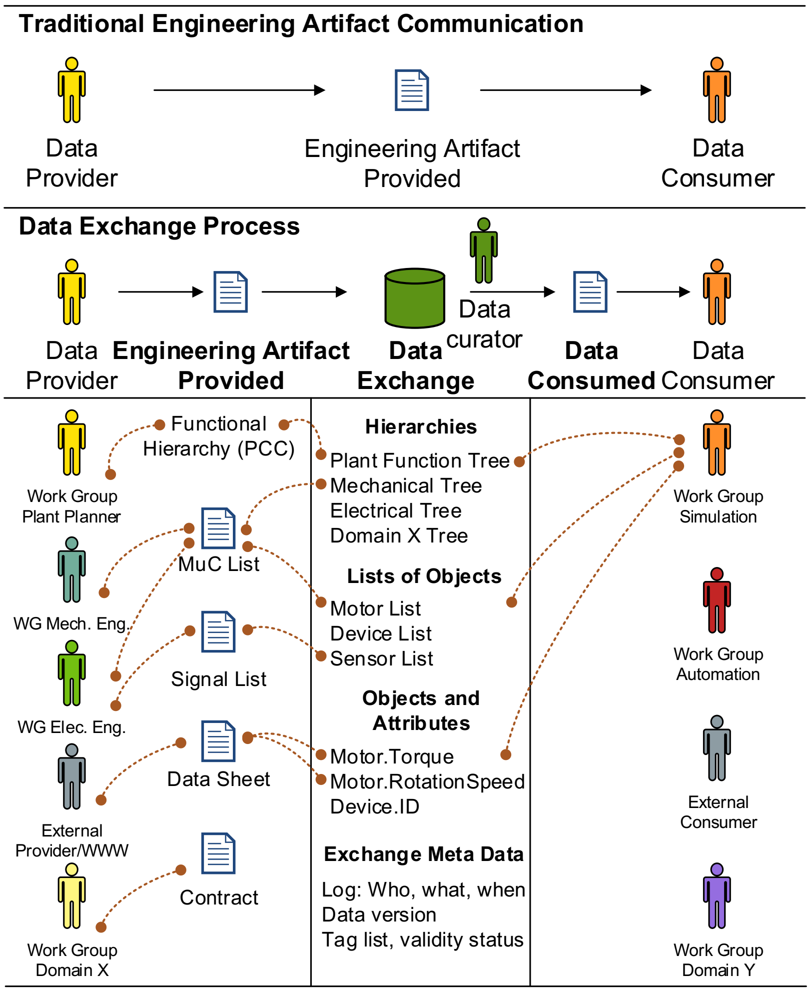


Figure 7r.3: Simple data processing map illustrating one data flow from several data providers to one simulation expert.

Figure 7r.3 illustrates the EDEx data flow for UC1 *Artifact provision*. The data flows from a data provider to a data consumer. In the traditional engineering artifact communication, the provider sends an artifact that contains the data relevant for the customer. However, the consumer has to bear the effort for extracting and validating the data from the artifact. In addition, there is no systematic traceability and quality assurance in the process. Therefore, we propose introducing an *engineering data* *logistics information system* (EDaLIS) as foundation for traceable, quality assured, and efficient data flows in an engineering team. In the following, we introduce use cases for individual EDEx data flows.

7r.2.2 Engineering Data Exchange Use Cases for Evaluation

Based on observations and discussions with our industry partners, we identified two illustrative use cases (UCs) that show the benefit of improved engineering EDEx: engineering data collection for *production system simulation* and for *production system engineering project monitoring*. The engineering of a typical industrial *production system* (PS), such as automotive assembly, requires at least the collaboration of - and EDEx - between the *plant planner* (PP), who plans the layout of the PS, *mechanical engineer* (ME), *electrical engineer* (EE), and *control programmer (CP)*. Each domain expert designs and updates complex and heterogeneous local models that are hard to understand by other domain experts.

UC Sim. Data exchange for production system simulation. In a typical advanced engineering environment, a *simulation engineer* (*SimE*) designs and runs simulation models to check the engineering results and to optimize production system parameters, such as safety risks, production throughput, and energy consumption. These design of the simulation models depend on the input of several other domain experts, such as the configuration parameters of motors and conveyers in a transport system and requirements of production processes, such as process duration (s) and production resources, such as length (m), size (m^2 or m^3), mass (t), heat radiation (kW), power consumption (kW), or maximal noise level (db).

The *SimE* requires this input from data providers to calculate characteristics, e.g., power consumption or movement dynamics, of a system part, e.g., a drive chain, to find out whether the system part will behave as intended and to provide feedback to the contributing engineering disciplines on risks and on necessary design changes.

If the simulation identifies infeasible system plans or significant risks, the engineers have to cooperate to adapt the plans in the individual disciplines.

A single change in a discipline may trigger a chain of adaptations in other disciplines and lead to unclear implications on the overall system and avoidable rework in later project phases. Therefore, project stakeholders would like to evaluate defined constraints as early as possible for each relevant change of a local model.

The manual synchronization of these data typically requires additional effort, tends to be error prone, and induces avoidable project risks.

UC PM. Production system engineering project monitoring. The project manager (PM) wants to use the input from data providers to the simulation engineer to assess project progress by analyzing the completeness and quality of data with respect to the project phase and planned deliverables. Missing or inconsistent data may be fine in an early design phase, but may pose a major risk in closer to a later design milestone and require action by the PM.

7r.3 Related Work

7r.3.1 Data Logistics in Multi-Disciplinary Systems Engineering

Analogue to the real world, *Engineering Data Logistics* describes the flow of data elements from a data provider to a data consumer according to the customer's requirements. As in traditional logistics, the change of, even single, parts of the data transport and exchange network may affect characteristics of the data logistics system, such as duration, quality, or cost of logistics, important aspects for the cost and risk of the engineering system using the data logistics as well as the business advantages of enabling frequent and cheap data updates between work groups that work in parallel (Hell, 2018) (Andersen *et al*., 2018).

In the *Production Systems Engineering* (PSE) process (Biffl *et al*., 2017; Biffl et al, 2018b), the content of the exchanged artifacts is important as these artifacts contain only part of the local models of the domain experts. Due to the inherent dependencies between these local models, such as dependencies between mechanical engineering defining cable routes, electrical engineering defining the applied wires and their location on cable routes and communication system engineering defining used communication lines all effecting the possible impact of electrical fields on communication system quality, domain knowledge is required on both the customer and the provider data models to interpret the content of the exchanged data. Therefore, it is necessary to move from delivering engineering artifacts to *engineering data exchange* (EDEx). Although *business process analysis* (OMG, 2011) is useful to better understand the relevant stakeholder groups, activities, and exchanged engineering artifacts, additional data modeling is required to represent the knowledge required for EDEx. Thus, the workflow analysis shall cover the aspects engineering decisions (engineering activities made), applied engineering tools, created and required artifacts covering engineering information, and involved humans with skills and competences (Schäffler et. al, 2013)

While EDEx is already important and difficult for traditional PSE, the migration towards cyber-physical systems is a complex task that requires an extensive solution, covering technical, operational, and human dimensions (Calà, *et al*., 2017). Due to this multi-dimensional complexity, traditional information systems have not yet adequately addressed the challenges imposed by collaboration in multi-disciplinary engineering systems: heterogeneous tools and data formats, diverging views on artifacts and their versioning are the most pressing ones (Drath *et al*., 2011). Optimizing and enriching the currently available engineering data and data exchange is a possible quick win that can be achieved by integrating EDEx (Sabou *et al*., 2017) based on the machine understandable representation of knowledge on how exchanged data elements fit to the local data models of the data providers and consumers.

While there are engineering tool suites that integrate several engineering functions in one set of tools with a common data model that greatly simplifies EDEx, most engineering projects use many tools with heterogeneous data models that are challenging to integrate (Biffl *et al*., 2017). The traditional EDEx process (Biffl *et al*., 2014a) is a point-to-point exchange of engineering artifacts between domain experts via e-mail, repository, or USB stick, typically in the form of spreadsheet tables, pdf or XML files.

Lüder *et al*. (2018) introduce an architecture for engineering data logistics, based on *AutomationML* (IEC 62714, 2014-2018; Vogel-Heuser *et al*., 2017), an open, XML-based format for the exchange of engineering data. The proposed architecture allows exchanging data between discipline-specific data models with varying hierarchical key systems. While this approach is useful in an *AutomationML* environment, the approach does not consider how to negotiate the EDEx between many data consumers and providers. Often the data providers tend to provide all kinds of data that someone might find useful in the future, leading to a pile of data that is expensive to provide and hardly used.

7r.3.2 Multi-Model Dashboard

This subsection introduces the *Multi-Model Dashboard* (MMD) approach (Biffl *et al*., 2014a; Biffl *et al*., 2014b) and points out gaps in research. The MMD approach extends the Decision Board approach (Holl *et al*., 2012) by adding the concept of constraints, formally defined using shared model parameters, and by automating the data extraction of parameter values from heterogeneous data sources with semantic data integration (Biffl *et al*., 2014). The tool-supported MMD process guides the systematic definition, design, monitoring, and evaluation of MMD parameters and constraints, visualized on the MMD. A dashboard provides the semantically integrated values of parameters and of constraints to the domain experts, as parameter values in various local models change during the project. The MMD provides promising capabilities for data extraction from engineering artifacts, often engineering models.

The MMD concepts of private workspaces and common team workspace in a heterogeneous System-of-System environment fit well to typical parallel systems engineering environments. The roles in the MMD approach, *data subscriber* and *publisher* can be mapped well to the *data consumer* and *provider* in the context of this paper. While we can build on the MMD strengths as foundation for the EDEx research in this chapter, the following limitations of the MMD approach require significant adaptation for data exchange in a system engineering project. The MMD does not consider the provision of data to consumers but focuses on the evaluation of engineering parameters and constraints. In practice, the MMD assumption of well-defined common concepts may be difficult as several disciplines may cooperate without one discipline clearly leading. The MMD Dashboard software architecture based on an *AML Hub*[4] is a limitation for a more general EDEx software architecture. The research questions in (Biffl *et al*., 2014a) focused on identifying

[4] http://www.amlhub.at/

common concepts in a heterogeneous System-of-System environment, software design options for change monitoring and for awareness design in a heterogeneous System-of-System environment, while the focus of this chapter is engineering EDaL based on EDEx definition and operation.

7r.3.3 Data Exchange in Information Systems & Software Engineering

Methods from business process management provide useful approaches, such as UML class diagrams (Brambilla *et al*., 2012) or BPMN (OMG, 2011), for EDEx definition by characterizing involved stakeholders, systems and, to some extent, data types and their relationships. Workflow management systems allow the automated set-up, performance and monitoring of previously defined processes, a common tool for industry use-cases is Aris Toolset.[5] However, these methods are generic and need to be adapted for new contexts, also in the case of heterogeneous engineering data integration (Rosemann and vom Brocke, 2015). In specialized domains, such as medicine, science and engineering, new approaches may be needed to optimize data exchange according to domain-specific requirements (Jimenez-Ramirez *et al*., 2018; Putze *et al*., 2018).

Semantic Web technologies are recognized for facilitating data exchange across applications and organizations in the web and have proposed engineering data integration approaches following *the interchange standardization approach* (Sabou *et al*. 2017). However, the manifold types of dependencies in PSE data models are different form typical Semantic Web requirements (Kovalenko and Euzenat, 2016) and the Semantic Web technology stack is therefore currently seldom used in engineering environments.

Model-driven software engineering (Brambilla *et al*., 2012) is a well-established software methodology, in which the abstraction of the problem domain is utilized to facilitate automated code generation, testing, and verification. Seamless Model-Based Development (Broy *et al*., 2018) is a desirable strategic goal that is hard to achieve in the current heterogeneous engineering reality with less-than-willing tool vendors, who prefer vendor lock in to open standards. However, as domain-specific languages, such as AutomationML, gain acceptance in PSE, a foundation for model-based approaches is likely to become stronger in the next few years.

Software engineering design patterns (Hohpe and Woolf, 2003) encapsulate best practices of software system design for commonly occurring problems, in our case data and tool integration. In the context of this work, we build on design patterns such as *message passing* and *publish-subscribe* to support the loose coupling of engineering work groups and tools.

7r.3.4 Technical Data Exchange Formats

To facilitate data exchange, technical data exchange formats have to be able to cover, possibly all but at least most of, the information required and/or produced

[5] https://www.softwareag.com/ch/products/aris_alfabet/bpa/default.html

within the PSE process by *data consumers* and *providers*. For these data exchange formats there are a set of (sometimes contradicting) requirements to be fulfilled (Lüder and Schmidt, 2017):

- The data format shall be adaptable to different application cases and flexible with respect to extensions and changes.
- The data representation shall be efficient.
- The data representation shall be human readable.
- The data representation shall be based on international standards.

These requirements lead to an XML based data format, which makes engineering tools standardized data exchange formats like STEP (Xu, 2012) and AutomationML (Drath, 2009) preferable as they represent a tree structure similar to the topologies common in engineering, such as functional, mechanical, or electrical hierarchies.

Following (Diedrich et al., 2011), the data exchange between engineering tools requires two levels of standardization, the syntax level and the semantic level. The syntax level defines the correct technical representation of the data objects in the data exchange format, including the vocabulary of the data exchange. In contrast, the semantic level defines the interpretation of data objects, i.e., the conceptual meaning of objects in the engineering tool chain. With respect to the intended EDEx approach, both levels are relevant, but the semantic level is more import as it enables the identification of common information exchanged between the *data provider* and *consumer*.

Technical data exchange formats can be defined in two ways, either they define syntax and semantics together, as in the STEP approach or the approach defined in VDI Guideline 3690 (VDI 3690, 2012-2017), or they define syntax and semantics separately, as in the AutomationML or the XMI approach (Grose, et. al, 2002). Since the separate definition of semantics enables better flexibility and adaptability of a data exchange format to application cases, this approach seems to be preferable.

7r.4 Research Questions and Approach

This section motivates the research questions and the research approach. In this chapter, we introduce a process for efficient Engineering Data Logistics (EDaL) to address these challenges and to automate data logistics in order to improve the value and reduce the risks of EDEx. We investigate the following research questions (RQs) based on *Design Science* research methodology (Wieringa, 2014).

RQ1. What are main elements of an Engineering Data Logistics (EDaL) approach in round-trip System Engineering? To address this research question, we define in Section 7r.5.1 the term *Engineering Data Logistics* (EDaL) and analyze key requirements for effective and efficient EDaL, such as support for clarifying data consumer and provider win conditions that may conflict or patterns for EDaL for the enrichment and backflow of engineering data in a *round-trip engineering* process. Section 7r.5.2 discusses EDaL design considerations to address the EDaL requirements. A key EDaL capability is the effective organization of exchanging engineering data. Therefore, we derive in Section 7r.5.3 requirements for defining and negotiating the required individual data flows between data providers and consumers.

RQ2. What are main elements of an effective and efficient engineering data exchange (EDEx) process in Multi-Disciplinary System Engineering? To address this research question, Section 7r.5.3 discusses requirements for the EDEx process collected in workshops with stakeholders at a large PSE company, and proposes steps for a consumer-driven *EDEx process* that address these requirements by defining, prioritizing, and designing EDEx data flows in a project team. A key capability of EDEx is to support the data integration of heterogeneous engineering data by representing the implicit relationships between engineering data coming from different domains as foundation for a common view on and efficient sharing of data.

For designing the EDEx process, we adapt in Section 7r.5.4 the *Multi-Model Dashboard* approach (Biffl *et al*., 2014a) from constraint evaluation to EDEx and replace the design requirement of an initial common concept model, which may not be available, with direct links between consumer and provider data elements. As the manual conduct of EDEx is inefficient, we derive in Section 7r.5.6 requirements for an EDEx information system that automates functions in EDEx process steps.

RQ3. What are main information system mechanisms that enable engineering data logistics for Multi-Disciplinary System Engineering? To address this research question, Section 7r.5.6 derives requirements for effective and efficient *EDaL information system* (EDaLIS) mechanisms: capabilities for data set specification and for the representation of dependency relationships as foundation for data integration and transformation. We discuss in Section 7r.6 the design of an EDaLIS that supports efficient tracing of data flows in an engineering team as foundation for analyzing the exchanged data and the EDEx process. Section 7r.7 reports on an evaluation of the effectiveness and effort of the proposed EDEx process with EDaLIS mechanisms in a feasibility case study with requirements and data from real-world use cases with domain experts at a large PSE company.

7r.5 Engineering Data Logistics Process

We define term *Engineering Data Logistics* (EDaL) as the exchange of engineering data between data providers and data consumers in an engineering process, following a sequence of exchange steps according to a pattern, such as R*ound-Trip Engineering* (RTE). Following the *design science cycle* in (Wieringa, 2014), we set up an initial problem investigation with workshops (Biffl *et al*., 2018a), outlining the context and problem space of research, and deriving the following requirements for EDEx capabilities that allow addressing the challenges introduced in Section 1: *C1. Data exchange requirements are not clear or conflicting* and *Ch2. Heterogeneous engineering data is hard to integrate for sharing*.

Section 7r.5 derives requirements for EDaL from the use cases described in Section 7r.2 and introduces steps for an *EDaL process* to address these requirements by the EDEx process for defining single data flows in an engineering team. We propose the concept of an EDaL process that can handle different data formats and discipline-specific views and derive requirements for EDEx capabilities, such as guidance for the definition and semantic mapping of engineering data elements for exchange.

7r.5.1 Requirements for an Engineering Data Logistics

From a workshop with domain experts and subsequent discussion of use cases, we derived the following requirements for an EDaL process.

Capa EDaL1. EDaL scope analysis. The EDaL approach should guide the collection and analysis of data consumers, providers, the engineering artifacts and data they want to exchange in the scope of an EDaL use case as foundation for clarifying win conditions for both data consumer and provider, such as compensation for extra effort coming from conducting EDaL tasks.

Capa EDaL2. EDaL use case analysis. The EDaL approach should guide the design of an EDaL by identifying and configuring EDaL patterns to derive a sequence of individual EDEx data flows specified in an EDaL language. EDaL designs should allow addressing the use cases described in Section 7r.2 on (UC1) engineering data provision, (UC2) sequential enrichment of engineering data, (UC3) parallel enrichment of engineering data, and (UC4) backflows of engineering data.

Capa EDaL3. EDEx specification. The EDaL approach should guide the design of an individual EDEx data flow in an EDEx language.

7r.5.2 Engineering Data Logistics Design

To address the EDaL requirements in Section 7r.5.1, we derived the following EDaL process for designing an EDaL solution.

EDaL Step 1. EDaL requirements analysis. The role *EDaL Data Curator* conducts an analysis with candidate data consumers and providers on their data exchange requirements, the engineering artifacts and data they want to exchange. Result is an EDaL requirements document and a data processing map. This data processing map shows a network of data providers and consumers represented as nodes in the network and a set of data flows between a provider and consumer depicted as arrows (see Figure 7r.4).

EDaL Step 2. EDaL use case design. The *EDaL Data Curator* designs with candidate data consumers and providers use cases that address their requirements using EDaL design patterns, such as the RTE pattern and the engineering backflow pattern. Result is a use case description including an initial set of EDEx data flows, e.g., data provider (artifact: data set) -> data consumer. Figure 7r.4 illustrates a solution design based on a central EDaLIS that mediates the data flow between providers and consumers. Providers send their engineering artifacts into the EDaLIS that extracts the data relevant for consumers for distribution to the consumers.

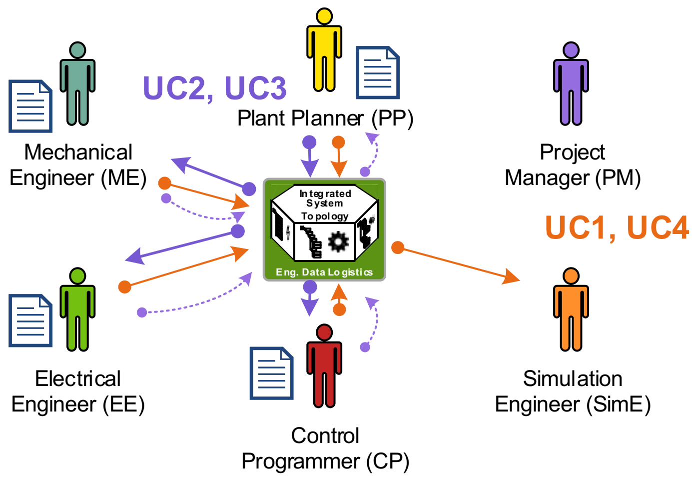


Figure 7r.4: ***Engineering Data Logistics*** **(EDaL) process concepts to address** ***round-trip engineering*** **(RTE) requirements.**

EDaL Step 3. List of EDEx flows. The *EDaL Data Curator* derives a list of EDEx specification candidates from EDaL patterns. Result is a refined set of EDEx data flow specifications, e.g., data provider (artifact: data set) -> data consumer, with a detailed description of the data set as a set of data elements specified in a domain-specific language, such as AutomationML.

7r.5.3 Requirements for an Engineering Data Exchange Process

From a workshop with domain experts and subsequent discussion of use cases, we derived the following requirements for an EDEx process.

Capa EDEx1. Engineering Data Representation. The EDEx approach should allow representing typical engineering data structures, such as tree hierarchies of the functions of a production system, (e.g., a work cell consists of devices), lists of objects (e.g., list of motors), and objects and their attributes (e.g., motor torque or rotation speed) and relationships forming networks (e.g., a work cell with an electric motor requires an electric power supply), both for data consumers and data providers. In addition, the technical data representation needs to be considered by identifying data storing and data exchange technologies that can be applied on the data consumer and data provider sides for encoding the engineering information to be exchanged.

Capa EDEx2. Semantic Link Knowledge representation. This capability concerns the representation of candidates for, overview on, and specifics for semantic links for, data integration between selected consumer and provider data elements. The explicit representation of tacit knowledge on these semantic links will allow reasoning on, improving, and automating data integration and data transformation for EDEx.

Capa EDEx3. Process Data Representation. This capability concerns the representation of metadata on the EDEx process, e.g., who provided what data when, versions of data elements, data quality and validity (e.g., unclear/checked valid, invalid data).

Capa EDEx4. Consumer- and Benefit-Driven EDEx Planning. The EDEx approach should be consumer-driven (with EDEx curator) and consider the likely cost-benefit of setting up and conducting a specific EDEx for prioritization in planning (not a value-neutral approach focusing on technology without considering economic benefits). The EDEx approach should help to identify what data to exchange, how to structure and integrate the data for exchanging.

7r.5.4 Engineering Data Logistics Process Design

To address the required capabilities in Section 7r.5.3 and the use cases in Section 7r.2, we introduce the main elements of an engineering data exchange (EDEx) process, a treatment design according to (Wieringa, 2014), based on the knowledge gathered in workshops with domain experts. The EDEx process adapts and extends the Multi-Model Dashboard process (Biffl *et al.*, 2014a) in the research scope of cooperating multi-disciplinary engineering work groups in a production system engineering project. The EDEx process is independent of a concrete implementation technology.

Figure 7r.5 gives an overview on the EDEx and operation phases. The EDEx operation phase assumes an agreement between data consumers and data providers on the data model and concepts for EDEx. Therefore, a negotiation of the data requested by consumers and the data published by providers is required, similar to a marketplace of well-defined data products. In this section, we introduce the roles and processes for a data negotiation marketplace as foundation for the data extraction and exchange between data providers and consumers. Key roles are the data consumer, the data provider, and the data curator. The *data consumer* requests data according to their local consumer data model from providers to conduct business processes more effectively or more efficiently. The *data provider* has artifacts that contain data that is relevant to a data consumer and knows how to extract from the artifacts this data following the local provider data model. The *data curator* has background knowledge on the business and relevant data models of all domain experts to mediate between data consumers and data providers using their local data models. The data curator has the capability link the local data models of consumers and providers with appropriate linking formulae.

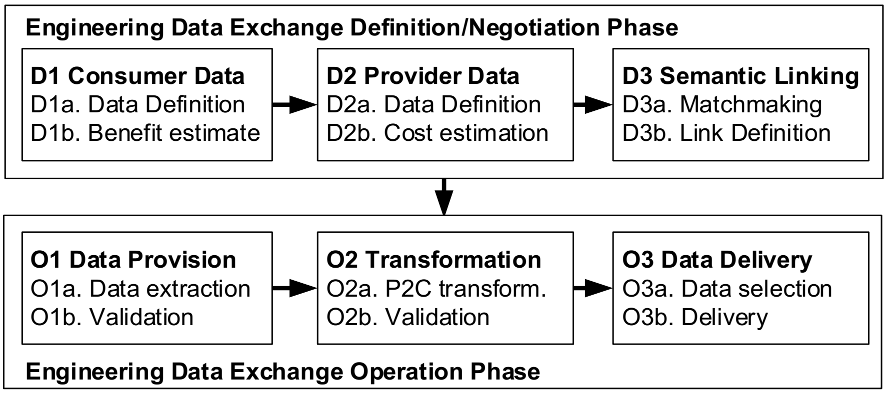


Figure 7r.5 Engineering Data Exchange process steps.

Data Exchange Negotiation Phase/Process (see illustrative example in Figures 7r.6 and 7r.7). The EDEx process consists of three main steps to identify feasible and beneficial data exchange instances.

D1. Consumer data definition and prioritization. *D1a. Consumer data definition*. Project stakeholders, who want to receive data from providers, have to define their data requests. In general, domain experts in PSE have to find out where to collect the data they need for conducting their engineering processes. Therefore, these data consumers know what data is available from which data providers. Outcome of this step is a data model of the local consumer data view, e.g., in UML, SysML, or AutomationML, or a sufficiently precise description in natural language based on the modelling concepts and vocabulary of the data consumer.

D1b. Cost-benefit estimate and prioritization. The EDEx curator validates with the consumer the definition of the requested data and estimates the likely benefit and cost of providing the data in order to focus on the most relevant EDEx instances first. Outcome of this step is a set of data model elements in the local consumer data view, with a semantic description that is understandable both to the EDEx curator and prospective data providers based on the modelling concepts and vocabulary of the EDEx curator (see Figure 7r.7 for examples). Note that this step can be repeated, if data consumers need to define additional data elements later in the project. A required mechanism for this step is a team workspace (see Figure 7r.6) that allows sharing the data requests on project level with prospective data providers. The EDEx overview (see Figure 7r.7, tag D1) shows the status of the data elements agreed for provision.

D2. Provider data definition and cost estimation. *D2a. Provider data definition as source for consumer data*. An EDEx provider can react to consumer data requests by *agreeing* to publish data that is semantically equivalent to (parts/aspects of) the requested consumer data. In general, providing the data will involve extracting the data elements from suitable engineering artifacts, often export results from a specific engineering tool, e.g., the mechanical structure of a work cell. Outcome of this step is a set of data model elements in the local provider data view, with a semantic description that is understandable both to the EDEx curator and prospective data providers based on the modelling concepts and vocabulary of the data provider (see Figure 7r.7 for examples).

D2b. Data provision cost estimation. Extracting data from engineering artifacts can take significant effort and cost, even to an expert. Therefore, the data curator has to validate that the provided data is equivalent to (relevant parts/aspects of) requested data items and elicit the likely cost for data extraction and transformation in a format that is suitable for EDEx, such as AutomationML. Outcome of this step is feedback to the provider whether the data is of sufficient quality and cost to continue setting up the EDEx. The EDEx overview (see Figure 7r.7, tag D2) shows the status of the data elements *agreed* for provision.

D3: Consumer-provider mediation and semantic link definition. *D3a. Economic matchmaking between data consumers and providers.* For each promising consumer data request, the EDEx curator tries to find sets of data providers that would allow providing the requested data covering both the required and available data themselves and the technical capability, such as data exchange formats) applicable to exchange the data. In the simplest case, one provider can provide the requested data in exactly the required data format. However, in typical cases, the data elements will come from several data providers in a variety of data formats (see the example in Figure 7r.6). Outcome of this step is a set of EDEx providers that could, together, provide the input data for transformation into the requested data elements. If there are several solutions, the solution options could be ranked by data quality and cost considerations.

D3b. Semantic linking between consumer and provider data models. For a suitable set of data providers that would allow providing the requested data, the EDEx data curator tries to establish for each requested data item a formal semantic link, i.e., a formula that specifies how to calculate the consumer data item value from one or more *published* provider data item instances using the modelling concepts and vocabulary of the EDEx curator. A semantic link formula can describe in a simple case, semantic identity between provider and consumer data elements. More advanced semantic relationships (Kovalenko and Euzenat, 2016) include basic string operations, mathematical calculations, and parameterized function calls to semantic transformation algorithms (see Figure 7r.9). Outcome of this step is a set of customer data, semantically linked to a set of provider data as foundation for designing the EDEx operation. The EDEx overview table (see Figures 7r.6 and 7r.7, tag D3) shows the status of the linked data elements. The EDEx process provides the foundation for conducting the *EDEx Operation* process.

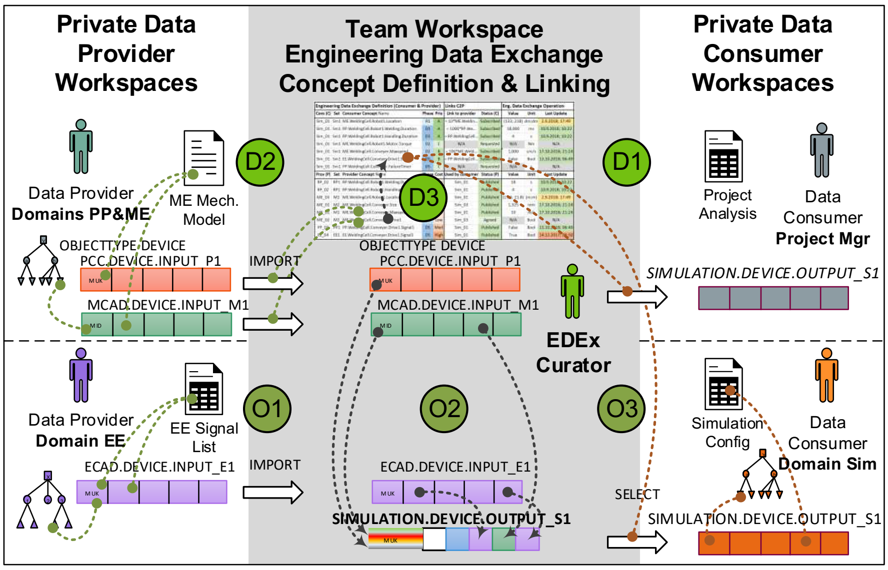


Figure 7r.6 Engineering data exchange definition/operation for one customer data set (based on (Biffl *et al.*, 2018b); tags in green circles refer to EDEx steps in Figure 7r.5).

EDEx Operation Phase/Process (see illustrative example in Figures 7r.6 and 7r.7). EDaLIS data structure of consumers subscribing to provider data enables flexible data exchange in engineering.

O1. Data provision and validation. *O1a. Data extraction and transformation*. The data provider extracts the data elements as agreed in the EDEx Negotiation process from their local engineering models and/or engineering tool outputs. Then the data provider transforms the extracted data into a data model and format that the EDEx IS can import (see Figure 7r.6, tag O1). Outcome of this step is a data set for import into the EDEx IS.

O1b. Traceable validation of data provision to data logistics. The data provider and the EDEx curator agree on a procedure to validate the data from extraction to input to the EDExIS to ensure that only correctly transformed data is imported. The data curator imports valid data into the EDEx IS. Outcome of this step is the import of valid data into the EDEx IS and feedback to the data provider on the validity of the provided data. The EDEx overview (see Figures 7r.6 and 7r.7) shows the status of the imported data elements.

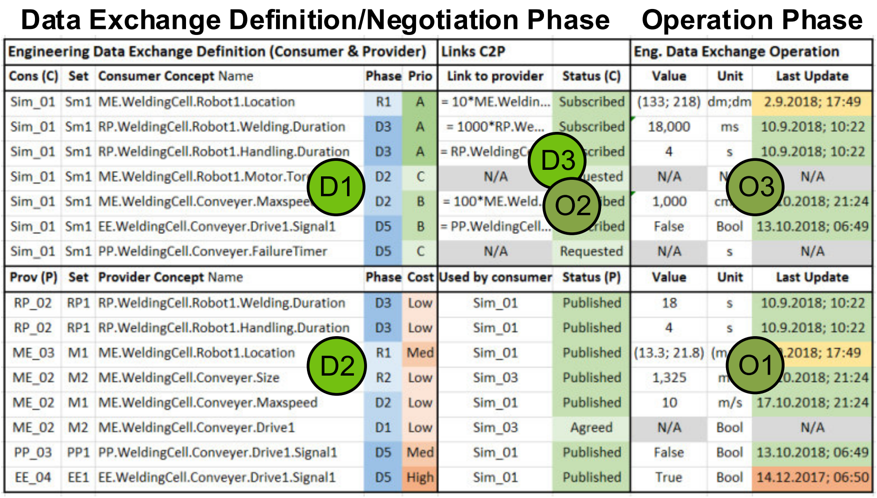
O2. Data transformation and validation. *O2a. Semantic transformation of provider to consumer data model*. The EDEx IS propagates the provided data along the semantic links to fill in or update consumer data sets (see Figure 7r.6, tag O2).

Outcome of this step are updated consumer data sets.

O2b. Validation of semantic transformation. The data curator can follow the propagation of the provided data along the semantic links to consumer data sets to check the correctness of the transformation. Outcome of this step is feedback on the validity of the semantic transformation of the most recently imported provider data set.

O3. Data selection and delivery. *O3a. Data selection by consumer*. The data consumer selects consumer data instances by providing the EDaLIS with the type of requested data and information to select the desired data instances, such as data identifiers or selection conditions, similar to a SQL query to a database. Outcome of this step is a set of selected data in the EDaLIS for delivery to the data consumer.

O3b. Data delivery from data logistics to consumer. Finally, the EDExIS delivers the result data to the data consumer (see Figure 7r.6, tag O3). Outcome of this step is the data set at the consumer in the agreed data format.



Engineering Data Exchange Definition (Consumer & Provider)					Links C2P		Eng. Data Exchange Operation		
Cons (C)	**Set**	**Consumer Concept** Name	**Phase**	**Prio**	**Link to provider**	**Status (C)**	**Value**	**Unit**	**Last Update**
Sim_01	Sm1	ME.WeldingCell.Robot1.Location	R1	A	= 10*ME.Weldin...	Subscribed	(133; 218)	dm;dm	2.9.2018; 17:49
Sim_01	Sm1	RP.WeldingCell.Robot1.Welding.Duration	D3	A	= 1000*RP.We...	Subscribed	18,000	ms	10.9.2018; 10:22
Sim_01	Sm1	RP.WeldingCell.Robot1.Handling.Duration	D3	A	= RP.WeldingC	scribed	4	s	10.9.2018; 10:22
Sim_01	Sm1	ME.WeldingCell.Robot1.Motor.Tor	D2	C	N/A	uested	N/A	N	N/A
Sim_01	Sm1	ME.WeldingCell.Conveyer.Maxspee	D2	B	= 100*ME.Weld.	ibed	1,000	cm	0.2018; 21:24
Sim_01	Sm1	EE.WeldingCell.Conveyer.Drive1.Signal1	D5	B	= PP.WeldingCell...	ribed	False	Bool	13.10.2018; 06:49
Sim_01	Sm1	PP.WeldingCell.Conveyer.FailureTimer	D5	C	N/A	Requested	N/A	s	N/A
Prov (P)	**Set**	**Provider Concept** Name	**Phase**	**Cost**	**Used by consumer**	**Status (P)**	**Value**	**Unit**	**Last Update**
RP_02	RP1	RP.WeldingCell.Robot1.Welding.Duration	D3	Low	Sim_01	Published	18	s	10.9.2018; 10:22
RP_02	RP1	RP.WeldingCell.Robot1.Handling.Duration	D3	Low	Sim_01	Published	4	s	10.9.2018; 10:22
ME_03	M1	ME.WeldingCell.Robot1.Location	R1	Med	Sim_01	Published	(13.3; 21.8)	(m	.2018; 17:49
ME_02	M2	ME.WeldingCell.Conveyer.Size	R2	Low	Sim_03	Published	1,325	m	0.2018; 21:24
ME_02	M1	ME.WeldingCell.Conveyer.Maxspeed	D2	Low	Sim_01	Published	10	m/s	17.10.2018; 21:24
ME_02	M2	ME.WeldingCell.Conveyer.Drive1	D1	Low	Sim_03	Agreed	N/A	Bool	N/A
PP_03	PP1	PP.WeldingCell.Conveyer.Drive1.Signal1	D5	Med	Sim_01	Published	False	Bool	13.10.2018; 06:49
EE_04	EE1	EE.WeldingCell.Conveyer.Drive1.Signal1	D5	High	Sim_01	Published	True	Bool	14.12.2017; 06:50

Figure 7r.7. EDEx definition/negotiation and operation overview table (based on MMD dashboard (Biffl *et al.*, 2018b); tags in green circles refer to EDEx process steps in Figure 7r.5).

Illustrating Use cases. Figure 7r.6 illustrates an overview on the roles, engineering artifacts, and exchanged data for the EDEx definition/negotiation and operation processes (see Figure 7r.5) for one consumer data set, in this case device parameters collected for the *SimE*. The data providers and data consumers, such as the PP, ME, EE, and *SimE*, operate in *private workspaces*. The *team workspace* contains shared data views as foundation for preparing and operating the EDEx processes.

Parameter exchange for production system simulation. In this use case, the *SimE* requires a set of parameters to configure the simulation for a device (see Figure 7r.6, lower right hand part, red bar), such as a robot or conveyer. The *SimE* requests the set of parameters from providers, such as the PP, ME, EE, and CP, who may agree and publish their local engineering data corresponding to a consumer request (see Figure 7r.7, left hand part). Then the *EDEx curator* links the set of parameters requested by the *SimE* with the set of parameters published by the PP, ME, EE, and CP (see Figure 7r.7, middle part for the ME and EE data) to enable the EDEx operation.

During the EDEx operation phase, the *team workspace* receives updates of provider data instances in engineering artifacts from the *private workspaces* of the PP, ME, EE, and CP (see Figure 7r.6, left hand side for the ME and EE) and transforms this input data according to the semantic links into output data for delivery to the *SimE* (see Figure 7r.6, right hand side, and example output data in Figure 7r.7, right hand upper part). The *SimE* can be notified as soon as relevant data for a requested data set is available or changed, so the *SimE* can consider when to retrieve which part of the currently available data.

Production system engineering project monitoring. In this use case, the PM can benefit from the EDEx for simple and advanced analyses. A simple analysis could be to subscribe to the same data sets as the *SimE* and analyze at specific points in the project for which data elements the engineering data is expected but missing.

Figure 7r.7 shows a snapshot of the EDEx overview table during operation: data instances coming from the providers have been processed according to the linking formulae to fill in data instances for consumers (tags O1, O2, O3). For consumers, the EDEx overview (tag D1) shows the status of the data elements as *requested*, *agreed* for provision, or *subscribed* for delivery. The EDEx overview table (tag D3) shows the status of linked data elements. For a requested data element, there may be several providers; therefore, the EDEx overview table (see Figure 7r.7) indicates the cost of providing a data element and the engineering process phase, in which the data will be available with sufficient precision, to support making an informed choice on the best provider. For example, *EE...Signal1* could be obtained from *PP...Signal1* at lower cost.

The concepts illustrated in Figures 7r.6 and 7r.7 are the foundation for prototype designs as input to the evaluation with domain experts in Section 7.r.7.

7r.5.5 Engineering Data Modelling with AML-1 and AML-2

One necessary foundation of the EDaL is the appropriate modelling of the engineering data within the different involved disciplines and within the data logistics. There for the view based approach presented in (Lüder *et al.*, 2018b).

Obviously within the EDaLIS only the engineering artifacts of data providers and consumers and the central data storage are relevant to be considered. While the data model of the central data storage has to be the union of the engineering data models of the individual tools.

Thus, in the case of the intended EDaL we can postulate two types of data models, Type 1 models and Type 2 models that can be represented by AutomationML both.

Type 2 models (identified as AML2 data models) correspond to the engineering artifact data models of the involved engineering tools. They can be modelled by a tool related set of role classes and interface classes to cover the relevant conceptual objects and system unit classes to represent their hierarchical structuring.

Type 1 models (identified as AML1 data models) correspond to the data model of the central data storage. They represent the union of all set of role classes and interface classes of the involved Type 2 models and Type 1 special system unit classes to represent all possible hierarchical structures.

7r.5.6 Requirements for an Engineering Data Logistics Information System

From a workshop with domain experts and subsequent discussion of use cases, we derived the following requirements for EDExIS and EDaLIS mechanisms.

Capa EDExIS1. EDEx management and overview. For managing the EDEx process, the EDExIS has to provide a mechanism providing the capabilities of the *EDEx overview table* illustrated in Figure 7r.7, including *EDEx definition functions* to request, agree on providing, publishing, and subscribing to data elements (see EDEx process steps D1 to D3), as well as setting relevant attributes of and searching the table for understanding the status of the EDEx definition in the project team.

Capa EDExIS2. EDEx data definition languages. For EDEx definition, the EDExIS has to process the languages for the specification of consumer and provider data sets using the modelling concepts and vocabulary of them, and the language for semantic link definition specifying (a) the dependencies between consumer and provider data sets and (b) the transformation of imported provider data into consumer data best based on the modelling concepts and vocabulary of the EDEx curator.

Capa EDExIS3. EDEx operation capabilities. For conducting the EDEx operation steps, the EDExIS has to be able (a) to import and validate provider data, (b) to store imported data versions including their metadata for processing, (c) to analyze the data and semantic links in order to correctly propagate the provider data to consumer data structures, and (d) to select and export consumer data.

Capa EDaLIS 1. Validation and versioning of exchanged engineering data. The EDaLIS should provide the capability to define validity conditions for exchanged data elements as foundation for checking the validity of data elements along the EDaLIS process. The EDaLIS should provide the capability to compare engineering data versions as foundation for the detection and analysis of changes.

Capa EDaLIS2. Consistency checking and change propagation. The EDaLIS should provide the capability to define consistency checks between semantically related provider data, knowing that these relationships and checks may differ according to the engineering phase, e.g., inconsistencies or large differences between disciplines may be ok in an early engineering phase but not in a later engineering phase. The EDaLIS should provide the capability to check the consistency between semantically related provider data and report the results as foundation for a systematic conflict detection and resolution process. The EDaLIS should provide the capability to define rules for change propagation between semantically related provider data as foundation for a semi-automated change propagation process.

Capa EDaLIS3. Provider and consumer notification. The EDaLIS should provide the capability to define notifications to providers and consumers on changes that are relevant to them as foundation for awareness in the engineering team on changes to relevant data and for analysis that support the effective and efficient resolution of missing, invalid, or inconsistent data while preventing unwanted notifications.

Together, the capabilities for an EDExIS and an EDaLIS provide the foundation for considering information system design options.

7r.6 Data Logistics Information System Design

This section discusses main design elements for effective and efficient *EDaL information system* (EDaLIS) mechanisms to address the requirements identified in Section 7r.5.6 on capabilities for data set specification and for the representation of dependency relationships as foundation for data integration and transformation. We discuss main design elements of an EDaL information system to provide these engineering data exchange capabilities for automating the EDaL process. As there is no suitable out-of-the-box technology to link discipline-specific views on data, we introduce a software architecture with a data model based on *AutomationML* data models that address these challenges.

The EDaLIS provides capabilities for the EDEx operation phase (Biffl et al, 2018b). We assume that the EDaLIS can handle AutomationML (AML) files in the so-called AML-2 and AML-1 formats (Biffl *et al*., 2018b). The data curator models AML-2 and AML-1 templates in the *EDEx Definition Phase*.

The AML-1 data model defines the central/core model of the EDExIS to transform data between several providers and consumers. Therefore, the AML-1 data model needs to represent several discipline-specific hierarchies that share some common concepts, such as machines or devices. The AML-1 data model consists of a set of AutomationML *RoleClasses*, *InterfaceClasses* and *SystemUnitClasses*. *RoleClasses* and *InterfaceClasses* define the data types and elements for discipline-specific hierarchies that the data curator can build on to define EDEx data flows.

An AML-2 data model defines a discipline-specific view on a provided engineering artifact. AML-2 uses a subsect of the AML *RoleClasses*, *InterfaceClasses* and *SystemUnitClasses* defined in the AML-1 core model to model the structure and content of an engineering artifact. The AML-2 data model is the foundation to configure a transformer that transforms an input engineering artifact into an AML-2 data structure. Therefore, there is a specific AML-2 data model for each engineering artifact.

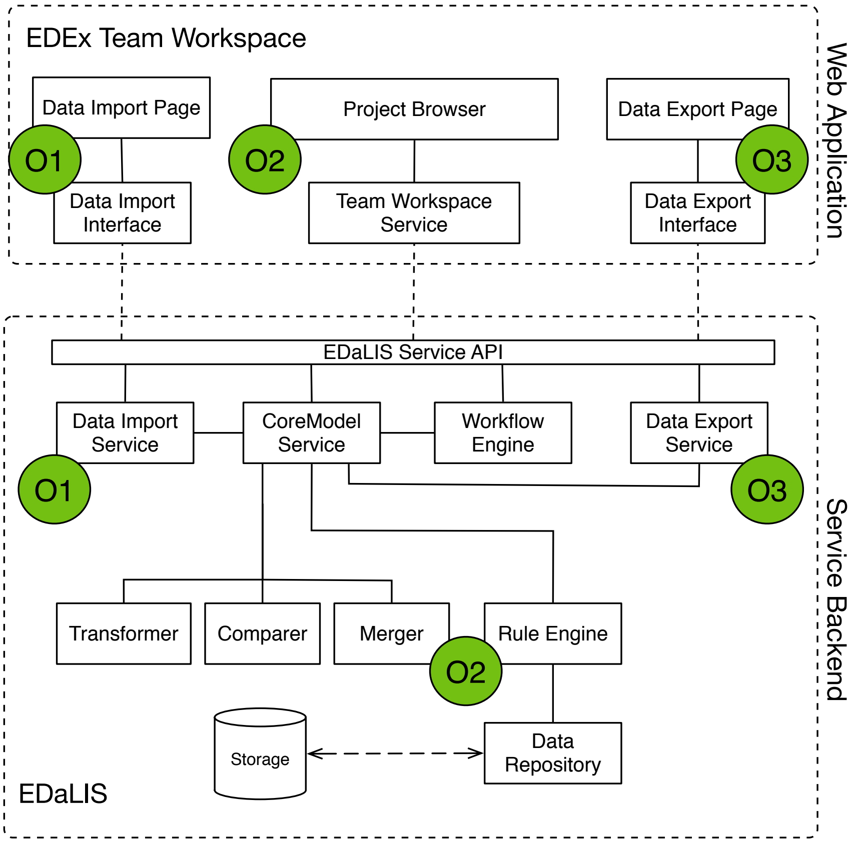


Figure 7r.8: System architecture overview of an EDaLIS system.

7r.6.1 EDaLIS System Components

Figure 7r.8 gives an overview on the conceptual system design of an EDaLIS. The EDaLIS consists of two main components: The service-oriented *backend* exposes the system capabilities, and the *web application* is the entry point for data consumers and data providers (and the data curator).

The *web application* represents the *EDEx team workspace* consisting of several pages for data consumers and providers, such as the *Data Import Page*, the *Project Browser*, and the *Data Export Page*. The web application communicates via software interfaces and the EDaLIS service API with the *backend*, which consists of the *Data Import Service*, the *CoreModel Service*, the *Merge Service*, the *Transformation Service*, the *Validation Service*, the *Merge Service*, the *Data Repository*, a *Workflow Engine*, a *Rule Engine* and the *Data Export Service*. The *EDEx team workspace* facilitates the import of provider data by communicating via the EDaLIS service API, as well as the export of required data to data consumers. The *Project*

Browser allows displaying an overview on the AML-1 data and results of selected data analyses, such as changes to data instances the core model.

In the backend, the *CoreModel Service* orchestrates the communication with the different services: AML-2 input data is validated by the *Validation Service* and *Rule Engine*, changes to data in the AML-1 core model are compared via the *Compare Service* and merged by the *Merge Service* to achieve a consistent new AML-1 data version for storing in the repository.

7r.6.2 EDaLIS Contributions to the EDEx Operation Phase

In this subsection, we assume the *EDEx Definition Phase* (Section 7r.5.4, steps D1 – D3) to be completed by the involved data providers and data curator. According to the EDEx operation phase, we discuss for each step the EDaLIS contributions.

EDEx O1. Data provision and validation. *O1a. Data extraction and transformation*. The data provider prepares an engineering artifact for import into the EDaLIS web application via the *Data Import Page*. In the future, the EDaLIS can be integrated with engineering tools to automate this process step by automatically transforming the engineering tool data into AML-2 and importing the AML-2 data into the system.

O1b. Traceable validation of data provision to data logistics. The data provider uploads the engineering data via the web application, from where it is transported to the backend for transformation and *validation*. If the data is valid, the *Compare Service* compares the new dataset to the current core model. The result is a list of changes that the provided data would cause to the core model, displayed in the *Project Browser*, from which the data provider can select the changes that should be merged into the AML-1 core data repository.

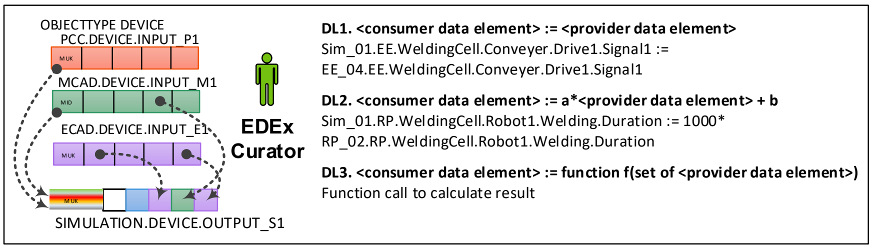


Figure 7r.9. Semantic link definition between consumer and provider data models.

O2. Data transformation and validation. *O2a. Semantic transformation of provider to consumer data model*. The selected changes from the provider data are merged into the AML-1 core model data with the *Merge Service*. The *Transformation Service* links the provider data to the consumer data sets. Therefore, the role classes of the provider data were mapped during the *EDEx Definition Phase* to the role classes in the core model. Figure 7.r9 displays examples of semantic link definitions between consumer and provider data models. In simple cases the transformation just requires converting the input values to appropriate scales of units, more advanced links may require defining and evaluating the results of complex algorithms.

O2b. Validation of semantic transformation. In this step the validity of the semantic transformation is checked by the *Rule Engine*, *e.g.,* that all links in the core model are set correctly. The data curator can check the validity of the content in the AML-1 repository against the original input data in the provided engineering artifact.

O3. Data selection and delivery. *O3a. Data selection by consumer.* The consumer can request certain data via the project browser. This can be realized by SQL[6]-like queries, or similar to XPATH[7] to specify the required data.

O3b. Data delivery from data logistics to consumer. The requested data can finally be exported to a consumer AML-2 representation and downloaded via the EDaLIS *Data Export Page* to the private workspace of the consumer.

7r.6.3 EDaLIS System Support for EDaLIS Mechanisms

According to the requirements in 7r.5.6 we investigate the capabilities of the EDaLIS mechanisms.

Capa EDExIS1. EDEx management and overview *Private and team workspace*. The EDaLIS serves as an interface between the private and the team workspaces. All involved parties (data curator, provider, and consumer) have one single point of entry for the required data, the *EDEx team workspace*. The workspace can display both discipline-specific views as well as the common model. This can be achieved by implementing a web application using the *Spring Boot*[8] framework for resource and service orchestration as well as providing the REST-interfaces[9] (*Representational State Transfer*).The system can also manage requests and subscriptions of consumers, and publishing data by providers, e.g., based on the *publish-subscribe*d design pattern.

Capa EDExIS2. EDEx data definition languages. The discipline-specific views share common concepts, such as machines, devices, and signals, which link the views across disciplines; however, the discipline-specific views differ in their hierarchies, such as the hierarchy of the mechanical structure for the ME, the hierarchy of electrical circuit areas for the EE, or the hierarchy of software functions for the CP. Furthermore, links, interfaces and roles need to be displayed. An adequate EDEx data definition language needs to facilitate the appropriate representation of such data structures. File formats such as CAEX or AutomationML (AML) were developed for such industry use cases.

---

[6]https://standards.iso.org/ittf/PubliclyAvailableStandards/c053681_ISO_IEC_9075-1_2011.zip

[7] https://www.w3.org/TR/xpath-31/

[8] http://spring.io/projects/spring-boot

[9] https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest

Capa EDExIS3. EDEx operation capabilities *Data import***.** The EDaLIS needs to be able to import and transform provided according to the agreed upon data formats and common concepts. Data input formats could be CSV spreadsheets or XML files, or AML-2 models (Biffl *et al*., 2018b). Combined with the data export this allows addressing *UC 1 Artefact provision* introduces in Section 7r.2.1.

Storing of input data. Another essential capability is to store the input data in order to process it. The transfer can be handled by REST or a similar data transfer protocol to an XML database to store it in AML-1 data structure, a graph consisting of linked discipline-specific trees.

Analysis of data and semantic links. The logic of the EDaLIS must be capable to analyze the input data and make the semantic links to transform the provider data to the given consumer data structures to represent them in the AML-1 core model. This transformation requires semantically similar attributes and identifiers (common concepts) and can be specified for processing by *XPath* for accessing XML-based files.

Selection and export of data. The enrichment of data in the context of RTE is an essential feature of an EDaL process, to enable the backflow of information. Therefore, the EDaLIS also needs to provide an export function, to download specific and general views of the core model in valid file formats such as AML-1 or CSV/XML-files.

Capa EDaLIS 1. Validation and versioning of exchanged engineering data. *Validator of input data***.** The core model service validates the provider data by checking the engineering artifacts with the given core model. Due to the parallelism of the multiple disciplines this validation is essential to verify, e.g., attributes to be compliant with the current core model. If this is not the case the merge process cannot be completed.

Versioning of input data. Versioning of engineering data is an important aspect of the EDaLIS. The repository allows storing and versioning engineering data (including meta data) as commonly known in software engineering by version control systems to compare different engineering data versions. Commonly used version control technologies such as *Git*[10] or SVN[11] can be used. Clear benefits are the traceability of changes and possibility to roll back to older versions if inadequate or incomplete data has been imported by a data provider, which enables *UC2b sequential enrichment with updates* in Section 7r.2.

Capa EDaLIS2. Consistency checking and change propagation. A language such as the *object constraint language*[12] (OCL) can be used in the system design to check automatically whether the engineering data complies with previously defined constraints, e.g., that all semantic links have to be connected in the AML-1, or to identify and resolve dead links.

---

[10] https://git-scm.com
[11] https://subversion.apache.org
[12] https://www.omg.org/spec/OCL/

The system should be able to check whether specific attributes or elements have valid attributes, e.g., rotation speed cannot be negative, which could also be used to model dependencies between components. Change propagation is executed by the display of the change attributes. Together, these mechanisms enable *UC3 parallel enrichment of engineering artifacts* in Section 7r.2.1.

Capa EDaLIS3. Provider and consumer notification. For this use case a workflow engine such as *Camunda*[13] or *Activiti*[14] can be integrated into the EDaLIS to further automate the EDaL process, e.g., by automatically notifying providers that data is requested from them or consumers as soon as the data provider has imported the required data into the system.

All in all, the EDaLIS supports the EDaL use cases, implementing all mechanisms required for addressing *UC4 backflows of artifacts* in Section 7r.2.1.

7r.7 Evaluation

This section derives a conceptual evaluation for the EDaL, and reports on the evaluation of the *engineering data exchange* (EDEx) process and requirements (a) in an initial feasibility case study (Runeson and Höst, 2009) with domain experts at a large *production systems engineering* (PSE) company, a systems integrator for metallurgic production systems, and (b) in a cost/benefit comparison of the EDEx definition and operation processes to the traditional process of point-to-point exchange of engineering artifacts between domain experts, closing an iteration of the design cycle (Wieringa, 2014) and providing knowledge for guiding future research.

7r.7.1 Conceptual Evaluation of the EDaL Process Study

Goal of the conceptual evaluation is to discuss to what extent the EDaL process introduced in Section 7r.5.2 allows addressing the EDaL requirements in Section 7r.5.1 regarding the use cases introduced in Section 7r.2.

EDaL Step 1, EDaL requirements analysis, addresses the capability *EDaL scope analysis* by systematically collecting the candidates for data consumers and providers as well as an initial set of data that could be exchanged between consumers and providers. Chapter 6r in this book describes a method for deriving a data processing map that helps identify engineering artifacts and the relevant engineering data they contain.

EDaL Step 2, EDaL use case design, addresses the capability *EDaL use case analysis* by systematically considering design patterns, such as the RTE pattern and the engineering backflow pattern, to identify a complete set of EDEx data flows for a use case context. The simple EDaL language describing an EDEx flow as *data provider (artifact: data set) -> data consumer* allows defining the main elements

[13] https://camunda.com/de/
[14] https://www.activiti.org

for a data flow as foundation for a more detailed analysis in the EDEx process definition phase. This approach goes beyond the EDEx approach to ensure that all relevant EDEx data flows are considered to address an EDaL use case.

EDaL Step 3, List of EDEx flows, details the data flows as input to the EDEx definition phase as foundation for the consumer-driven design and implementation of the data flows. The EDaL list of data flows can act as a checklist to ensure the EDEx process to finally result in a network of EDEx data flows that allow fulfilling the EDaL design pattern required for addressing the required use cases.

Therefore, the EDaL process elements, based on the underlying EDEx process, allows addressing the general use cases introduced in Section 7r.2 *UC1 Artifact provision, UC2/3 Sequential/Parallel enrichment of artifacts*, and *UC4 Backflows of artifacts* as well as the specific use cases *UC Sim. Data exchange for production system simulation* and *UC PM. Production system engineering project monitoring*.

7r.7.2 Feasibility Study of EDEx process

Goal of the feasibility study is to evaluate the basic concept of the EDEx process with domain experts by following the steps of the EDEx process description (see Section 7r.5.4 and Figure 7r.5). Based on the use cases introduced in Section 7r.2.2, we designed prototypes of selected user interface elements, such as the overview table, data specification, linking, and retrieval as electronic mock up artifacts with data from domain experts. We collected data on the usability and usefulness of the EDEx process based on the *Technology Acceptance Model* questionnaire (Davis, 1985; Biffl *et al*., 2018).

Further, we developed technology prototypes of the IS capabilities to explore the feasibility of designing the EDaLIS concepts with available technologies, including *AutomationML* for data specification (Lüder *et al*., 2018a), an *Excel* dialect for the specification of dependency links, *Java* code for transformations, and *BaseX* as data storage. We conducted and discussed the EDEx steps in a workshop with domain experts representing the roles *data provider* (PP, ME, EE, CP in the use cases), *data consumer* (*SimE*, PM), and *EDEx curator*.

Overall, the domain experts found the EDEx process feasible, useful, and usable for basic cases that make up most of the data exchange use cases in their typical project context, assuming that the EDaLIS provides effective tool support to automate the data transformation, storage, and selection tasks. The domain experts provided improvement suggestions for the user interfaces, and for describing the data transformation and linking formulae in their context. Further, the domain experts noted that more complex cases may take considerable effort to design and automate; therefore, cost-benefit estimates in the EDEx process are important to guide planning the EDEx implementation. Nevertheless, they indicated that more advanced cases, such as the EDaL use cases described in Section 7r.2.1, will also enable more advanced engineering data usage exploiting trusted and quality ensured data enabling automation of engineering steps leading to significant cost reductions within the overall engineering process.

7r.7.3 Cost-Benefit Considerations

To evaluate the costs and benefits of the EDEx process via a *team workspace* in comparison to the traditional manual process of point-to-point e-mail based EDEx, we elicited needs and estimates from domain experts, who are responsible for engineering and project management of large-scale metallurgic production system projects.

Table 7r.1 presents an overview of the findings for of the EDEx process steps in the use case *Parameter exchange for production system simulation* by comparing the effectiveness, i.e., correctness of results for a task, and the effort of a stakeholder conducting a task. We applied a 5-point Likert-Scale (++, +, o, -, --), where "++" indicates very positive effects, and "--" very negative effects. Positive effects refer to high effectiveness of the investigated approaches and to low effort for implementation and application.

Table 7r.1. Comparison of the effectiveness and effort of traditional manual and EDEx processes. Legend: ++ very good, + good, o average, - weak, -- very weak.

	Effectiveness		Effort	
EDEx Process Step	Manual	EDEx	Manual	EDEx
D1. Consumer data definition & prioritization.	o	+	+	o
D2. Provider data definition & cost estimation.	--	+	+	-
D3. Consumer-provider semantic link definition.	--	++	N/A	-
O1. Data provision and validation.	-	o	+	o
O2. Data transformation and validation.	--	+	--	+
O3. Data selection and delivery.	-	++	--	++

Regarding effectiveness, the EDEx process was found effective to very effective by the interviewed stakeholders, both providers and consumers, because they were able to exchange data elements in a traceable and validated way. In the traditional approach, the data consumers had to define, procure, transform, and validate the required data with significant effort and prone to errors. However, the application of the EDEx process requires additional effort, especially during the EDEx definition (D2) and linking (D3), in particular for providers and for the new role of the *EDEx curator*.

On the upside, the results of the linking step (D3) significantly improve the representation of shared knowledge in the engineering team regarding an overview on the dependencies between the engineering roles on data element level. Domain experts and the PM can always get a current overview on the status of data deliveries and can identify missing engineering data and unfulfilled requests by consumers. In addition, the EDaLIS can provide the benefit of immediate feedback on changed engineering data elements efficiently, without additional effort by the domain experts.

7r.8 Discussion

This section discusses the evaluation results regarding the research questions introduced in Section 7r.4and compares the results to related work.

RQ1.: What are main elements of an Engineering Data Logistics (EDaL) approach in round-trip System Engineering?

Section 7.r.5.2 introduced the EDaL elements in the process for EDaL, an EDaL *curator* identifying data providers and consumers, their candidate engineering artifacts and data to exchange according to EDaL design patterns, described by an EDaL specification language. In an initial conceptual evaluation, we found the EDaL approach adequate to address the core use cases for *round-trip System Engineering* introduced in Section 7r.2.1, assuming an effective underlying EDEx process for the data flow between a consumer and her data provider(s).

As a next step in the design science approach, the initial conceptual evaluation will be the foundation for an empirical study to investigate what methods and mechanisms typical domain experts will require to apply the EDaL approach effectively in their engineering context.

RQ2. What are main elements of an effective and efficient engineering data exchange (EDEx) process in Multi-Disciplinary System Engineering?

Section 7r.5.4 introduced as main EDEx process elements EDEx roles, process steps, and data structures. The new role of the *EDEx curator* mediates between data consumers and providers. In the feasibility study, a domain expert filling this role informally was identified. The EDEx data structures represent the necessary knowledge on engineering data, semantic links between consumer and provider data, and the status on the EDEx process as foundation for effective EDEx for the use cases introduced in Section 7r.2.2 and according to the required capabilities for EDEx in multi-disciplinary engineering, discussed in Section 7r.5.3. Further, the EDEx process facilitates efficient EDEx (a) by considering the benefits of EDEx for consumers and the cost for providers to focus first on the data sets with the best cost-benefit balance and (b) by automating the EDEx operation with support by the EDaLIS.

As potential drawback of the EDEx process, the domain experts noted the need to convince data providers to take over the task and extra effort of extracting requested data from their engineering artifacts. For this task, specific tool support will be required according to the project context as well as appropriate compensation for the extra effort. A company internal cost balancing scheme shall be investigated enabling the transfer of cost reductions at consumer side to the data provider side that can be organized by the EDEx curator.

From a data model point of view, the local data models in a discipline-specific view of a provider or consumer are, in general, trees. The common model in EDaLIS links these trees to a graph via semantically equivalent concepts, such as system part, device, or signal. However, the effective and efficient identification of relevant semantically equivalent concepts may take considerable effort and requires research on methods for supporting the EDEx data curator.

RQ3. What are main information system mechanisms that enable engineering data logistics for Multi-Disciplinary System Engineering?

The EDaLIS mechanisms for management and overview, data definition languages, and operation capabilities addressed the requirements for EDEx capabilities in Section 7r.5.6 on a conceptual level. Together, the EDaLIS mechanisms facilitate efficient round-trip-engineering among domain experts, i.e., the enrichment of common engineering concepts in iterations from several disciplines (use cases 2 and 3 in Section 7r.2.2), as the domain experts may act both as consumers and providers.

The design of an operational EDaLIS will have considerable impact on the efficiency of the EDEx process in the application context and requires further investigation regarding the interfaces to domain experts and their tools, regarding the languages to specify EDaL and EDEx aspects, and regarding data structures to process and store the data required for addressing the EDEx and EDaL use cases.

Limitations. As all empirical studies the presented research has some limitations that require further investigation.

Conceptual Evaluation of EDaL. We evaluated the EDaL concepts with typical use cases in the context of a large PSE company. However, these use cases may be specific to the company and not representative for typical PSE companies. Therefore, we plan to evaluate the EDaL concepts in a wider set of representative PSE companies.

Feasibility study. We evaluated the EDEx process approach with focus on specific use cases in cooperation with domain experts in a typical large company in PSE of batch production systems that can be seen as representative for systems engineering enterprises with project business using a heterogeneous tool and technology landscape. The evaluation results are based on observations from a limited sample of projects, stakeholder roles, and data models. To overcome these limitations, we plan a more detailed investigation in a wider variety of domains and application contexts.

The *expressiveness of data specification and linking languages*, used in the evaluated prototype, can be considered as a limitation. The prototype is able to address an initial set of simple data types, while industrial scenarios showed that value ranges and aggregated ranges have to be expressible in the desired data and link languages for specification and validation. While the evaluation worked well with data provided in tables, the evaluation of advanced data structures such as trees or graphs remains open.

7r.9 Summary and Outlook

This section summarizes the findings of the book chapter and proposes future research work.

Digitalization in *production system engineering* (PSE) (Vogel-Heuser *et al.*, 2017) aims at enabling flexible production towards the *Industry 4.0* vision and at shortening the engineering phase of production systems. This results in an increase of parallel PSE, where the involved disciplines have to exchange updates of engineering information for synchronization due to dependency constraints between the engineering disciplines.

In this chapter, we introduced and investigated PSE use cases for *engineering data logistics* (EDaL) and for the *engineering data exchange* (EDEx) process to provide domain experts in parallel PSE with a systematic approach to define and efficiently exchange agreed sets of data elements between heterogeneous local engineering models as foundation for agile, traceable, and secure PSE. EDaL and the EDEx process provide the foundations for addressing the major challenges introduced in Section 1.

C1. Data exchange requirements are not clear or conflicting. The EDEx definition phase results in an EDaL network of stakeholders linked via data representing engineering information they exchange as foundation for EDaL patterns, such as RTE. This EDaL network enables potential data providers to understand better which project participants require what kind of data at what point in time in the project. The EDaL network can grow iteratively, going beyond the insight of a one-time process analysis, as the specific relations between engineering artifacts and their content within an engineering project can change during the project execution. The data in the EDaL network enables the analysis of stakeholder priorities and relationships in an engineering project to provide the knowledge on which stakeholders require what data by when in the PSE process. The analysis of data exchange requirements allows identifying and addressing conflicts between data providers and consumers on the extra effort for efficient EDaL .

Ch2. Heterogeneous engineering data is hard to integrate for sharing. While the data provided by engineering tools is typically specific for a discipline and not designed for use with other disciplines or with the project they contribute to, semantic linking allowed the integration of heterogeneous data in the evaluated EDEx use cases. The semantic linking enables seamless traceability in the EDEx process that, for the first time, gives all stakeholders the opportunity to know and analyze which role provided or received which kind of engineering data, which addresses a major awareness shortcoming in the traditional EDEx process. EDaL support for the EDEx in an engineering team can build on the explicit representation of common concepts as semantic links between heterogeneous engineering data sets enable automation of EDEx and analyses. Further, the EDEx semantic linking improves the representation of shared knowledge in the engineering team in a way that is understandable for machines, a prerequisite for introducing Industry 4.0 applications by supporting knowledge preservation in an aging engineering society.

C3. RTE changes on engineering data are hard to trace and analyze. The EDaL approach provides a data processing map, a network of stakeholders and the engineering artifacts and data they exchange during the engineering process, as foundation for automating analysis of changes to the content of exchanged data. Therefore, a data consumer in the RTE process can efficiently track back changes in the data versions s/he receives from several data sources to enable analyses of the received data and meta data, e.g., for identifying missing or inconsistent data. The EDaL support in the EDaLIS keeps track of EDEx flows, including roles and rules for process conduct. Therefore, the EDaLIS facilitates frequent synchronization between work groups to reduce the risk of divergent local designs, rework, and project delays. The price to pay is the introduction of a new stakeholder role, the EDEx curator, having the knowledge and responsibility to coordinate the EDEx definition phase and to supervise the EDEx operation process.

Future Work. We foresee the following avenues of future research work to investigate applications of the EDaL and EDEx capabilities and to address limitations of the research in this work.

Case study on EDaL concepts. To explore the EDaL approach, we will conduct an empirical study to investigate what methods and mechanisms typical domain experts will require to apply the EDaL approach effectively in their engineering context.

Advanced analyses on the exchanged data and associated metadata. The EDEx data will enable consumers and researchers to conduct advanced analyses, such as on expected but missing values, data validity and consistency, and symptoms for security risks. The EDEx metadata allows analyses of PSE process characteristics.

Semantic linking between consumer and provider data models. During the use of EDEx, the complexity of links may grow considerably with the number of data elements, consumers, and providers, which will require research on the scalability of EDEx. While the EDEx process identifies direct links between consumer and provider data sets, it may be more efficient on a larger scale to identify common concepts (Sabou *et al*., 2017) in the engineering data model and link the consumer and provider data via these common concepts.

IT Security considerations. Centralizing knowledge in the EDaLIS will require research on threats to the integrity of collected knowledge and of industrial espionage.

EDEx and EDaLIS application. Future work will include the application and evaluation of the EDEx process and an operational EDaLIS in various engineering domains and application areas.

Acknowledgment

The financial support by the Christian Doppler Research Association, the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

1. (Andersen *et al*., 2018) Andersen, A. L., ElMaraghy, H., ElMaraghy, W., Brunoe, T. D., & Nielsen, K. (2018). A participatory systems design methodology for changeable manufacturing systems. International Journal of Production Research, 56(8), 2769-2787.
2. (Biffl *et al.*, 2014a) Biffl, S., Winkler, D., Mordinyi, R., Scheiber, S., & Holl, G. (2014, September). Efficient monitoring of multi-disciplinary engineering constraints with semantic data integration in the multi-model dashboard process. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*.
3. (Biffl *et al.*, 2014b) Biffl, S., Lüder, A., Schmidt, N., & Winkler, D. (2014, October). Early and efficient quality assurance of risky technical parameters in a mechatronic design process. In Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE (pp. 2544-2550). IEEE.
4. (Biffl *et al.*, 2017) Biffl, S., Gerhard, D., & Lüder, A. (2017). Introduction to the Multi-Disciplinary Engineering for Cyber-Physical Production Systems. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems* (pp. 1-24). Springer, Cham.
5. (Biffl *et al.,* 2018a) Biffl, S., Eckhart M.,Lüder A., Müller T., Rinker F. & Winkler, D. (2018) *Data Interface for Coil Car Simulation (Case Study) Part I*; Tech. Rep. CDL-SQI-M2-TR02, TU Wien.
6. (Biffl *et al.,* 2018b) Biffl, S., Eckhart M.,Lüder A., Müller T., Rinker F. & Winkler, D. (2018) *Data Interface for Coil Car Simulation (Case Study) Part II - Detailed Data and Process Models*; Tech. Rep. CDL-SQI-M2-TR03, TU Wien.
7. (Brambilla *et al.*, 2012) Brambilla, M., Cabot, J., & Wimmer, M. (2017). Model-driven software engineering in practice. *Synthesis Lectures on Software Engineering*, *3*(1), 1-207.
8. (Calà, *et al.*, 2017) Calà, A., Lüder, A., Cachada, A., Pires, F., Barbosa, J., Leitão, P., & Gepp, M. (2017, July). Migration from traditional towards cyber-physical production systems. In Industrial Informatics (INDIN), 2017 IEEE 15th International Conference on (pp. 1147-152). IEEE.
9. (Davis, 1985) Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results* (Doctoral dissertation, MIT).
10. (Diedrich *et. al*, 2011) Diedrich, C., Lüder, A., & Hundt, L. (2011). Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen. at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik, 59(7), 426-438.
11. (Drath, 2009) Drath, R. (Ed.). (2009). Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA. Springer-Verlag.
12. (Drath *et al.,* 2011) Drath, R., A. Fay, and M. Barth (2011) "Interoperabilität von Engineering-Werkzeugen, at–Automatisierungstechnik 59." Issue 7: 451-460.
13. (Grose, *et al.*, 2002) Grose, T. J., & Doney, G. C. (2002). PhD. Stephan A. Brodsky. Mastering XMI. Java programming with XMI, XML, and UML.
14. (Hell, 2018) K. Hell (2018) Methoden der projektübergreifenden Wiederverwendung im Anlagenentwurf: Konzeptionierung und Realisierung in der Automobilindustrie , Dissertation, Fakultät Maschinenbau, Otto-v.-Guericke Universität Magdeburg.
15. (Hohpe and Woolf, 2003) Hohpe, G., & Woolf, B. (2003) "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", *Addison-Wesley Longman*.
16. (IEC 62714) IEC 62714 Engineering data exchange format for use in industrial automation systems engineering - Automation markup language, 4 Parts 1, IEC, 2014-2018.
17. (Jimenez-Ramirez *et al.*, 2018) Jimenez-Ramirez, A., Barba, I., Reichert, M., Weber, B., & Del

Valle, C. (2018, June). Clinical Processes-The Killer Application for Constraint-Based Process Interactions?. In Int. Conference on Advanced Information Systems Engineering (pp. 374-390). Springer, Cham.

18. (Kovalenko and Euzenat, 2016) Kovalenko O, Euzenat J (2016) Semantic Matching of Engineering Data Structures. In Biffl S, Sabou M (Eds.) *Semantic Web for Intelligent Engineering Applications*. Springer
19. (Lüder and Schmidt, 2017) Lüder, A., & Schmidt, N. (2017). AutomationML in a Nutshell. In Handbuch Industrie 4.0 Bd. 2 (pp. 213-258). Springer Vieweg, Berlin, Heidelberg.
20. (Lüder *et al.*, 2018a) Lüder A., J.-L. Pauly, K. Kirchheim, F. Rinker, S. Biffl; Migration to *AutomationML* based Tool Chains – incrementally overcoming Engineering Network Challenges; Proc. of 5th *AutomationML* User Conference, Göteborg, Sweden, 24./25.10.2018.
21. (Lüder *et al.*, 2018b) Lüder, Arndt; Pauly, Johanna-Lisa; Rosendahl, Ronald; Biffl, Stefan; Rinker, Felix Support for engineering chain migration towards multi-disciplinary engineering chains In: 2018 14th IEEE International Conference on Automation Sci-ence and Engineering (CASE 2018): Munich, Germany 20 - 24 Au-gust 2018$h - Piscataway, NJ: IEEE, S. 671-674
22. (OMG, 2011) OMG. (2011). Business Process Model Notation (BPMN) version 2.0. *OMG Specification, Object Management Group*, 22-31.
23. (Putze *et al.*, 2018) Putze, S., Porzel, R., Savino, G. L., & Malaka, R. (2018, June). A Manageable Model for Experimental Research Data: An Empirical Study in the Materials Sciences. In Int. Conference on Adv. Information Sys. Eng. (pp. 424-439). Springer, Cham.
24. (Rosemann and vom Brocke, 2015) Rosemann, M., & vom Brocke, J. (2015). The six core elements of business process management. In *Handbook on business process management* 1 (pp. 105-122). Springer.
25. (Runeson and Höst, 2009) Runeson, P., Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131-164.
26. (Sabou *et al.*, 2017) Sabou M, Ekaputra FJ, Biffl S (2017) Semantic Web Technologies for Data Integration in Multi-Disciplinary Engineering. In Biffl S, Lüder A, Gerhard D (Eds) Multi-Disciplinary Engineering of Cyber-Physical Production Systems, Springer.
27. (VDI 3690, 2012-2017) VDI (2012-2017) VDI-Richtlinie: VDI/VDE 3690 XML in der Automation, VDI verlag,
28. (Vogel-Heuser *et al.*, 2017) Vogel-Heuser B., T. Bauernhansl, M. ten Hompel (Hrsg.) (2017): *Handbuch Industrie 4.0*, Bände 1-4, VDI Springer Reference.
29. (Wieringa, 2014) Wieringa, R. (2014) *Design science methodology for information systems and software engineering*. Berlin [u.a.]: Springer.
30. (Winkler *et al*., 2017) Winkler D., Sabou M., Biffl S.: “Improving Quality Assurance in Multi-Disciplinary Engineering Environments with Semantic Technologies” In: L.D. Kounis “Quality Control and Assurance – An Ancient Greek Term ReMastered”, Book Chapter 8, pp.177-200, Published 16.02.2017, INTEC Publishing, 2017.
31. (Xu, 2012) Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, *28*(1), 75-86.