Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in Industrial Engineering and Management

Leveraging standard manufacturing resource representations in production simulation software

Concept, potential applications and prototypical use cases

Kusti Hämäläinen

Master's thesis
2024

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Kusti Hämäläinen

Title of thesis Leveraging standard manufacturing resource representations in production simulation software: Concept, potential applications and prototypical use cases

Programme Industrial Engineering and Management

Major Operations Management

Thesis supervisor Asst. Prof. DSc. Siavash Khajavi Haghighat

Thesis advisor(s) MSc. Fernando Ubis

Collaborative partner Visual Components Oy

Date 18.11.2024 **Number of pages** 94+4 **Language** English

Abstract

Digital Twin (DT) is a concept that includes representing a physical entity virtually, facilitating interaction with the physical entity, and augmenting knowledge of the physical entity using simulation. An asset data interoperability standard entitled Asset Administration Shell (AAS) aims to accelerate the development of DT:s by defining standardized virtual representations for different manufacturing resources, and an Application Programming Interface (API) for inter-firm exchange of data. Recent literature has illustrated production line DT:s using an AAS representation. What new applications would a standardized virtual representation or API enable when co-used with a production simulation environment?

This thesis aims to define the overlapping application areas of production simulation and AAS, this way identifying use cases for the integration and the consequent technical requirements for the simulation software. This study is conducted using open-ended interviews with ten representatives of six companies, and by building prototypical use cases that demonstrate interoperability between AAS representations and the production simulation software.

Based on the interview data, application areas of production simulation and AAS overlap in two areas: interfacing production simulation with product life cycle management (PLM) systems, and enriching the information in AAS representations during early-phase engineering of production systems. Based on the technological experimentation, applications of reusing AAS representations in simulation are limited by the lack of relevant AAS specifications in current state. Generalizable applications for AAS-to-simulation data flows should use complementary Automation Modelling Language (AML) detail models packaged in AAS:s, not make assumptions on the locations of properties in AAS representations, or wait until AAS specifications for the purpose are developed and published.

Keywords Asset Administration Shell, Digital Twin, production simulation

Tekijä Kusti Hämäläinen

Työn nimi Tuotantovälineiden standardisoidun esitystavan hyödyntäminen tuotantosimulointiohjelmassa: Konsepti, potentiaalisia sovellutuksia ja käyttötapausten prototyyppejä

Koulutusohjelma Tuotantotalous

Pääaine Operations Management

Vastuuopettaja/valvoja Apulaisprofessori TkT Siavash Khajavi Haghighat

Työn ohjaaja(t) DI Fernando Ubis

Yhteistyötaho Visual Components Oy

Päivämäärä 18.11.2024 **Sivumäärä** 94+4 **Kieli** Englanti

Tiivistelmä

Digitaalinen kaksonen on käsite, johon kuuluu fyysisen olion virtuaalinen esitys, vuorovaikutus fyysisen olion kanssa, ja sitä koskevan tiedon täydentäminen simuloidulla tiedolla. Laitetietojen yleistoimivuusstandardi Asset Administration Shell (AAS) kiihdyttää teollisten digitaalisten kaksosten kehitystä määrittelemällä standardisoituja virtuaalisia esitystapoja erilaisille tuotantovälinetyypeille, ja ohjelmointirajapinnan (API) yritystenväliseen tiedonvälitykseen. Lähivuosien kirjallisuudessa on esitelty tuotantolinjoja kuvaavia digitaalisia kaksosia käyttäen AAS-esitystapaa. Mitä uusia sovellutuksia standardisoitu virtuaalinen esitystapa tai ohjelmointirajapinta voisi mahdollistaa tehdassimulointiympäristöjen yhteydessä?

Tämän diplomityön tavoitteena on määrittää tuotantosimulaation ja AAS:n päällekkäiset sovellutusalueet, näin tunnistaen tarkat käyttötapaukset ja niistä seuraavat tekniset vaatimukset simulointiohjelmalle. Tämä tutkimus on toteutettu avoimella haastattelumenetelmällä kuuden eri yrityksen kymmenen edustajan kanssa, ja kehittämällä kokeellisia sovellutuksia tuotantosimulointiohjelman yleiskäyttöisyydestä AAS-teknologian kanssa.

Perustuen haastatteluihin, kaksi sovellutusaluetta ovat yhteisiä AAS:lle ja tuotantosimuloinnille: tuotantosimulointiohjelman yhdistäminen tuote-elinkaaren hallinnan tietojärjestelmiin, ja AAS-esitysten tiedon täydentäminen aikaisessa vaiheessa tehtaiden suunnitteluprosessia. Perustuen teknillisiin kokeiluihin, AAS-esityksiä simulaatiossa hyödyntäviä sovellutuksia rajaa asiaankuuluvien AAS-spesifikaatioiden vähyys nykytilanteessa. Yleistettävien AAS-tietoa simulaatioon tuovien sovellutusten tulisi hyödyntää täydentäviä AAS-esityksiin liitettyjä Automation Modelling Language (AML)-malleja, välttää oletuksia vakioiden sijainnista AAS-esityksessä, tai odottaa tarkoitukseen kehitettyjen AAS-spesifikaatioiden julkaisua.

Avainsanat Asset Administration Shell, digitaalinen kaksonen, tuotantosimulointi

Table of contents

Preface and acknowledgements

I want to thank my thesis supervisor Assistant Professor Siavash Khajavi Haghighat and my thesis advisor Fernando Ubis for their advice and insightful discussions on this research topic.

I would also like to thank my manager Jarkko Soikkeli for great practical help in executing this project, several colleagues for help in sampling the interviews, and every interviewee for their eager participation.

This thesis has been partially funded through the European Commission's Horizon Europe Innovation Action programs Zero-SWARM (Grant Agreement 101057083) and RE4DY (Grant Agreement 101058384).

Espoo, 18 November 2024
Kusti Hämäläinen

Abbreviations

AAS	Asset Administration Shell
AI	Artificial Intelligence
AML	Automation Markup Language
API	Application Programming Interface
AR	Augmented Reality
BoM	Bill of Materials
BoP	Bill of Process
CAD	Computer Aided Design
CPPS	Cyber-physical Production System
CPS	Cyber-physical System
DES	Discrete Event Simulation
DIAMOND	Digital Plant Modelling with Neutral Data Formats (project)
DM	Digital Model
DS	Digital Shadow
DT	Digital Twin
ERP	Enterprise Resource Planning
FA3ST	Fraunhofer Advanced Asset Administration Shell Tools
FMI	Functional Mock-up Interface
HTTP	Hypertext Transfer Protocol
I4.0	Industry 4.0
IDTA	Industrial Digital Twin Association
IoT	Internet of Things
JSON	JavaScript Object Notation
ML	Machine Learning
MES	Manufacturing Execution System
MQTT	Message Queueing Telemetry Transport
OPC UA	Open Platform Communications Unified Architecture
P&P	Plug-and-Produce
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
RAMI4.0	Reference Architecture Model Industry 4.0
REST	Representational State Transfer (API)
SCADA	Supervisory Control and Data Acquisition
SoS	System-of-Systems
W3C WoT	World Wide Web Consortium Web of Things
XML	Extensible Markup Language

1 Introduction

Discrete-event simulation (DES) has developed since the late 1950s and has fortified its position as a technique of operational research in manufacturing (Hollocks, 2006). Modern production system DES applications have a myriad of use cases also outside the domain of operational research enabled by interfacing capabilities of commercial DES environments. Both historical and real-time data could be used in simulation models that guide decision-making, or even control a physical system real-time (Onggo et al. 2021). Real-time applications of DES models include integration in systems that optimize production schedules as Siatras et al. (2023) demonstrated, or even create supply chain plans as Park et al. (2021a) illustrated.

However, challenges are faced in integrating factory simulation solutions to processes of manufacturing organizations. Small and medium-sized enterprises are challenged by the lack of platform-independent interfacing capabilities in commercial simulation software and time-intensiveness of building models (Yu & Zheng 2021). Provider of a production simulation software could address these challenges by allowing interoperable reuse of engineering models describing the real-world manufacturing system, or perhaps allow interfacing a simulation environment with standardized application programming interfaces (API). Asset Administration Shell (AAS), an emerging technology from manufacturing sector, could be a means to achieve this.

AAS is defined as "standardized digital representation of an asset" (IEC63278-1, 2023). In addition to that, AAS offers manufacturer-neutral API access to manufacturing resources whose communication interfaces can be proprietary (Ye et al. 2021a). AAS specifications, virtual representation template models and a software for editing the models are developed by Industrial Digital Twin Association (IDTA), which involves over 110 manufacturers, software companies and research institutions.

AAS is described as "a concrete adaptation of the generic Digital Twin (DT) concept tailored to fit the needs of industrial production" (Jacoby et al. 2022). DT is a concept of virtually replicating a physical entity, involving both historical and real-time data, and communicating bidirectionally between the virtual and the real world (Onaji et al. 2022). The role of common data representation formats in DT:s is described by IDTA (2022) as follows: "The interoperable DT makes it possible to consolidate data from different data sources, creating an open ecosystem".

1.1 Context and motivation

The collaborative partner of this thesis develops software solutions for 3D production simulation with additional connectivity features that provide building blocks for production system DT applications. Competitive advantages of the software include the multitude of pre-built models of manufacturing resources, and customizability in modelling proprietary components. The main applications of the simulation software are design, planning, optimization and virtual commissioning of manufacturing systems.

Virtual commissioning refers to detailed automation and control engineering work made in advance before the real-world commissioning, which reduces the total lead time of commissioning a manufacturing system (Bangsow, 2012). Virtual commissioning utilizes simulation, multi-domain system modelling and bilateral data synchronization (Wang et al. 2023). There have been efforts to automatize the process of creating simulation models for virtual commissioning, but the lack of standardized data models hinders these efforts (Striffler & Voigt 2023).

From the perspective of a production simulation software, AAS is an interesting emerging technology for three reasons. First, it has an interrelation with the concept of DT. Second, a standardized virtual representation could allow using data in the virtual representations as simulation inputs or perhaps even transforming virtual representations to functioning simulation models. Thirdly, the standardized API of AAS could enable novel applications of production simulation with data flows over the boundaries of companies or information systems. From a broader standpoint, supporting open standards for information modelling in software systems could improve digital continuity and transparency across the value chains of manufacturing resources, reduce switching costs, and make software systems equally accessible for also smaller enterprises by eliminating the need to own bundles of complementary information systems that use only proprietary interfaces.

1.2 Aims and research questions

Fundamentally this research aims to identify, how and to which purposes industrial companies are leveraging AAS in the current state. After that, attractive overlapping application areas of AAS and factory simulation could be recognized, and features for interoperability with AAS could be specified and developed for the production simulation software. From technological perspective, this study illustrates a proof-of-concept application for interaction between AAS representations and production simulation environment to recognize unsatisfied requirements for an implementation, and to create knowledge on technological feasibility of such applications.

The understanding of industrial application areas for AAS is gathered as empirical research. The empirical research is conducted as open-ended interviews with representatives of manufacturing companies that participate in AAS standardization or utilize AAS in some application. More specifically, the interviewees are chosen so that they either participate in the development of AAS applications personally or use the factory simulation software in a company that drives AAS development in IDTA. Two internal interviewees from the collaborative partner are interviewed to gather aggregate perspective of demand for the technology in current markets.

The intermediate goal of recognizing overlapping application areas of AAS and production simulation is addressed with a research question, whose subquestions are set to be collectively exhaustive: Either AAS or production simulation already exist in an industrial use case to solve some problem, or neither of them exists in a hypothetical new use case. The research question guiding the empirical study could be formulated as follows:

RQ1: What problems in manufacturing does AAS address, and which of these problems could discrete-event factory simulation help in solving?

RQ1.a: What existing simulation use cases could be improved by increasing simulation environment interoperability?

RQ1.b: What existing AAS use cases could be improved by incorporating factory simulation in the application?

RQ1.c: What potential new use cases exist for AAS-interoperable factory simulation?

In addition to the application areas of AAS, it is necessary to understand the structure and building blocks of manufacturing resource AAS representations, so that the technological feasibility of leveraging interoperable virtual representations could be assessed. This understanding is gathered by analysing AAS modelling approaches from literature, analysing the currently available landscape of AAS modelling specifications, and by implementing a selection of prototypical use cases as a proof-of-concept for reusing data from AAS representations in simulation. This part of the study is guided by the following research question and sub-question:

RQ2: How are AAS virtual representations of manufacturing resources structured?

RQ2.a: What data stored in AAS virtual representations could be used as inputs for factory simulation models?

1.3 Outline of the thesis

After this section, section 2 presents a theoretical frame surrounding AAS by presenting recent developments in operational technology, by reviewing DT concepts in manufacturing to understand the theoretical connection between production simulation and AAS virtual representations, and finally by reviewing conventions of AAS modelling as well as application areas of the technology. After that, section 3 will present the empirical study of this thesis, i.e. interview study and technological experimentation conducted in parallel. Section 4 illustrates the findings from the empirical study, and finally, section 5 reflects on the results.

2 Manufacturing resource virtual representations in literature

Standardized virtual representations are a product of technological development giving birth to new problems. To understand the purposes of standardized manufacturing resource virtual representations, the contemporary technological phenomena that have shaped the technology of AAS are presented in this section as a literature review. This section begins with an overview of the ongoing fourth industrial revolution or Industry 4.0 (I4.0), renewed manufacturing paradigms, and standardization efforts by I4.0 realization initiatives. After that, the second subsection elaborates on the concept of DT as a theoretical background for AAS-interoperable simulation applications. The third subsection zooms in on the technology of AAS, providing a theoretical background on modelling virtual representations of manufacturing systems and analysing application areas of AAS reported in literature. The fourth and final subsection reviews gaps in DT and AAS research.

The material cited in this literature review consists of journal articles, conference papers, and technical specifications in section 2.3. The articles and conference papers were filtered according to the ranking of the publishing channel determined by Publication Forum of the Federation of Finnish Learned Societies, accepting only top-, leading- and basic level publishing channels evaluated by the forum. However, one exception is made to the selection criteria of the literature: a presentation by Schweichhart (2016) is referenced, as it is the original source of certain concepts that later became popular in I4.0 literature.

2.1 Overview of Industry 4.0 research field

Over the 2010s and 2020s, operational technology and ways of organizing production have undergone paradigm shifts. Traditional industries have adopted solutions that self-optimize or self-configure the manufacturing system, manage mass customization, allow accessing manufacturing resources over a network, and even exchange real-time data across a supply chain (Rahmanzadeh et al. 2023). This section provides a brief overview of I4.0 phenomena as a context for virtual representations and standardization.

I4.0 is characterized by convergence of physical and virtual worlds, mobility, and growth of technologies such as DT:s, artificial intelligence (AI), augmented and virtual reality, simulation, and big data analytics. (Anumbe et al. 2022). Wagire et al. (2020) divides the research field of I4.0 in five principal research areas, 10 major research themes, and 19 minor research themes. The five principal research areas are new business models, smart factory, real-time data management, standards and reference architecture, and I4.0 realisation strategies. (Wagire et al. 2020). The research field taxonomy by Wagire et al. (2020) is presented in entirety in Table 1. This section will present some of the listed research areas briefly, before moving in the research area of real-time data management in the next subsection to discuss simulation and its role in DT:s.

Table 1: Taxonomy of I4.0 research field (Wagire et al. 2020).

Principal research area	**Major research themes**	**Minor research themes**
New business model	Cloud manufacturing and services, IoT triggered business models	Mass personalisation, I4.0 enabling components, horizontal and vertical integration
Smart factory	Work organisation, advanced manufacturing systems	Machine-to-machine communication, human-robot interaction, 3D printing, CPS, AI, augmented and virtual reality
Real-time data management	Simulation and digital factory, big data analytics	Mobile applications, predictive maintenance, data-driven manufacturing
Standards and reference architecture	Interoperability standards, safety and cyber security	OPC UA communication architecture, information systems ontology, product-service architecture, wireless communication standards
I4.0 realisation strategies	Smart city and infrastructure, I4.0 ready workforce	Use of ICT, roadmap and maturity model, skills map, Lean practices, supply chain of I4.0

2.1.1 Mass customization and new manufacturing paradigms

Wagire et al. (2020) lists mass personalisation as a research theme relating to new business models. Mass personalisation, also known as mass customization, refers to offering individually customized products for large markets with only a small trade-off in cost or lead time (Åhlström & Westbrook 1999). Reported benefits of mass customization -capable manufacturing include increased market share, increased customer satisfaction and knowledge, reduced order response time, and reduced manufacturing unit cost (Åhlström & Westbrook 1999).

In addition to handling increased variety in offering, variation in form of market changes has created a need for adopting solutions that allow reconfiguring the manufacturing system rapidly (Renna & Ambrico 2015). Traditionally, dedicated manufacturing systems have been used to achieve high throughput with the downside of low equipment utilisation during downward fluctuations of demand, whereas multi-purpose flexible manufacturing systems such as computer-controlled machines have introduced high production costs (Bensmaine et al. 2014). Novel reconfigurable manufacturing systems address the production cost and utilization challenges by reconfigurable machines or tools that can be cost-effectively reconfigured to process different products (Bensmaine et al. 2014). The type of reconfigurability in manufacturing systems is categorized to ability, sequence and capacity reconfigurability (Trierwiler et al. 2020).

Another novel manufacturing concept are matrix manufacturing systems, i.e. individually plannable, standardized, product-independent process modules arranged in grid structure and connected with flexible material flows (van Erp et al. 2023a). Control of such manufacturing systems may involve resource virtual representations that represent processing capabilities of the resources as abstractions called "skills" (van Erp et al. 2023a). In comparison with dedicated and reconfigurable manufacturing systems, a matrix manufacturing system introduces the benefits of reduced throughput time and amount of work in progress, as well as improved throughput rate (Renna, 2024).

A further level of manufacturing system reconfigurability would be reached, if arbitrary off-the-self devices could be added to the system in a modular manner. Plug-and-produce (P&P) is an engineering concept that refers to a capability of commissioning new devices to the manufacturing process on the fly (Nguyen et al. 2024). P&P simplifies the required manual configuration efforts in situations where a device needs to be replaced (Ye et al. 2020).

2.1.2 Smart factory technologies

Smart factory is a manufacturing plant concept in which objects can negotiate with each other, technologies of I4.0 are integrated to improve quality, performance, transparency and controllability of processes, and information from physical and virtual sources is combined and utilized in manufacturing tasks (Mabkhot et al. 2018). Smart factories are instances of intelligent manufacturing, i.e. a manufacturing system with connectivity-based horizontal and vertical integration, self-optimization, and autonomous operations (Zhou et al. 2022).

A concept related to machine-to-machine negotiation is product-centric control. Product-centric control is a production method where the manufactured product is supplemented with information that the manufacturing process could use, resulting in simplified material handling, process control and product customization, as the product itself requests processes or transportation via its virtual counterpart (Onaji et al. 2022).

2.1.2.1 Cyber-physical systems

Cyber-physical system (CPS) is a term omnipresent in I4.0 literature and categorized under the theme of smart factory research by Wagire et al. (2020). CPS:s are defined as "physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core" (Rajkumar et al. 2010).

The history of CPS:s originates from the 1970s, when having computational resources in control loops of automation systems motivated developing real-time computation, i.e. deadline-based computational task scheduling techniques. The 1990s introduced interaction between the virtual and physical systems, as well as compact sensors and modelling phenomena of manufacturing plants using computational models. Finally, the term CPS was introduced in 2006 to describe these trends. (Kim & Kumar 2012.)

In manufacturing context, a CPS is called a cyber-physical production system (CPPS). A CPPS consists of human operators, manufacturing equipment, and interaction interfaces for accessing information generated by the manufacturing equipment or the human operators (Ribeiro & Björkman 2018). The purpose of CPPS:s is integrating of advanced analytic capabilities with OT of manufacturing systems, enabling decision-support or even autonomous control settings (Thiede et al. 2021).

Architecturally a CPS is nestable, i.e. it may consist of other CPS:s, and can therefore be hierarchically structured (Stock et al. 2019). A common method to model the complex architecture of CPS:s is a layered framework (Hu et al. 2023). Different three- and five-layer CPS frameworks are proposed in literature. Drath & Horch (2014) proposes three levels that are required for a CPS: physical objects, data models of the objects in cloud, and services based on the available data. Wagner et al. (2017) refers to the previous architecture, but renames the levels respectively as asset, data, and administration layers, and implies that an I4.0 Component is a CPS. A five-level CPS architecture proposed by Lee et al. (2015) in turn consists of smart connection, data-to-information conversion, cyber, cognition and configuration levels. The smart connection level provides unified access to sensors, the data-to-information conversion level executes analytics, the cyber level contains an information model of assets and stores data centrally, the cognition level comprises simulation and visualization aimed for decision-makers, and finally, the configuration level autonomously controls the physical objects according to the decisions set in the cognition level (Lee et al. 2015). The architectures by Wagner et al. (2017) and Lee et al. (2015) both comprise a layer with information model of the asset between the connection to physical world and administrative services.

2.1.2.2 Internet of Things

Connectivity and sensing of physical entities plays a crucial role in intelligent manufacturing. Internet of Things (IoT) is a term born decades before I4.0 in 1980, when radio frequency identification technology was first used to identify physical objects (Ben-Daya et al. 2019). In its modern form, IoT is described as a global network infrastructure that consists of connected devices, and is based on communication, sensing and information processing (Xu et al. 2014). Architecturally, IoT is composed of a sensing layer depicting the physical devices, a network layer, a service layer consisting of supporting web services, and an interface layer that acts as the software access for the IoT (Xu et al. 2014).

2.1.3 Interoperability, standards and reference architecture

The interconnected nature of intelligent manufacturing has introduced challenges with interoperability (Zeid et al. 2019). Interoperability is a major research theme in field of I4.0 literature, and research on topics related to standards and standardization has experienced growth since 2016 (Wagire et al. 2020).

The stage of achieved interoperability is described as levels of interoperability, which are defined as device, network, syntactic, semantic and platform interoperability (da Rocha et al. 2022). The levels of interoperability are visualized in Figure 1. Device interoperability refers to a capability of fundamental connection between devices, and network interoperability refers to different networks being capable to connect each other (da Rocha et al. 2022). Syntactic interoperability refers to unified data serialization formats, whereas semantic interoperability refers to describing the meaning of transferred data so that both the sender and the receiver understand it similarly (Zeid et al. 2019; da Rocha et al. 2022). Approaches for achieving semantic interoperability include ontologies, vocabulary definitions and linked data (da Rocha et al. 2022).

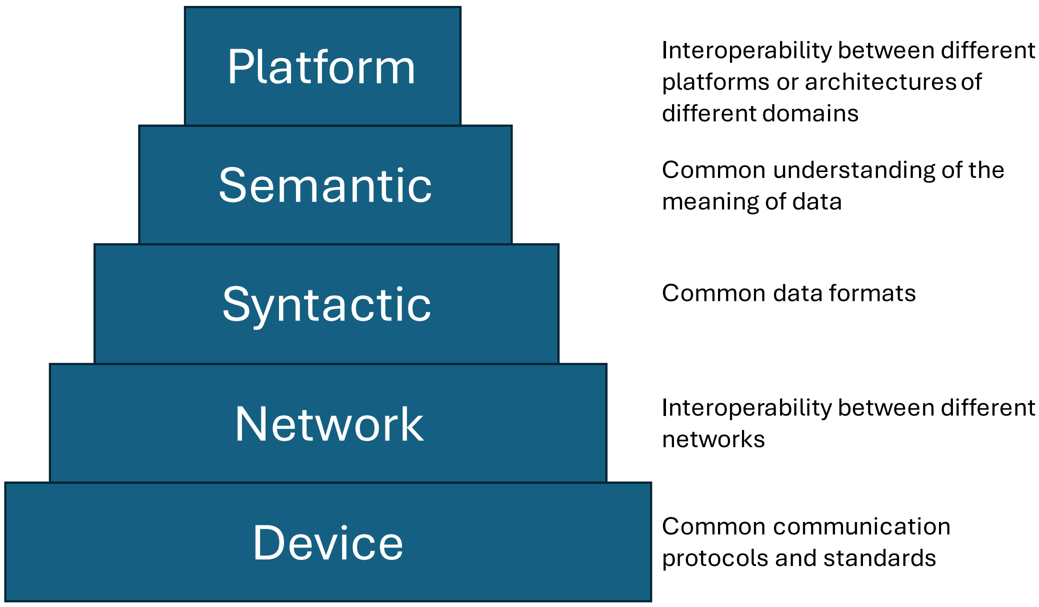


Figure 1: Levels of interoperability. Adapted from da Rocha et al. (2022).

Open Packaging Conventions Unified Architecture (OPC UA) is an established standard for interoperability in machine-to-machine communication and is named as one of the standards and reference architecture research themes by Wagire et al. (2020). OPC UA specifications have been published since 2006 (Bruckner et al. 2019). During its history, OPC UA has evolved from a data transfer standard to also a structured information modelling technology (Bruckner et al. 2019; Anumbe et al. 2022).

2.1.3.1 I4.0 Component

A vision for realizing interoperable data access to manufacturing resources is a concept called I4.0 Component. I4.0 Component was first introduced by Schweichhart (2016) in a slideshow presenting the goals of working group "Standardization and Reference Architecture" of initiative Plattform Industrie 4.0. The I4.0 Component is a concept of any technical asset of a factory being supplemented with a virtual representation, which can interact with other components using known digital interfaces (Fraile et al. 2019). A visualization of I4.0 component is shown in Figure 2.

The concept of I4.0 component is presented as layers. The first layer is the asset itself, and the second layer is the digital connection with the asset. Above the asset, a communication layer is responsible for facilitating communication. In the top, information, functional and business layers form an "administration shell" for accessing and manipulating the data associated with the asset. (Bauer & Mäkiö 2019.)

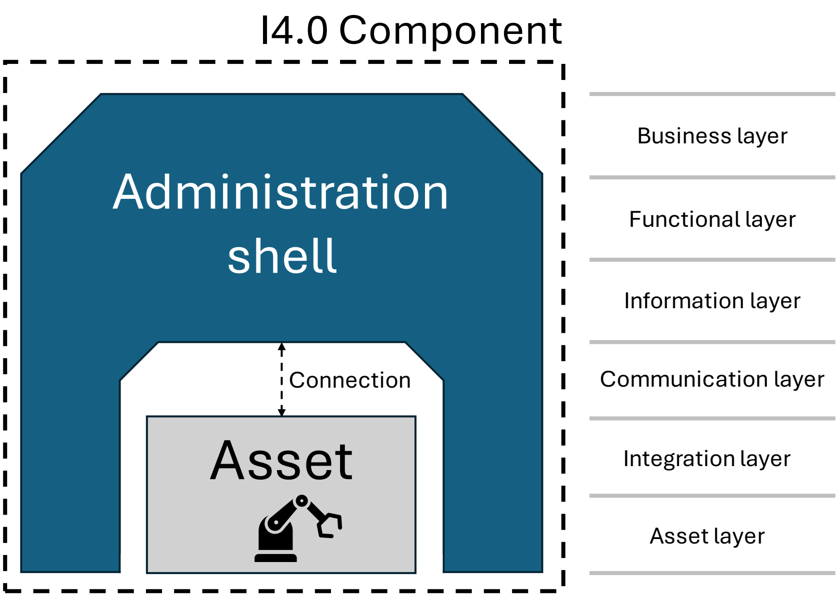


Figure 2: Elements of I4.0 Component. Combined from Fraile et al. (2019) and Bauer & Mäkiö (2019).

The information layer of I4.0 Component contains a data model describing the asset and provides "service atoms", i.e. simple operations for manipulating data items within the model. The functional layer is composed of both domain-agnostic platform services and application-specific services. Finally, the business layer adds business logic for the I4.0 component in form of software-based services, i.e. interactions with enterprise information systems. (Diedrich et al. 2017.)

2.1.3.2 Reference architectures

Reference architecture models are "blueprints" that describe a system physically, functionally and by allocating physical parts to functions (Moghaddam et al. 2018). Industrial reference architectures have undergone a paradigm shift during I4.0, as described next.

A traditional functional reference architecture, i.e. categorization of activities in a manufacturing organization, is ANSI/ISA-95 "automation pyramid". The automation pyramid defines five hierarchical levels of activities: production process, sensors and actuators, automation and control, manufacturing operations management, and business planning and logistics (Ismail et al. 2019). Information systems associated with the levels are Enterprise Resource Planning (ERP) software at the business planning and logistics level, Manufacturing Execution System (MES) at the manufacturing operations management level, and Supervisory Control and Data Acquisition (SCADA) systems at the automation and control level (Martinez et al. 2021). Lower-level information systems execute more real time -critical tasks than the upper-level systems (Ribeiro & Björkman 2018). The automation pyramid of information systems and activities is illustrated in Figure 3.

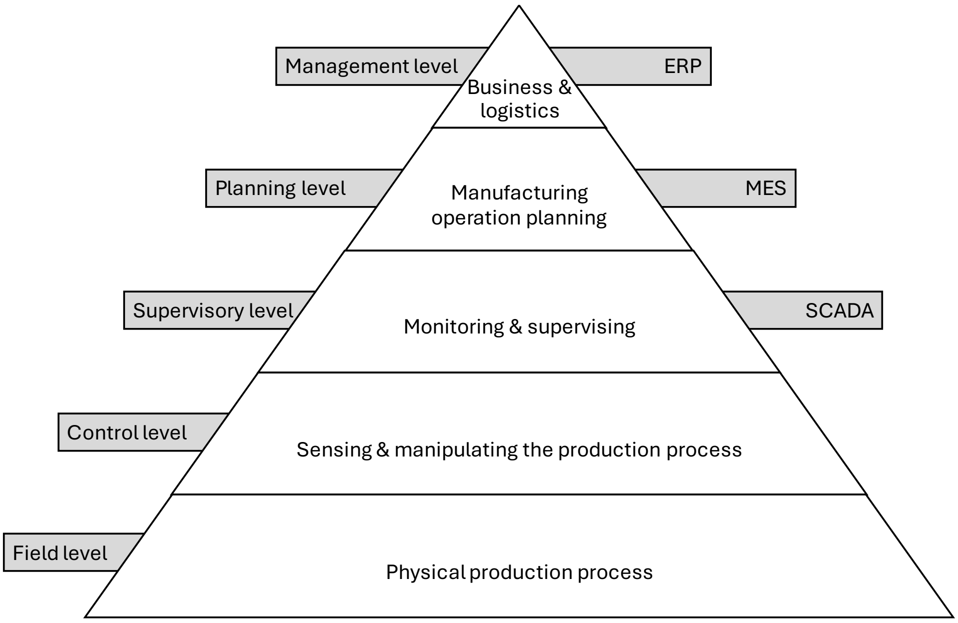


Figure 3: ANSI/ISA-95 activity hierarchy and related information systems (Martinez et al. 2021).

However, the architecture of modern smart manufacturing is characterized by CPS-based automation networks rather than a hierarchy of activities; higher-level information systems may be directly interconnected with the lowest-level systems to provide analytics and data management capabilities, enabling a decentralized mode of operation (Monostori et al. 2016). To describe the complex control and value networks of I4.0, a three-dimensional reference architecture model called Reference Architecture Model Industry 4.0 (RAMI4.0) has been designed (Bauer & Mäkiö 2019). RAMI4.0 was first proposed in the same presentation by Schweichhart (2016) as the concept of I4.0 Component.

The three dimensions of RAMI4.0 are the layers of the I4.0 Component, hierarchy levels of a manufacturing organization, and a time dimension depicting asset life cycle. (Bauer & Mäkiö 2019). The hierarchy levels dimension is based on the automation pyramid (Bauer & Mäkiö 2019). However, there are small differences: A product level is added to the bottom, the planning level is divided to work center and station levels, and a level entitled "Connected world" is added above the management level to describe enterprise-to-enterprise interactions (Fraile et al. 2019). RAMI4.0 is visualized in Figure 4.

The asset life cycle dimension of RAMI4.0 consists of two major phases. First, the asset exists as a type, i.e. a product that will become an asset is undergoing product development (Fraile et al. 2019). Then, the product materializes as instances, which are produced, sold, commissioned, operated, and finally decommissioned. (Fraile et al. 2019).

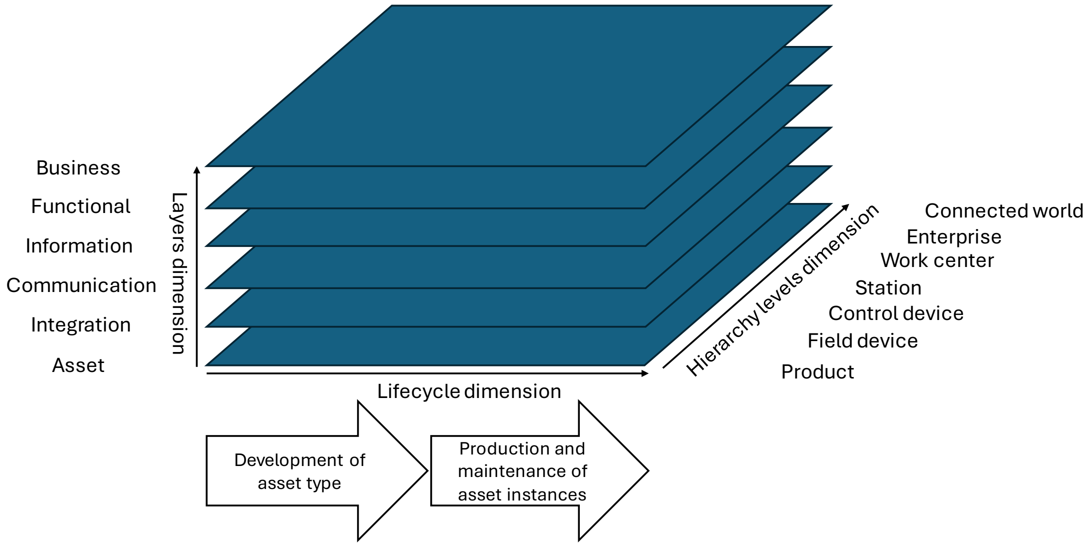


Figure 4: Dimensions of RAMI4.0 (Fraile et al. 2019).

2.2 Digital twins

The previous section presented phenomena of I4.0 following categories of I4.0 research themes. However, the review article by Wagire et al. (2020) did not position DT in any category of the research themes, but proposed a future direction of researching DT in the theme of real-time data management. This subsection focuses on literature related to the concept of DT, aiming to shed light on the interconnection between production simulation and standardized virtual representation in industrial DT:s.

The history of DT:s starts from 2003, when the first concept of a DT was introduced on a course concerning product life cycle management (PLM). In 2012, the National Aeronautics and Space Administration defined DT as "multiphysics, multiscale, probabilistic, ultrafidelity simulation that reflects, in a timely manner, the state of a corresponding twin based on the historical data, real-time sensor data, and physical model." (Tao et al. 2019.)

Applications for DT:s in manufacturing include designing and testing products, material selection, product customization, production planning and control optimization, layout planning, maintenance prediction and strategy development, real-time monitoring of operations, equipment remote diagnostics, energy consumption forecasting, analysing user behaviour, recovering waste for remanufacturing products, human-robot collaboration safety validation, and usage as collaboration tool (Singh et al. 2022).

The term DT is often used to describe virtual representation applications that have no data integration with the real world, which is why terms Digital Shadow (DS) and Digital Model (DM) are proposed to describe virtual representation applications that are less integrated with the real world (Kritzinger et al. 2018). A DS has unidirectional data transfer from physical to the virtual object, whereas DM:s have no data exchange (Kritzinger et al. 2018). A DS is purely descriptive, but more cost-efficient to implement than a DT, and applications often have a more holistic virtual representation than DT:s (Bhandal et al. 2022). A factory simulation application utilizing user-inputted data could be considered as a DM as it replicates a physical system, albeit in limited domains.

Literature in the research area of DT:s are categorized by Kritzinger et al. (2018) by four dimensions: publication type, level of integration, focused area in manufacturing, and DT-enabling technologies. The publication types are categorized into concept development and description papers, case studies, reviews, and definition papers. In the level of integration dimension, DS papers are the most common, DM papers the second most common, and DT papers the least common category. The focused areas in DT studies are production planning and control, layout planning, process design, product life cycle, maintenance, and manufacturing in general. Finally, enabling technologies referred to in papers are very heterogenous, but a majority can be categorized to simulation technologies, communication protocols, and I4.0 core technologies. (Kritzinger et al. 2018.)

2.2.1 Role of simulation technologies in digital twins

As one angle to understand the theoretical connection between an industrial DT and production simulation, the role of simulation technologies in DT:s is reviewed and discussed next.

Modelling technologies of DT:s are categorized to geometric, physics-based, data-driven, physics-informed machine learning, and systems modelling by Thelen et al. (2022), and presented in Figure 5 with their respective subcategories. Geometric modelling comprises solid modelling, i.e. creating 3D models of physical objects using computer-aided design (CAD), laser scanning the external shape of objects, and augmented reality (AR), i.e. overlaying contextual data on assets using light projection or a headset with transparent lenses. Physics modelling comprises solid body structural analysis, thermal and fluid flow analysis, kinematic and dynamic analyses of mechanisms, and multiphysics simulations, i.e. simulating coupled physical phenomena by exchanging data between models. Data-driven modelling is used when the modelled phenomena are complex, not fully understood or too computationally time-consuming, and it comprises statistical and machine learning (ML) modelling. Physics-informed ML modelling is used for more accurate models and faster model training. Finally, systems modelling refers to high-level modelling of interactions and structure of components in a system, and it comprises modelling languages and ontology modelling. (Thelen et al. 2022.)

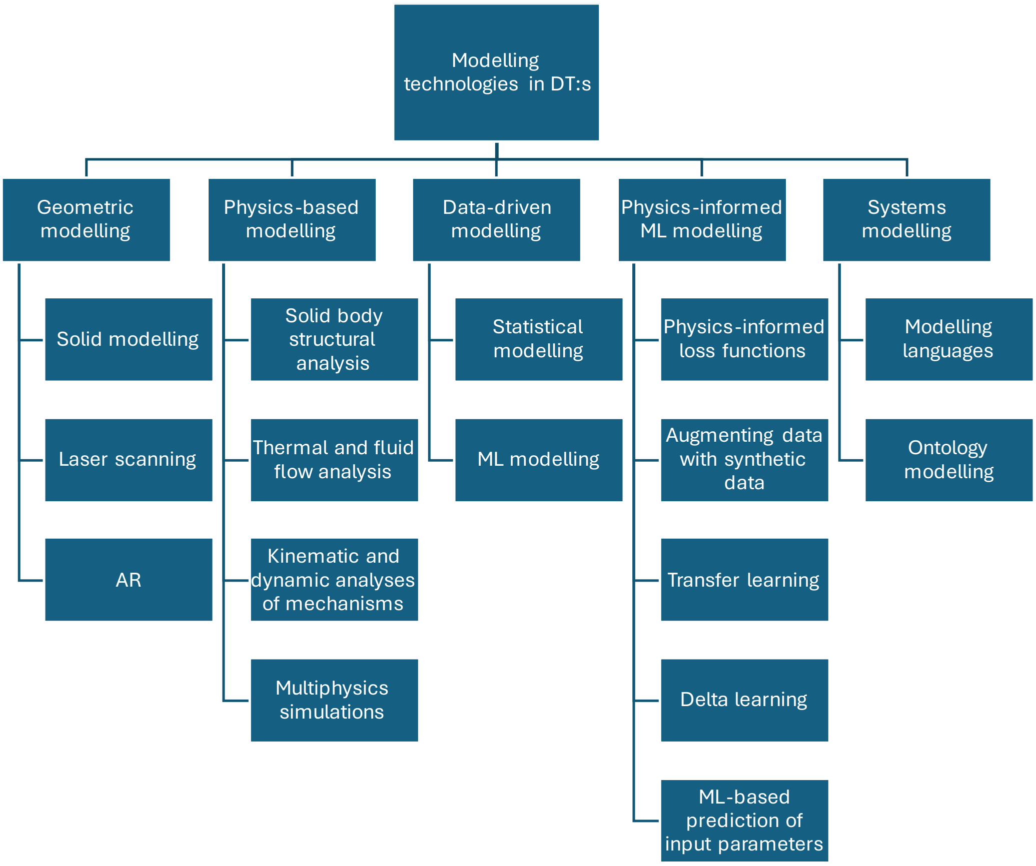


Figure 5: Categories of modelling technologies used in DT:s (Thelen et al. 2022).

The categorization by Thelen et al. (2022) names simulation-based technologies in physics domains as modelling technologies used in DT:s, but does not name factory or material flow simulation in any category or subcategory. Another review by van der Valk et al. (2020), presented and elaborated in Table 2, places emphasis on categorizing simulation-based DT applications rather than modelling technologies inside a DT. DT applications based on simulation modelling can be categorized by seven dimensions: the flow of simulation time, the presence of random inputs, time-static or time-dynamic modelling, model-building procedure, model scope, verifying and validating results, and simulation timespan (van der Valk et al. 2020).

By the categories proposed by van der Valk et al. (2020), the factory simulation software in the context of this study has discrete progress in time, dynamic model character and model scope of a system, with rest of the dimensions being model-specific. By the terms of Thelen et al. (2022), the factory simulation software utilizes kinematic and dynamic analyses from physics-based modelling technologies, and all listed geometric modelling technologies.

Table 2: Taxonomy of simulation-based DT applications. Adapted from van der Valk et al. (2020).

Dimension	Characteristics	
Progress in time	Continuous: The state of the simulation changes continuously	Discrete: Simulation time increments by a number
Probabilities	Deterministic: Does not use random inputs	Stochastic: Contains random inputs
Model character	Static: Represents a fixed time state	Dynamic: Represents behaviour over time
Model-building procedure	Yes: The simulation model is built following a "structured and target-oriented procedure or approach"	No: Such procedure or approach was not used
Model scope	Single entity: Represents one entity	System: Represents multiple entities
Verification and validation	Conducted: The model is both verified and validated	Not conducted: Either verification or validation is not conducted
Time horizon	Terminating: Finite simulation time	Perpetuity: The simulation does not terminate

2.2.2 Architecture of digital twins

Similarly to CPS:s, layered frameworks are used to model the architecture and elements of DT:s. Similarly to the concept of I4.0 Component, each additional layer introduces further abstraction of the physical entity. Layered DT architectures are presented in literature with four to six layers. Hu et al. (2023) lists the frameworks proposed by Zheng & Sivabalan (2020), Tao et al. (2018), and Redelinghuys et al. (2020). Frameworks by Onaji et al. (2022) and Jeong et al. (2022) were found additionally while conducting this literature review.

The DT architecture proposed by Zheng & Sivabalan (2020) introduces four layers. The first layer is a collection of assets. The second layer executes data extraction and consolidation using asset-specific communication protocols. The third layer, cyberspace, augments a DM with data generated using computational models. The fourth layer, interaction, consists of applications and devices that are used to access the DT. (Zheng & Sivabalan 2020.)

A five-layer DT framework proposed by Onaji et al. (2022) consists of integrated physical assets, product or process virtual models, intelligent, data, and enterprise layers. The virtual models layer contains physics, geometric, rule-based and simulation models, whereas the intelligent layer initiates optimization of the physical assets and creates evolved data for decision-making using AI. Finally, the data layer federates data received from the rest of the layers, including information systems of the enterprise layer. (Onaji et al. 2022.)

In a five-layer DT framework presented by Jeong et al. (2022), the layers are incremental evolutionary stages of a DT application. The first layer is an objectified representation of the asset. The second layer consists of synchronizing the virtual and the physical entities. The third layer consists of simulated predictions and analysis. The fourth layer federates the work of multiple digital twins of different types, and the fifth layer comprises intelligent life cycle services. (Jeong et al. 2022). This framework implies that the life cycle of a DT starts from a bare virtual representation, i.e. a DM, and that it evolves into a DS or DT as more features to enable higher level of integration are implemented.

A five-dimension DT model proposed by Tao et al. (2018) consists of physical entity, virtual equipment referring to a DM of an asset, services referring to self-optimizing, DT data referring to a synthesized information from different sources, and connections between each dimension and the DT data dimension. The dimensions are not in a particular order, even though the DT data can be considered as a topmost abstraction layer. Tao et al. (2019) names the activity of federating data from different sources to the DT data dimension as “data fusion”.

A six-layer DT architecture by Redelinghuys et al. (2020) has two layers for physical objects; one for devices whose communication is limited to signals, and one for control and data collection devices. Above them, local data repositories layer provides a vendor-neutral communication interface for inter-layer communications by an OPC UA server. The fourth layer, IoT gateway, supplements the data with information of the context, and the fifth layer, cloud information repositories, stores historical information on the system. Lastly, the sixth layer consists of simulations and emulations that access the data from the lower layers, and may also contain a 3D visualization of the system. (Redelinghuys et al. 2020.)

Commonalities can be noticed between layered DT frameworks proposed in literature, despite heterogenous definitions of upper layers. A comparison is presented in Table 3. First, most frameworks consider the physical entities as part of DT, as by the definition of Kritzinger et al. (2018), a DT requires the existence of the physical counterpart in real-time. Second, each reviewed framework includes a structured virtual representation or a DM. Third, connections are a pre-requisite for real-time integration, and therefore a crucial part of a DT. Fourth, analytics or optimization technologies based on simulation, AI or mathematical models, more generally computational models, are a recurring element in different DT frameworks. Lastly, most of the compared layered frameworks consider federating or fusing data from different sources, i.e. lower layers, simulation outputs and external information systems. These core elements of DT result in a generic five-layer DT framework.

Table 3: Common elements in layered DT frameworks.

	Zheng & Sivabalan (2020)	Onaji et al. (2022)	Jeong et al. (2022)	Tao et al. (2018)	Redelinghuys et al. (2020)
Physical entities	X	X		X	X
Connections	X	X		X	X
Structured virtual representation	X	X	X	X	X
Computational models	X	X	X	X	X
Data fusion or federation	X	X	X	X	X

2.2.3 Relationship between DT and phenomena of I4.0

The concept and elements of DT have similarities with other phenomena of I4.0, such as CPS, IoT and I4.0 Component. The interrelations between these concepts are discussed here briefly.

The relationship between CPS and DT is best described by a pattern of terms used in publication titles, as numerous publications are titled either "*Digital twin-driven … cyber-physical system*" or "*Digital twin-based … cyber-physical system*", hinting that the role of DT:s is serving as a basis and an enabling technology for a certain subset of CPS:s. In addition to that, both concepts involve a physical entity and elements in a virtual world.

As Diedrich et al. (2017) defined, an element in the functional layer of I4.0 component are domain-specific application services. Simulation technologies connected to assets on real-time could be therefore defined as domain-specific application services in the functional layer of I4.0 Component CPS.

DT, CPS and IoT have multiple definitions of their interrelation. According to Jacoby et al. (2022), some activities of IoT and DT overlap, but IoT is described to be below the layers of a DT, as IoT focuses on sensing and communication, whereas a DT organizes collected data in structured form for use in analytics. According to Ben-Daya et al. (2019), CPS is interrelated to IoT, as a CPS requires monitoring and communication capabilities that IoT can provide. According to definition by Latanzi et al. (2021), DT resides in the virtual world despite the parallel existence of a physical entity, CPS comprises both the physical and virtual counterpart, and IoT is a communication space for entities. In conclusion, a CPS can be based on the analytics capabilities of DT and the sensing network capabilities of IoT.

2.3 Standardized virtual representation of an asset

This subsection zooms in the information layer of the I4.0 Component, describing the essence, structure, current state of standardization, and competing technologies of AAS. First, the concept of AAS and its background are discussed. After that, modelling approaches for system-level virtual representations are reviewed. Then, competing or related technologies for AAS are presented in a brief overview. Lastly, the application areas of AAS described in literature are reviewed.

An asset is a real-world entity that is represented virtually in DT:s. The term does not only refer to physical entities, but also immaterial tangible digital entities serving as manufacturing resources, such as software, optimization agents, an algorithm, or a production plan. Lu & Xu (2018) defines the term "asset" as synonym for manufacturing resources and categorizes them in hard and soft resources. The term hard resource comprises machining resources, computational resources, and materials, whereas the soft resources include software, personnel, tangible knowledge, and intangible knowledge (Lu et al. 2014; Lu & Xu 2018).

2.3.1 Overview of AAS

On its essence, AAS is an asset information model that paves path for semantic interoperability. AAS service implementations could also be defined as data-structuring software frameworks for developing DT:s (Gil et al. 2024). An AAS information model is used over the life cycle of an asset and supplemented with information generated during the operational phase of an asset (Wagner et al. 2017).

The history of AAS started during the mid-2010s. The concept of administration shells was first presented in the same slideshow by Schweichhart (2016) that also presented the concepts of RAMI4.0 and I4.0 component for the first time. Later that year, Grangel-González et al. (2016) presented a predecessor for AAS using Resource Description Framework for creating the virtual representation in administration shells. The entire term "AAS" was first used in a journal article by Tantik & Anderl (2017), discussing the potential of AAS in context of service-oriented business models and mass customization. According to keyword co-occurrence analysis by Li et al. (2024), the co-occurrence of terms AAS and DT has emerged during the 2020s.

AAS service implementations are categorized to passive, reactive and proactive AAS:s, according to their interaction patterns. A passive AAS refers to exchange of virtual representations using a common file format and possibly a file server, a reactive AAS has additionally a representational state transfer (REST) API for uniform exchange of information, and a proactive AAS has the required infrastructure for autonomous peer-to-peer communication between AAS:s. (Ye et al. 2021b).

Multiple service implementations based on AAS exist by different authors. AASX Package Explorer by IDTA, SAP I4.0 AAS, NOVAAS by NOVA University Lisbon, PYI40AAS, and Eclipse Basyx are open-source implementations of AAS (Gil et al. 2024). Additionally, Fraunhofer Advanced Asset Administration Shell Tools (FA3ST) is an open-source reactive AAS implementation that has an additional service for asset connection using selected communication protocols (Jacoby et al. 2022).

In RAMI4.0, AAS could be positioned to the information and communication layers, covering the entire life cycle dimension. There are minor differences in views regarding the position of AAS in a layered architecture. Heppner et al. (2023) places AAS as the data layer of a three-layer CPS. Wagner et al. (2017) defines AAS to cover both the data and administration layers of the three-layer CPS model, and the entire asset life cycle. Bauer & Mäkiö (2019) however implies that an administration shell covers the administration layer entirely and the data layer only partly, as the communication layer of six-layer I4.0 component is outside the scope of administration shells. In the intersection of these three definitions is the information layer of I4.0 component, which contains an information model representing an asset and is therefore equivalent to a passive AAS. However, a reactive AAS supports a limited set of communication interfaces, implying that a reactive AAS could be placed on the RAMI4.0 communication layer. An AAS service implementation, such as Jacoby et al. (2022), can also contain elements from the RAMI4.0 functional layer.

2.3.2 Similar and competing technologies

AAS is not the first standard for DT or information modelling purposes. Several commercial and open standards exist for similar purposes. Publications by Jacoby & Usländer (2020), Miny et al. (2023) and Gil et al. (2024) survey these standards and technologies.

Jacoby & Usländer (2020) compares existing standards for DT and IoT applications. The compared standards include AAS, DT Definition Language by Microsoft, Next Generation Service Interfaces Linked Data API, Open Data Protocol, SensorThings API, and World Wide Web Consortium Web of Things (W3C WoT). Main differences between the standards are the existence of API specification or resource discovery mechanisms for networked assets. DT Definition Language and W3C WoT are the only compared standards that do not define unified API for resource access. In addition to that, DT Definition Language and AAS are the only compared standards that do not have a resource discovery mechanism, but it is noted that an implementation of such is under discussion for AAS. (Jacoby & Usländer 2020.)

Miny et al. (2023) compares a set of standards for asset information modelling. This comparison includes Automation Markup Language (AML), OPC UA Process Automation – Device Information Model, Digital Factory

Framework, Module Type Package and Field Device Integration as well as W3C WoT and AAS. AML is noted to be specialized in modelling production systems, whereas Field Device Integration is specialized in representing field devices and Digital Factory Framework in "production system assets". In this comparison, weaknesses of AAS and Digital Factory Framework include the lack of description metadata for asset properties and services, but it is noted about AAS that a submodel enabling these is under development. (Miny et al. 2023.)

Gil et al. (2024) reviews technologies comparable to AAS from a different perspective, defining AAS service implementations as software frameworks for DT development. DT software frameworks are categorized in six emergent clusters. First, structured data frameworks, including AAS implementations such as FA3ST and NOVAAS, define common data structures and facilitate bidirectional communication between the physical and virtual entities. As the second category, domain specific frameworks specialize on DT:s of a certain domain. Third, language specification frameworks define common data structures but no services or communication interfaces. As the fourth cluster, geospatial data frameworks offer tools for the areas of smart cities and weather modelling. Fifth, 3D-based and infrastructure-oriented frameworks offer a visualization engine, but not necessarily tools for DT communication. Finally, co-simulation and model-based DT frameworks focus on behavioural modelling of entities using one or multiple simulation models. (Gil et al. 2024.)

2.3.3 Structure of AAS virtual representations

Structurally, an AAS consists of two parts: a header and a body. The header contains identification information of the administration shell and the asset itself. The body contains "submodels", i.e. domain-specific information models that represent functionalities of the asset (Chilwant & Kulkarni 2019.)

The AAS submodels are composed of information objects called SubmodelElements. In total, 17 subtypes of SubmodelElement exist: Capability, Entity, SubmodelElementCollection, SubmodelElementList, DataElement, File, ReferenceElement, Range, Blob, Property, MultiLanguageProperty, Operation, OperationVariable, EventElement, BasicEventElement, RelationshipElement, and AnnotatedRelationshipElement (IDTA-01001-3-0, 2023).

The IDTA publishes AAS specifications in two forms: base specifications covering topics such as AAS REST API or AASX file format, and submodel specifications for published submodel templates. By the time of writing this thesis, the IDTA has published 26 templates for submodels, develops 41, has received proposal for developing 22, has one in review, and one on hold (IDTA, 2024a). The existing submodels serve heterogenous purposes, such as representing assets with hierarchical structure, manufacturer contact information, reliability of a resource, carbon footprint, or available communication interfaces of an asset. Some submodels relevant for simulative DT:s are presented next.

Three archetypal modelling conventions for hierarchically structured AAS:s are described with recommended use cases for each. First archetype is called full hierarchy, i.e. several levels of Entity elements hierarchically under one AAS submodel, and it is recommended for modelling assets with low dynamics and fixed hierarchy. Second archetype is called "one-down" hierarchy, i.e. modelling subordinate entities as separate AAS:s and referencing them using a ReferenceElement, and it is recommended for modelling assets with changing hierarchy and dynamic environment. Last archetype is called "one-up" hierarchy, i.e. referencing a higher-level AAS in AAS of a lower-level component, and it is recommended for describing the installation location of an asset. (IDTA-02011-1-0, 2023.)

An AAS submodel that contains relevant inputs for factory simulation is "Data Model for Asset Location" (IDTA-02045-1-0, 2024), as it represents location of an asset, required both for visualization and accurate modelling of intralogistics. Another submodel containing potential simulation inputs is "Reliability" (IDTA-2013-1-0, 2022), as it characterizes unplanned downtime of an asset with property "mean time between failures".

Relevant from the perspective of interoperable communication in DT applications, submodel "Asset Interface Description" (IDTA-02017-3-0, 2024) describes known communication interfaces of an asset, supporting interfaces based on communication protocols Message Queueing Telemetry Transport (MQTT), Hypertext Transfer Protocol (HTTP) and Modbus. The asset interface description is designed following the open standard W3C WoT Thing Description to allow reuse of existing representations (IDTA-02017-3-0, 2024). A complementary submodel "Asset Interfaces Mapping Configuration" (IDTA-02027-1-0, 2024) describes relations between asset interfaces and AAS representations, with one side as data source and one as destination.

AAS submodel "Provision of Simulation Models" specifies a standardized representation for simulation models, modelling entities as separate simulation models instead of a monolith system simulation model. According to the submodel specification IDTA-02005-1-0 (2022), simulation outputs, inputs and acausal connections are communicated via ports, i.e. interfaces of the model, which are accompanied by descriptions of the variables in each port. Heppner et al. (2023) illustrates a hybrid simulation framework utilizing this submodel and physics simulation models that follow a separate standard called the Functional Mock-Up Interface (FMI).

Functionalities of assets are intended to be represented using "Capability" submodel elements. This aspect of the standard is not published yet, and a submodel specifying the use of capabilities is under development. The working principle of capabilities in the upcoming submodel is described as follows: "Submodel to express capability definitions of a production resource, allowing to connect capability definitions and their properties from ontologies/catalogues to executable skills and their parameters" (IDTA, 2024a).

2.3.4 Virtual representations of system-level assets

AAS currently lacks the specifications for representing manufacturing activities in work center level. By the time of writing this thesis, the IDTA has received a proposal of creating and publishing AAS submodel "Value chain, material flow and process simulation", but has not yet assigned a working group to develop it. Another planned submodel to related purpose is "Bill of Process" (BoP), which is in development by the time of writing this thesis. According to its description, the submodel is "intended to provide all the necessary and basic information for a digital process description in the field of production" and will be built "based on process knowledge from various companies in the wire harness industry" (IDTA, 2024a).

However, four recent publications have illustrated AAS models depicting production lines, i.e. manufacturing systems. Therefore, it is purposeful to analyse how material flows of a manufacturing system are represented, as it is a key requirement for utilizing AAS information models in a factory simulation application. A summary of the compared system-level modelling approaches is presented in Table 4.

Quadrini et al. (2023) models the production flow of a simple manufacturing process in AAS by using four types of AAS: assembly line, resource, work order, and product. In this implementation, the assembly line AAS serves merely as a list of resources, and the resources store semantic identifiers of the operations they have capability to execute. The work order AAS represents the schedule and quantities of products that will be produced, and the sequence of the production steps is stored in the product AAS as references to the resources. (Quadrini et al. 2023.)

An alternative approach for representing a process by Park et al. (2021b) introduces five entity classes: plant, resource, plan, product, and process. In this implementation, the entity classes are modelled separately from AAS:s, but connected to physical world via product-, field device-, control device-, and work station -level AAS:s. The sequence of production steps is not stored in whole in any class or AAS, but material handling solutions access the information of where the product is transported next via an application or a fixed reference. In addition to that, each process stores a list of previous processes where input materials for the process must be fetched from. (Park et al. 2021b.)

Arm et al. (2021) demonstrates a plant virtual representation using five types of proactive AAS: MES, product, store, transport unit, and service unit. The store AAS:s depict both material and finished product stores with the same virtual representation, and the transportation unit AAS:s represent both autonomous transportation and human operators. Each AAS has a negotiation submodel that specifies the process of requesting and providing services between administration shells. The MES AAS stores the production plan and archived product data, and creates product AAS:s with required production operations. The product AAS sends material requests to the store AAS:s, moving requests to the transport unit AAS:s, and production requests to the service unit AAS:s. Finally, the store and the service unit AAS:s communicate with the transport unit AAS:s during the transportation, and the service unit AAS:s receive production parameter updates from the MES AAS. (Arm et al. 2021.)

Luxenburger et al. (2023) describes an AAS representation of a human-robot collaborative assembly factory. On the work center level, a layout AAS submodel specifies locations of work units in a 2D grid. Stations or work units consist of production and transportation units, whose physical properties as well as capabilities are described in AAS. On lower levels, control components, sensors and signals are represented with designated AAS submodels. Product types have both a Bill-of-Materials (BoM) representation of their structure and a representation of the required manufacturing process, i.e. the order of tasks, preconditions for starting the tasks, and semantic capability and operation descriptions. Production plan is not represented as AAS but originates from an external service. Finally, an AAS submodel called Topology presents the hierarchical structure of the production system similarly to the automation pyramid. (Luxenburger et al. 2023.)

It is noted that station-level processes have different levels of granularity, i.e. a station can be used execute a single action, a larger task or an entire sequence of tasks. Therefore, it is proposed that an AAS representation of process could be modelled hierarchically so that processes could be composed of smaller subprocesses. (Luxenburger et al. 2023.)

Table 4: Entity types in system-level virtual representations illustrated in literature.

	Arm et al. (2021)	**Luxenburger et al. (2023)**	**Park et al. (2021b)**	**Quadrini et al. (2023)**
Work center entity	-	Layout	Plant	Assembly line
Station entity	Service unit, transport unit	Production unit, transportation unit	Resource, process	Resource
Control device entity	-	Control component	-	-
Field device entity	-	Sensor, Signal	-	-
Product entity	Product	Product	Product	Product
Production plan entity	MES	-	Plan	Work order
Other entities	Store	Topology	-	-

The implementation by Arm et al. (2021) is only generalizable to a rather narrow field of manufacturing use cases due to its use of proactive AAS, machine-to-machine negotiation, and the CPS architecture that is distributed rather than hierarchical. On the other hand, the implementation by Park et al. (2021b) is not ideal for the purpose of this thesis either, as the implementation does not model the manufacturing system in AAS but “inherits” data from AAS:s into a third information model located between the AAS:s and a simulation model. Finally, neither of these two implementations model material flows of a work center in the AAS representation.

The implementations by Luxenburger et al. (2023) and Quadrini et al. (2023) are similar in the sense that a product AAS contains information of the sequence of process steps. Architecturally, this allows modelling flexible material flows when multiple product types with different sequences of production steps are manufactured in the represented work center.

An alternative approach for modelling system-level assets in AAS would be using the modelling conventions in AML. According to a recently published AML specification, AML models could be packaged in AAS:s to enable reuse of detailed engineering information later in the asset life cycle (AutomationML Consortium, 2024). According to an AML modelling convention, a functional model of a plant is composed of structural representations of the manufactured products, physical resources, and the processes representing the executed activities (Drath, 2021). This modelling convention is called “Product-Process-Resource concept” due to the combination of product-centric, process-centric and resource-centric views of a system in the modelling convention (Drath, 2021).

Proceeding one hierarchy level up, system-of-systems (SoS) level refers to the interaction of different production lines or different factories (Onaji et al. 2022). Therefore, a SoS-level AAS model would represent interactions on enterprise or connected world levels of RAMI4.0. Literature on SoS-level DT:s forms approximately 14 % of DT publications, and earliest publications date to 2017 (Onaji et al. 2022). However, literature on SoS-level AAS:s is scarce and lacks practical implementations. In di Orio et al. (2019), SoS-level AAS is stated to be a future research direction for NOVAAS, i.e. the open-source implementation of a reactive AAS. Anumbe et al. (2022) presents a concept of multi-factory AAS designed for both horizontal and vertical data exchange, i.e. sister factories or factories of a supply chain communicating with each other. In this concept, AAS representations of factories are hierarchically under a common AAS. (Anumbe et al. 2022).

2.3.5 Existing application areas of AAS

In this subsection, existing application areas of AAS are reviewed in a comprehensive literature review. This topic has already been reviewed by Abdel-Aty et al. (2022), finding 29 AAS applications in literature and listing all assets that were virtually represented in each use case. However, the field has evolved quickly, and 17 more applications were found in more recent literature. The assets modelled in each application are tabulated in the end of subsection, and the AAS literature reviewed both by Abdel-Aty et al. (2022) and this study is analysed to illustrate the focus of current AAS application areas.

AAS provides opportunities for business model innovation in five ways: it fosters novel value creation approaches based on life-cycle data of assets, it paves the way for adoption of AI to support decision-making, facilitates novel collaboration over company boundaries and distributed manufacturing, enables digital services within assets, such as product AAS negotiating its individual material flow path on a manufacturing system, and lastly creates a need for novel cyber security and intellectual property protection technologies. (van Erp et al. 2023b). Most of the recent literature reviewed in this study illustrates applications involving life cycle data or digital services within assets, specialized in domains of maintenance, life cycle management or managing P&P-capable production systems. The following subsections will present AAS applications in these domains.

2.3.5.1 Asset maintenance

Maintenance applications and especially condition monitoring services are a lucrative setting for AAS applications due to three factors: the uniform AAS REST API could provide access to component virtual representation over company boundaries, up-to-date maintenance documents could be stored digitally instead of paper, and as Jacoby et al. (2022) noted, the structured virtual representation can act as a basis for advanced analytics.

Maintenance strategies can be categorized to reactive and preventive maintenance. Reactive maintenance refers to policies in which maintenance is initiated by a stop of the manufacturing resource, whereas preventive maintenance is categorized to predetermined maintenance referring to scheduled maintenance actions, proactive maintenance referring to engineering improvements made to prevent failures, and predictive maintenance referring to detection of a degradation mechanism in early phase by measurements (Khazraei & Deuse 2011). Condition-based maintenance is a predictive maintenance tactic, whose implementation approaches include human inspection, continuous condition monitoring and autonomous prognostics. (Khazraei & Deuse 2011). Maintenance applications for AAS in literature could be categorized as condition-based maintenance.

Himmelstoß et al. (2023) presents an abstract design for a condition monitoring service provided by a component manufacturer. The service is designed for condition monitoring over company boundaries so that customer service of a component or machine vendor could have information of equipment condition before arriving to customer site, and workers of the plant operator could utilize the condition information for condition-based maintenance (Himmelstoß et al. 2023). According to IDTA (2024b), this design has been implemented by the collaborative partner of Himmelstoß et al. (2023), and the key benefit of AAS in the application is described as reduced integration costs of proprietary interfaces.

The condition monitoring case study by Kim et al. (2022) focuses on mapping condition data of a process plant into AAS format to achieve interoperability between equipment diagnostics and plant-level portal systems. Cavalieri & Gambadoro (2024) illustrates a condition monitoring DS of a water supply system, synchronized with sensor-collected data. Moreno et al. (2023) describes the implementation of a remote performance and maintenance monitoring DS where logs of completed production, operating hours since last maintenance, and cumulated number of machine strokes are exchanged over the internet in AAS format and processed for visualization in a client application. The data exchange is implemented within a data space, i.e. a secure data exchange ecosystem in which data remains in the source and is exchanged on-demand (Moreno et al. 2023).

2.3.5.2 Life cycle management

Product life cycle management (PLM) refers to managing the products of a company across their entire life cycle, motivated by reducing product-related costs and delivery lead times. The role of AAS in PLM is to provide a standardized interface through which product data can be accessed during all phases of the product life cycle. (Deuter & Imort 2021.)

Digital Product Passport is a concept of storing information on the structure, materials, spare parts, recycling procedure and repairability of a product. Plociennik et al. (2022) utilizes AAS to implement a similar concept with the exception that data can be added along the product lifecycle. A use case was implemented for automatized sorting of decommissioned electronics products according to their materials and value, based on three sensors detecting properties of the items and finding a matching product type AAS. However, it is noted that sensor-based matching of product types is feasible for only a small product variety, and data of individual product instances is not accessed. Therefore, other asset identification approaches such as radio frequency identification tags or barcode labels should be used in future applications. (Plociennik et al. 2022.)

Dickopf et al. (2023) illustrates a prototypical application of extracting product information from a PLM system in AAS format, using the AAS representation of products for 3D printing and providing product AAS representations via a customer portal. Volz et al. (2023) illustrates an implementation of exchanging carbon footprint calculations of products in a data space, utilizing a carbon footprint AAS submodel template published by IDTA. Rauh et al. (2022) describes an implementation of version controlling digital assets stored in products, using AAS for digital asset self-description when remote updates are made for the products. Jacoby et al. (2022) describes a quality traceability setting, where inspection results are updated to product DM from a production line DS using a reactive AAS infrastructure.

Fimmers et al. (2023) describes an implementation of tool life cycle monitoring using AAS. Quantities such as operating time, torque, and electrical current were transferred from tools with simple connectivity features and proprietary communication protocols. When a tool was placed on a rack, the data was updated to corresponding AAS representation that accumulates a time series of the measured quantities. The application was used to define threshold values for replacing the tools based on operational variables. (Fimmers et al. 2023.)

2.3.5.3 Plug-and-Produce

P&P and reconfigurable manufacturing systems are an application area where interoperable data exchange formats play a role of key technology. P&P can be achieved by fulfilling five requirements: a physical connection to the new resource, a mechanism for detecting the presence of a new device, communicating device information, assessment of functionalities and requirements of the new device, and applying network configuration to the new device (Monostori et al. 2016). AAS itself fulfils two of these requirements: it stores device information in interoperable format, and functionalities and requirements of a device can be represented using Capability elements as Luxenburger et al. (2023) demonstrated. On top of that, as Jacoby & Usländer (2020) noted, a resource discovery mechanism is possibly implemented for AAS in the future, after which three of the P&P requirements would be fulfilled.

Stock et al. (2021) demonstrates an AAS service implementation for device self-description in a 5G-networked CPPS. Nguyen et al. (2024) illustrates a configuration feasibility-checking DT of a robotized assembly cell, using AAS REST API as a bridge between a web production orchestrator application and an OPC UA server that connects a 3D visualization and the real-world assets with the AAS:s.

Benefits of P&P-driven reconfigurable manufacturing systems include feasibility of small batch sizes, dynamic re-allocation of resources, and flexibility in customizing products. (Anumbe et al. 2022). Therefore, AAS-enabled P&P applications could potentially generate increase in revenue due to better matching of production capacity and demand-side product mix. However, practical implementations are scarce so far.

2.3.5.4 Other miscellaneous applications

Siatras et al. (2023) illustrates an AAS use case in production schedule optimization using multiple scheduling agents and DES models. The term agent refers to “digital assets with advanced characteristics in autonomous decision making that are deployed within the environment”. The role of AAS in this application is to initialize and parametrize an agent capable of requested optimization method, and to act as interface when an I4.0 component triggers the need for rescheduling. (Siatras et al. 2023.)

Bavelos et al. (2022) illustrates an implementation of utilizing AAS data format in providing operators with task instructions on AR headsets. Assembly, inspection, process recovery and reconfiguration instructions were implemented, as well as a process status view. The data used in task instructions is stored in product, machine and robot manipulator AAS:s and sent to an AAS of the AR application. Also, a “production orchestrator” AAS is implemented for storing the status of the manufacturing system. Reported benefits include shortened assembly times. (Bavelos et al. 2022.)

2.3.5.5 Analysis of the application areas

As a summary of the reviewed applications, Table 5 summarizes AAS virtual representations by positioning them within the hierarchy level and lifecycle dimensions of RAMI4.0, including also the applications illustrated earlier in the context of system-level virtual representation. The abstraction layer dimension is simplified to a binary variable denoting the existence of the I4.0 component communication layer, as the applications either have or do not have connectivity to the physical asset.

Joining the sets of literature reviewed in this study and the literature reviewed in Abdel-Aty et al. (2022) results in a 46-row table with a list of assets represented in the application on each row. The assets could be intuitively categorized by their level in RAMI4.0 to describe which assets are virtually represented in AAS applications. Digital assets were excluded from this categorization, as categorizing them to hierarchy levels is not intuitive.

A complete table of the literature included in this analysis is presented in Appendix A. In addition to that, a bar chart representing the distribution of assets between different hierarchy levels of RAMI4.0 is presented in Figure 6, including a comparison of the literature in this subsection and the literature in Abdel-Aty et al. (2022).

Table 5: AAS applications in literature positioned in RAMI4.0.

Application	Assets	Asset connection?	Hierarchy levels of assets	Life cycle phases
Task instructions and support (Bavelos et al. 2022)	Product, machine, robot, orchestrator, AR application	Yes	Product, field device, station	Instance operation
DS of a water supply system (Cavalieri & Gambadoro 2024)	Fluid control equipment, facilities, communication systems	Yes	Field and control devices, station	Instance operation
Exchange of product data with a PLM system (Dickoph et al. 2023)	Products	No	Product	Entire life cycle
Tool lifecycle monitoring (Fimmers et al. 2023)	Connective tools	Yes	Field device	Instance operation
Condition monitoring as a service (Himmelstoß et al. 2023)	Components	Yes	Field device	Instance operation
Quality traceability (Jacoby et al. 2022)	Product instances, inspection station	Yes	Product, station	Instance operation
Process plant condition data exchange (Kim et al. 2022)	Components, equipment, process units	Yes	Field and control devices, station, work center	Instance operation
Performance and maintenance remote monitoring system (Moreno et al. 2023)	Machines	Yes	Station	Instance operation
Configuration feasibility checking DT (Nguyen et al. 2024)	Robots, conveyor, mobile robot	Yes	Field device, station	Instance operation
Digital lifecycle passport (Plociennik et al. 2023)	Electronic devices	No	Product	Entire life cycle
Production system representation for a DT (Park et al. 2021b, Luxenburger et al. 2023)	Production equipment, products, line	Yes	Product, field and control devices, station, work center	Instance operation
Production system representation in a DM (Quadrini et al. 2023)	Production equipment, products, line	No	Product, station, work center	Instance operation
Self-description in digital asset version control (Rauh et al. 2022)	Files inside a smart product	No	Digital asset	Instance operation
Production planning and control (Siatras et al. 2023)	Scheduling agent manager	Yes	Digital asset	Instance operation
Device self-description in 5G CPPS (Stock et al. 2021)	Production equipment	Yes	Field device	Instance operation
Carbon footprint information exchange (Volz et al. 2023)	Product	No	Product	Entire life cycle

As can be seen from Figure 6, the AAS virtual representations in literature are clustered to the lower end of RAMI4.0 hierarchy levels, with a majority of the papers illustrating product or field device AAS:s. Implementations modelling control devices are scarce, but stations such as machines or manufacturing cells are modelled in nearly thrice as many papers. The portion of publications modelling products is growing compared to the review by Abdel-Aty et al. (2022), and information modelling of work center -level assets has emerged as a fully new subset of AAS literature, however with very few AAS specifications to guide the implementation at this moment.

One possible explanation for the high number of papers modelling station-level assets would be the life cycle phase of an asset: when the AAS is modelled by a device or system vendor, it is in the product level, but will later be located on a higher level during its operational phase, i.e. after being commissioned into a production system as a manufacturing resource.

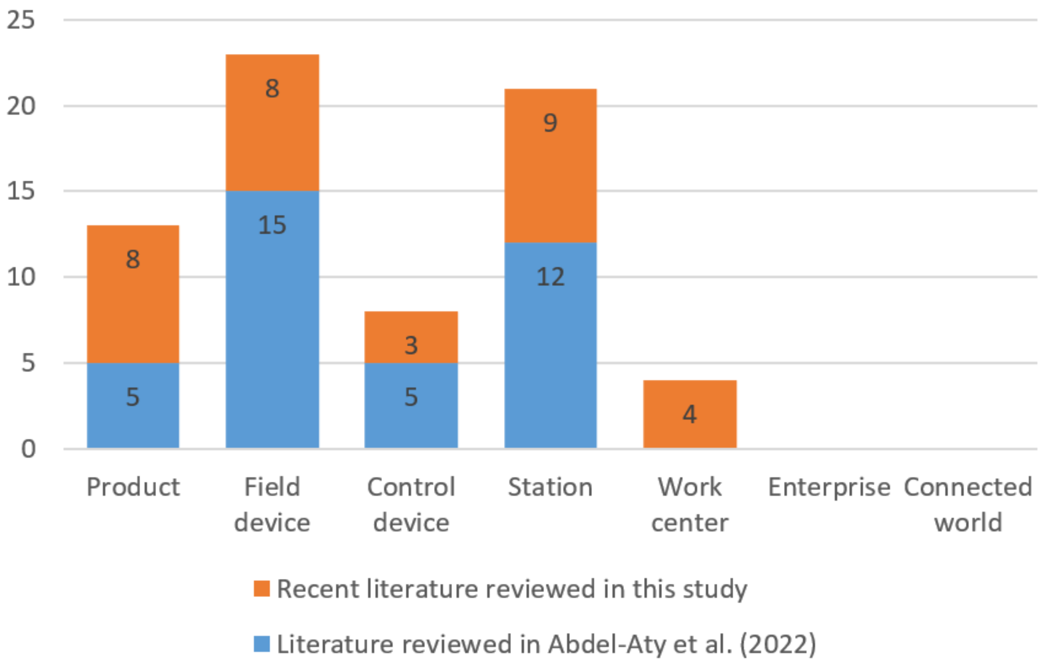


Figure 6: Number of publications implementing virtual representations of assets on different hierarchy levels, including literature reviewed both in this section and in Abdel-Aty et al. (2022). Data points of this chart are presented in Appendix A.

2.4 Conclusions on the reviewed literature

This literature review discussed literature from three distinct clusters; I4.0, DT:s, and AAS. The review of I4.0 literature illustrated novel manufacturing concepts such as matrix manufacturing and P&P, pointing out that these concepts make use of device virtual representations. The reviews of DT and AAS literature illustrated that a connection between AAS and DT concepts is visible, but also a gap between state-of-the-art DT concepts and practical implementations of AAS is noticeable.

AAS has interrelations with both DT and CPS concepts. First, AAS is related to DT, as all layered DT architectures in literature comprised a structured or objectified virtual representation of the physical entity, which a passive AAS can realize in a standardized manner for the entire asset life cycle. Second, AAS is originally based on realizing the concept of I4.0 Component, which is a subtype of nestable CPS. Third, AAS currently has application areas with emphasis on PLM, maintenance, and reconfigurable manufacturing system management, which are close to the original purpose of DT in domain of PLM that Tao et al. (2019) described.

A majority of the reviewed AAS applications are not fully integrated DT:s, but rather DS:s or DM:s. Conceptually AAS does not enable the entire concept of DT, as it is strictly focused in semantically interoperable structured information modelling, and it does not offer building blocks to analytics, data fusion or autonomous control functions present in state-of-the-art DT:s. Applications featuring connection to asset, such as Jacoby et al. (2022) and Nguyen et al. (2024), have the interactions with physical world implemented using established communication standards such as OPC UA or MQTT.

2.4.1 Research gaps

Fairly recent review articles discussing themes of DT and AAS were found to highlight research gaps on both literature clusters. From DT literature, Semeraro et al. (2021) names three areas with research challenges: improving sustainability-related applications of DT especially on value chain level, interoperability between DT:s to achieve data aggregation from a larger pool of data sources, and design of modular DT architectures to handle model complexity as well as allow flexibility and reusability of models in DT:s in other applications.

In AAS literature, Abdel-Aty et al. (2022) names three research gaps: quantitatively or qualitatively analysing the progress of both AAS and DT concepts, AAS-modelling for simulation purposes, and bidirectional data exchange between the AAS and the physical asset. The analysis of recent literature in this study also revealed research gaps in applications with enterprise- and connected world -level AAS virtual representations, as well as in application during the early phases of asset life cycle.

Based on the cited gaps in literature, the main novelties of this study are researching the potential of AAS in production simulation context, using qualitative research methods in research theme of AAS, and experimenting on automatic simulation model generation based on AAS virtual representations.

3 Research material and methods

The empirical study consists of two parts: qualitative interview study and technological experimentation. As was stated in section 1.2, RQ1 and its sub-questions will be answered by the interview study. RQ2 and its sub-questions are however more technical and require more detailed analysis of AAS modelling conventions. The landscape of existing AAS modelling specifications, relevant standard submodels, and implementations of system-level AAS models in literature were analysed in sections 2.3.3 and 2.3.4, but the illustrations in literature did not present the content of used AAS models in sufficient level of detail to deem this research question answered. Therefore, this research question will be answered by conducting technological experimentation, i.e. implementing a selection of prototypical use cases, collecting comments and feedback from the interviewees regarding the feasibility of these use cases, and by reporting recognized ambiguities in the used standard AAS submodels.

The process of the study is presented in Figure 7 as a Gantt chart. The empirical study was conducted after creating a draft of the literature review. The qualitative interview study was conducted in parallel with the technological experimentation. Last technological experiments were conducted shortly during September. Analysis of the interviews was started after conducting four interviews.

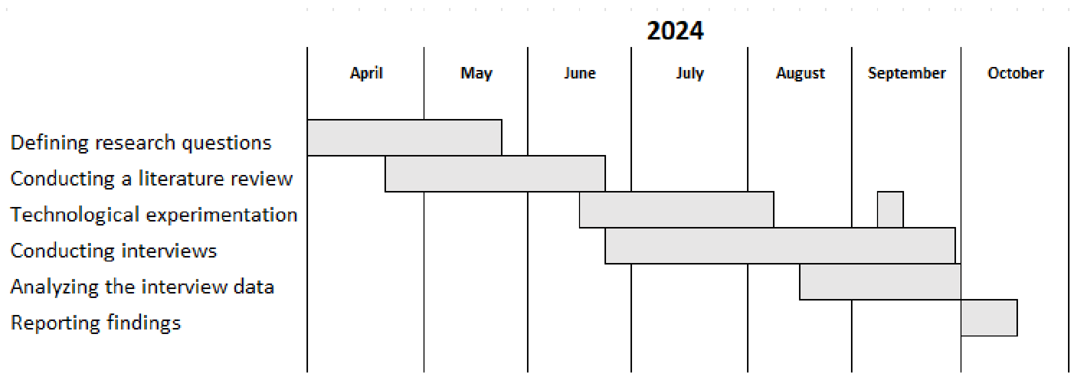


Figure 7: Process of the study as a Gantt chart.

3.1 Qualitative study design

Based on the comprehensive review in section 2.3.5, AAS is versatile, and therefore a possible implementation for an AAS-interoperable production simulation environment requires exhaustive fieldwork to define specific use cases for the technology before the technical requirements could be defined. Qualitative research methods are used for this explorative purpose.

3.1.1 Interview sampling

The interview sampling consists of three distinct groups of people. First, industrial experts with personal contributions to developing AAS specifications or applications were interviewed to access a designer's perspective. Second, factory simulation users from manufacturing companies that have existing AAS applications were interviewed to access a customer's perspective. Third, experts from the simulation software provider were interviewed to gain aggregate perspective of many use cases but with lower level of detail, and to learn about known customer needs that overlap with the means AAS provides.

The interview sampling was designed heterogenous, as finding a sufficient number of interviewees with knowledge on both AAS and factory simulation would have been difficult to impossible, considering the relatively small set of manufacturing companies that publicly express their investment on AAS, and the even smaller intersection with companies that were known to use the factory simulation software.

The industrial interviewees were found via four channels. First, simulation users from companies using AAS were found with help from sales managers of the collaborative partner, by showing a list of manufacturing companies that are members of IDTA. This channel resulted in contact information for six prospective interviewees. Second, authors of AAS-related publications or whitepapers with current position in an industrial company were searched from the internet, resulting in seven prospective interviewees. Third, the interviewees were asked to introduce potential additional interviewees, resulting in two prospective interviewees. Finally, one interviewee was found via collaboration networks of the collaborative partner.

In total, contact information was found for 16 prospective interviewees in eight distinct industrial companies and one applied research institute. Out of the 16 prospective industrial interviewees, two had switched companies recently, being currently in positions where they could not discuss industrial use cases for AAS or simulation. Five persons did not reply, and one prospective interviewee refused the interview.

Finally, the sample includes eight interviewees from five different industrial companies, plus two interviewees from the simulation solution provider. The total duration of the interview recordings is 10 hours and 46 minutes. Anonymized descriptions of the interviewees are presented in Table 6. In the column "Role of interviewee" of Table 6, the current individual role of the interviewee within their organization is shown. In column "Interviewee's background with AAS or simulation" the perspective the interviewee has for the research topic is described, including contribution to AAS research, specifications or applications and experience or familiarity with the Visual Components production simulation software.

Table 6: Anonymized descriptions of the interview participants.

Interviewee	Industry	Role of interviewee	Interviewee's background with AAS or simulation	Interview duration
AutomationCo	Automation equipment manufacturing	Research and development scientist	Contribution to AAS research and specifications	0:58:49
ElectricalCo1	Electrical equipment manufacturing	Programmable logic controller (PLC) engineer	Contribution to AAS specifications, user of the simulation software	1:01:18
ElectricalCo2	Electrical equipment manufacturing	Business development manager	Contribution to AAS applications, familiar with the simulation software	0:48:57
MachineryCo1	Machinery building and engineering	Automation system architect	Familiar with AAS, user of the simulation software	0:57:30
MachineryCo2	Machinery building and engineering	Product DT expert	Contribution to AAS specifications and applications	0:59:43
PumpCo	Fluid control equipment manufacturing	Production simulation specialist	Familiar with AAS, user of the simulation software	1:33:52
RobotCo1	Robot manufacturing	Research and development scientist	Contribution to AAS specifications, familiar with the simulation software	1:55:42
RobotCo2	Robot manufacturing	Robot simulation specialist	User of the simulation software	0:59:37
SimulationCo1	Simulation software	Sales manager	Working for the simulation software provider	0:53:35
SimulationCo2	Simulation software	Technical requirements manager	Working for the simulation software provider, responsible of monitoring emerging technologies	0:37:52

3.1.2 Conducting the interviews

The interviews were conducted as open-ended interviews due to the exploratory nature of finding prospective use cases for AAS-interoperable simulation. However, prospective thematic areas were planned to provide starting points and consistency for the discussions. The discussion themes and example interview questions from each theme are presented in Appendix B.

All interviews were conducted remotely in Microsoft Teams to simplify the recording and transcription process. Each interview was started by introducing the interviewer and the study briefly, reciting how the interview data is used, confirming consent for recording and data usage, and then asking the interviewee to introduce themselves for the record, even though their background with either AAS or factory simulation was already known to some extent from finding suitable interviewees. The open-ended interview protocol was rather dynamic and interactive for the remainder of the interviews, discussing themes common for all interviews, but asking further questions on certain themes according to the current interviewee's background.

Primarily, the interviewee was not directly asked to name intersecting use cases of production simulation and AAS, even though the goal of the study was communicated in the beginning. When existing simulation use cases were discussed, the interviewee was asked follow-up questions on whether some specific feature of AAS, such as REST API, semantic reference -based capability descriptions, centralized data over the life cycle, or asset identification, could solve problems or reduce costs on the context. On the contrary, when existing AAS use cases were discussed, follow-up questions aiming to fit simulation in the context were asked. Only after these discussions the interviewee was asked, if they could imagine some specific application for AAS-interoperable simulation.

When discussing the existing production simulation use cases, follow-up questions on topics such as simulation interfacing, data synchronization over the internet or reusing data to reduce modelling effort were asked to recognize potential application areas for AAS-interoperable simulation.

When discussing existing application areas of AAS, the interviewees were asked to describe the purposes of their AAS applications, future ideas for leveraging AAS, the motivation behind choosing AAS over other similar technologies, and challenges they have faced in using AAS.

In case a demonstrator of the prototypical use cases was shown during the interview, it was emphasized that the use case utilizes AAS from only one possible angle, and that the final application areas are open. The demonstrator was shown as late as possible after discussing the user's simulation use cases, so that the potential application areas found in the interview would not be limited to the application areas in the prototypical use cases.

3.1.3 Coding and analysing the interviews

The interviews were recorded and automatically transcribed. After each interview was conducted, the quotes in the transcription were parsed as rows in comma-separated file format using a Python script and imported in one Excel spreadsheet for analysis. Important quotes in the transcripts were marked with one-level answer theme coding while listening to the recordings.

After all interviews were conducted, the interview data was prepared for analysis by arranging quotes from each discussion theme in a two-dimensional matrix with code of the interviewee and theme of the quote as the dimensions. The resulting themes of quotes were grouped in 13 categories: Existing use case of AAS, benefit of AAS, challenge with AAS, weakness of AAS, criticism towards AAS, distinct role of AAS in information modelling, distinct role of AAS in communication, future expectations for AAS, use of AAS in automotive industry, co-use of REST API and simulation, PLM, potential use case in exchanging engineering data, and comments on a prototypical use case. The outline of the table used for analysing the interviews is presented in Appendix C.

The interview data was finally analysed by comparing quotes in same themes between interviewees, with the goal of recognizing emergent phenomena that multiple interviewees report independently of each other, or aspects in which the interviewees disagree with each other.

3.2 Technological experimentation

This part of the study aims to experiment on the technical feasibility of reading and writing AAS files by building a proof-of-concept application. The application demonstrates interoperability between AAS information model and Visual Components 4.9 factory simulation software. The goals set for the application are reading AAS data, using it in simulation, and writing simulation outputs to AAS format. The prototypical application is implemented using Python API of the simulation software.

The Visual Components factory simulation software is designed to construct simulation models in a layered manner, with the modelling layers divided in dedicated tabs within the user interface. The first tab, "Home", comprises functionalities for layout building and configuration using a catalogue of over 3000 pre-built device simulation models, "components". The second tab, "Process", contains editors to define product structures, process logic and material flow within the simulated factory layout. The third tab, "Modeling", is used for modelling arbitrary components with desired properties, behaviours, kinematic structure and geometry. The fourth tab, "Program", is used to define executable programs to simulate programmable manufacturing resources. The fifth tab "Drawing" can be used for exchanging factory CAD drawings. An additional tab "Connectivity", available in a software add-on, contains functionalities for synchronizing variables in the simulation models with real world variables using communication interfaces such as OPC UA.

Discussions with colleagues and reviewing related works in literature resulted in a decision that prototypical use cases could demonstrate automatic simulation model generation from AAS:s of different asset types. According to Jacoby et al. (2021), key requirements for hybrid DT:s combining different predictive approaches are interoperability with external systems, interoperability between models of a DT, extensibility, and reusability so that different models need not to be developed from scratch. The prototype aims to fulfil these requirements in a simplified setup of AAS information model and Visual Components, as well as reduce the effort and cost of duplicate modelling of building a simulation layer in a production line DT. Therefore, the prototypical application is not a DT application itself, but instead a set of supporting services for building hybrid DT applications using the simulation software.

3.2.1 Related works and definitions

In general terms, the built application comprises file deserialization and automatic model generation services and a mapping model between two information models: AAS representations and the underlying information structure in the simulation environment. Inspiration to the implementation and prototypical use cases is drawn from related works by Laemmle & Gust (2019), Sorvari (2021), Gollner et al. (2021), Isometsä (2018), and Platenius-Mohr et al. (2020).

In Laemmle & Gust (2019), automatic model generation was studied in the context of Visual Components simulation software, albeit using AML as source information model language. However, station components were not created from scratch based on behavioural modelling in AML virtual representation, but by finding an existing model with matching manufacturer and model name from component catalogue of the simulation software. In Sorvari (2021), a similar setup was implemented, but this time using Systems Modelling Language instead of AML. Yet again, station components were created by finding a match from existing components. In Gollner et al. (2021), mechanical and electrical dynamic simulation models of a conveyor system were automatically generated from AAS:s. This setup made use of AAS submodel "Provision of Simulation Models", the acausal connections between subsystems that were modelled in the AAS submodel, and existing FMI simulation blocks.

A common characteristic between each of the three model auto-generation studies is that simulation models of the subsystems, machines or components of the system were pre-provided. However, this study attempts to make use of an alternative model generation approach: matching a virtual representation to a known component class, generating a set of simulated behaviours associated with the class, and populating matching numeric properties with values from AAS. In Isometsä (2018), component classes and sets of simulated behaviours associated with each class were formulated for the simulation software for a standardized approach in component model creation.

For matching the properties between a simulation model and an AAS, mapping strategies defined in Platenius-Mohr et al. (2020) are used. In Platenius-Mohr et al. (2020), the proprietary information model of a commercial monitoring DS solution was mapped into an AAS representation. The authors propose four distinct strategies for mapping: preserving the source information model structure, minimizing or maximizing the number of created models, transforming according to name-based heuristic rules, or referencing to semantic dictionaries that define structures of different asset classes (Platenius-Mohr et al. 2020). Name-based heuristics strategy is chosen for this implementation, as integration to semantic dictionaries would have enlarged the scope of this prototype significantly.

3.2.2 Prototypical use cases and input data

Six distinct prototypical use cases, summarized in Table 7, are implemented to demonstrate using the AASX file format in simulation. In the first use case, simulation components such as machines and conveyors are generated from corresponding AAS representations. In the second use case, a production line layout is generated from a work center -level AAS using existing simulation components. In the third use case, a hierarchical AAS representing an assembly product is mapped into simulation inputs used in assembly simulation. In the fourth use case, a manual assembly process simulation model is generated from an AAS representation with product BoP. In the fifth use case, variables are bidirectionally connected between the AAS representation file and a corresponding simulation component using name-based heuristics mapping. In the final use case, the simulation software is used together with the FA3ST Service to exchange simulation outputs over the internet using the AAS REST API.

The AAS representations used as input data for the prototypical use cases represent manual workstations, a router machine, a conveyor, two different assembly products, a matrix assembly system, and a robot arm. The representations were crafted, as acquiring a real-life AAS dataset depicting the station- or system-level assets turned out difficult. AASX Package Explorer by IDTA was used as file editor for building the dataset, as the editor provides hints and file validation as well as AASX file serialization so that the resulting files are compliant to the AAS specification.

Submodels “Nameplate”, “Hierarchical structures enabling BoM”, “Technical data” and “Data Model for Asset Location” published by IDTA were used in the crafted input files. A summary of the used AAS submodels in each use case is shown in Table 7.

Table 7: Prototypical use cases in chronological order of implementation.

Prototypical use case	Input dataset	Submodels in AAS:s of entities
Simulation component generation from AAS representation	Conveyor AAS, machine AAS, manual workstation AAS	Unit: Nameplate, Technical Data, 3D CAD model
Simulation layout generation from work center -level AAS	Hierarchical production line AAS	System: BoM Unit: Nameplate, Technical Data, Asset Location
Generating assembly product logical structure	Product AAS	Products: BoM, Asset Location, 3D CAD model
Assembly process simulation model generation	Product AAS, matrix assembly system AAS	System: BoM Unit: Nameplate, Asset Location Products: BoM, Asset Location, BoP*
Connecting simulation variables with AAS file	Robot arm AAS, existing simulation model of the 6-axis robot arm	Unit: Operational Data*
Accessing simulation outputs via REST API	Robot arm AAS, existing simulation model of the 6-axis robot arm	Unit: Operational Data*

*: Not official submodel

For the input datasets of model generation use cases, CAD models depicting the items and units were acquired from training materials of the simulation software, and packaged in the corresponding AAS:s. However, real-life AAS samples representing field and control devices were used to test basic functionalities of the prototype, such as extracting the CAD model or reading properties from the AASX file.

In addition to the official submodels, a minimalist submodel for representing the BoP of products, shown in Figure 8, was developed from scratch to serve as a placeholder until the official submodel for the purpose is published. As was cited earlier in the topic of system-level AAS representations, the future BoP submodel specification is built considering processes in wire harness industry, i.e. electrical assemblies, which is why the placeholder submodel was developed to represent both assembly and non-assembly processes. Another custom submodel "Operational Data" was built for the last two use cases to represent position of a robot arm as numeric properties. A robot arm was chosen as the focal asset for this use case, as it could visualize multiple numeric properties as its joint angles.

Figure 8: BoM and custom BoP submodels of a product subassembly in AASX Package Explorer.

However, assumptions were made in the input dataset to simplify the prototype implementation. First, the dataset only contains references to AAS:s inside one AASX package file, in contrast to AAS:s that contain references to other AASX files within an AAS repository administrated by a company. Second, the asset coordinates in submodel Asset Location are assumed to point in the center of the bottom side of a bounding box fitted around the 3D object, and the origin of 3D CAD models is automatically moved to the same point.

3.2.3 Working principle of the proof-of-concept application

The most critical function for accessing data in the interoperable data format is a deserializing function, which reads the data for programmatic use following conventions specified in IDTA-01005-3-0 (2023). According to the specification, an AASX file contains an XML file that represents all AAS:s and submodels packaged in the file. The XML file is located in a specific folder inside the package, and the AASX package follows packing conventions of ZIP files.

The deserializing function simply extracts the only XML file from subfolder "aasx" inside the AASX archive to access the logical structure of the virtual representation. Other files, such as 3D models, are extracted from the AASX file by different functions only when needed. The extracted files are placed in a temporary folder, which is cleared in the end of program execution. The deserializing function is supported by several helper functions, built for navigating the underlying XML file, searching for submodels and submodel elements with a given identifier, and manipulating the XML tree.

When writing the simulation outputs back into the file format, a different serializing function is used. The serializing function works as follows: All files except the XML file are copied in a new ZIP archive with file extension "aasx", and the XML tree modified within the simulation environment is written to the ZIP archive. The writing was implemented using the file copying procedure, as attempting to write in one file inside an existing ZIP archive destroys file integrity.

Writing the AASX file is supported by helper functions, each dedicated to writing some AAS element type, such as "assetAdministrationShell", "submodel", "submodelElementCollection", and "property". Different element types have mandatory fields that are required by AASX Package Explorer to open the files, and creating these was implemented in these helper functions.

The first four prototypical use cases are built to read official submodels by finding specific properties from specific locations within the submodel. The last two use cases do not make any assumptions on the structure of the AAS representations, but only consider submodel elements of type "property".

For the simulation model generation use cases, a recursive algorithm to navigate the structure of system virtual representation using the BoM submodel was developed. The algorithm starts from given top-level AAS, iterates through submodels referenced by the AAS, tries to find matches for CAD, location, technical data and BoM submodels, jumps to subordinate AAS:s referenced in the BoM, and repeats the process until a BoM submodel is no longer found on the hierarchy level of the asset. On each level, the algorithm tries to find primarily a CAD model or secondarily the dimensions of the asset to initialize it as a 3D object. The design of the recursive algorithm was dictated by the structure of the underlying XML format of AASX files: submodels are not hierarchically under the AAS elements, but under a different section “submodels”, and referenced from the AAS element using a unique identifier. The top-level structure of the underlying XML format is illustrated in Figure 9.

```
▼<environment xmlns="https://admin-shell.io/aas/3/0">
  ▶<assetAdministrationShells>
   ...
   </assetAdministrationShells>
  ▶<submodels>
   ...
   </submodels>
   <conceptDescriptions/>
 </environment>
```

Figure 9: Top-level structure of the underlying XML tree in AASX files.

When generating device- or station-level models, the type of the source AAS is matched to a small number of categories using name-based heuristic rules. AAS:s recognized as conveyors are initialized by creating the start and the end of the conveyor path to the direction specified by AAS property “Length”. Other assets, such as machines, are set up as generic executable processes that process a product for the time specified in product AAS, unless the process time is specified in the station AAS. However, robots or mobile robots are not initialized with functionalities, as any public AAS submodel does not yet specify domain-specific necessary properties such as kinematics.

The last prototypical use case experiments with a reactive AAS and its REST API for communication. In the AAS REST API, requests are made to Uniform Resource Locator (URL) addresses of AAS submodel elements, which respond with information in JavaScript Object Notation (JSON) format (IDTA-01002-3-0, 2023).

The working principle of the URL addresses in AAS REST API is described in specification IDTA-01002-3-0 (2023) as follows: “Submodels define a hierarchical structure. Certain operations use an idShort-path to access deeper parts in the hierarchy ... Identifiers of Identifiables are base64url-encoded to be passed to the HTTP/REST API”. In other words, the URL address to access a specific part from an AAS representation can be derived from the tree-structured virtual representation with known operations. The URL address of an example API call and its structure are illustrated in Figure 10.

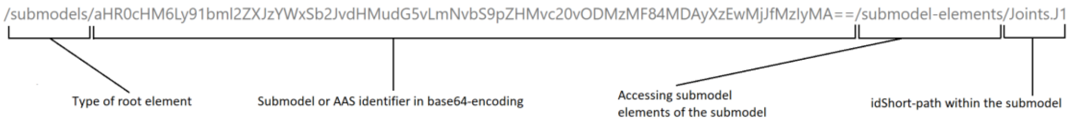


Figure 10: Structure of URL address in AAS REST API example call that accesses a joint angle of a robot.

4 Results

The potential use cases defined in the study are presented in three categories. Potential use cases discussed with interviewees are presented as the first category in the end of section 4.1. The implemented prototypical use cases are presented as the second category in section 4.2, accompanied with feedback and notes on technological feasibility of the use cases. Finally, conceptual use cases based on further synthesis of the interview data, combined with learnings from the technological experimentation, are formulated and presented in section 4.3.

4.1 Results of the interview study

This subsection illustrates the interview discussions in different topics. First, the industrial applications of AAS reported by interviewees are described briefly. Then, reported benefits and challenges in using AAS are discussed. After that, the distinct role of AAS within the ecosystem of industry standards is discussed. Next, the interviewees' expectations regarding the future of AAS are discussed. Finally, the discussion regarding potential use cases for AAS-interoperable production are illustrated and synthesized.

The current state of industrial applications for AAS is best summarized by a quote from interviewee ElectricalCo1; even though AAS currently supports only a limited scope of asset information modelling, there is a collective motivation among manufacturers to adopt the technology.

> *"It's still not so mature. It's still more young; the models are still coming, but it also has a strength: it's politically strong, so it got really good adoption."* -ElectricalCo1

4.1.1 Industrial applications of AAS reported by interviewees

This subsection describes the industrial use cases of AAS reported by interviewees with personal experience on working with applications for AAS. The interviewees reported in total eleven distinct applications. As some of the applications are currently under product development, the applications are described on an abstract level and without quotes as agreed with the interviewees. An abstract summary of the applications is presented in Table 8.

Seven of the reported applications could be categorized as applications with representation scope of products, with purpose of exchanging or managing product data. Only four applications of this category were currently in operational use, with one of them limited to providing information of a few selected products. The remaining three product information applications are either currently in development, developed as a one-time demonstrator, or only in internal use.

One of the reported applications involves a proactive AAS, however in experimental stage. The interviewee commented directly afterwards that potential industrial use cases for the application are currently unknown.

Table 8: Industrial AAS applications discussed in interviews.

General description of application	**Scope of virtual representations in application**	**Maturity of application**
Product information portal	Product	Launched
A software framework for building customized PLM portals	Product	Launched
Product information portal	Product	Launched for a small scope of products
Product information REST API	Product	In use with a limited group of users
Accessing AAS-format product information on a project collaboration platform	Product	Under development for an existing platform
Provision of product information	Product	One-time demonstrator
Consuming product information to automatize device configuration during system integration	Product	No information
Asset monitoring with connection to external PLM and IoT software	Unit	Experimental setting
Machine AAS with supplemental AML and FMI models	Unit	Experimental setting
Experimental setting of a proactive AAS	Unit	Experimental setting
P&P service for changing manufacturing system configuration	System	Launched

4.1.2 Benefits of interoperability in industrial use cases of AAS

When discussing existing use cases of AAS, the interviewees were asked to describe the experienced benefits of AAS in their application contexts. First, interviewee ElectricalCo1 reported reduced workload in exchanging product information, and additionally reduced workload in configuring purchased devices. Interviewee MachineryCo2 reported similar benefits of reduced workload, when data originating from various vendors is processed.

> *"If we can provide easily the data to the customer, the customer saves a lot of time. He saves time, but also internally I think we can save a lot of time because customers are not calling anymore 'I did not find it on your website, please send it', because everything would be there directly and interoperably."* -ElectricalCo1

> *"We could import data from third party devices much easier. So, for the PLC engineering, you need always the device description files for the network communication. If you can get them in a standardized way, you can also save a lot of time."* -ElectricalCo1

> *"If there are more than five different components, more than five vendors, you will need to have standards and specifications, otherwise you have a lot of manual work to do"* -MachineryCo2

Interviewee MachineryCo2 also reported that AAS has enabled fusing data from various sources into centralized representations, opening opportunities in service and support operations.

> *"For service and support you need insights and what were the life time activities of this product. And to get this information to the target systems we are using DT:s, because today you often need to connect different systems over and over again ... Harmonize the different data sources or the meaning of data. So, in the end you can connect different sensors or different customer information views from SAP, but for consuming systems it looks the same."* -MachineryCo2

Finally, interviewee RobotCo1 reported improvements in working with engineering information systems, as fewer information systems could be used to access the same information that previously required multiple systems due to proprietary information formats.

> *"So AAS has the capability for exchanging data in a standardized and centralized manner. And the more I want to do, the more I get trouble with all the engineering tools and applications outside these I have, because there I typically have workflows with tool chains and tools the customers use, and so a change of all these tools is a very big impact."* -RobotCo1

A common element among each of the reported benefits introduced by AAS is information exchange between companies and their respective information systems, which was enabled by syntactic and semantic interoperability within a common information format.

Relating to the benefits of interoperability, different interviewees described the role of automotive industry in their adoption of AAS, and the aspirations of automotive companies in leveraging AAS. Interviewees RobotCo1 and ElectricalCo2 reported that customers in automotive industry request product information in AAS format, namely fundamental technical properties and carbon footprint submodels representing the products. Additionally, interviewee ElectricalCo1 noted that not many automotive companies develop AAS inside the IDTA but concentrate their efforts for AAS development in project Digital Plant Modelling with Neutral Data Formats (DIAMOND), aiming to combine AAS and AML.

> *"So we discussed this topic because an automotive customer comes to us and just wants to have this kind of data for a robot ... So the use case of the customers is to have very fundamental technical data to plan future production lines; to have a view on which products [RobotCo] can provide and plan in a plant next year."* -RobotCo1

> *"Automotive customers are, how to say it, demanding it ... They [customer of the interviewee] already have received requests for the carbon footprint, and they requested it in the form of the AAS."* -ElectricalCo2

> *"Some automotive vendors are going for AAS. I think some others are in, maybe you heard about the research project DIAMOND. They're really going strong for AML. Also,*

they are trying to evaluate the combination of AAS and AML. But in IDTA I would say there's not so many automotive vendors present and strong pushing, there are more like regular machine builders active" -ElectricalCo1

4.1.3 Challenges in using AAS

To recognize potential pitfalls in creating an AAS application, challenges in AAS application contexts and weaknesses of the technology were discussed as one of the interview question themes.

Interviewees with personal experience on working with AAS reported issues in creating the AAS representations. Interviewees ElectricalCo1 and ElectricalCo2, from the same company, named the lack of suitable digital tools as a challenge. Interviewee AutomationCo highlighted manual effort and file editor compatibility issues as challenges when transforming AAS models from an older version of the specification to the current one. The challenge reported by interviewee MachineryCo2 was different, as they had faced difficulties in transforming product information of old products into AAS format due to pieces of relevant data missing.

"Problem is at the moment that there's often no 3D CAD tool that already exports nice AAS files which you could use for your work." -ElectricalCo1

"At the moment we are creating them by ourselves, there's no standardized platform in the background that creates them on demand" -ElectricalCo2

"The tooling could be better … There are some issues now and then that are coming up, this is for the tools in general that are used, and also some compatibility issues that can be there … There's recently been this version 3 of AAS that has come out, and the tooling and the submodel templates have to be updated then to make sure they're supporting version 3" -AutomationCo

"So we don't start today to maintain all the new products that have a twin. We also go back to the 1990s. Of course we don't have all data there." -MachineryCo2

As a theme related to the challenges in using AAS, interviewees PumpCo, SimulationCo1 and MachineryCo2 expressed concerns that the resource intensity of creating the virtual representations could hinder larger industrial adoption of AAS.

"What I didn't see is the time you spend on creating the structure, and I have a feeling it's not something you do in half an hour, and then all is good … If the AAS is difficult to use, if you need to be a trained specialist, no one's going to use it … If you need to dedicate people for training for maybe three months or so before they can use the technology, and then the people leave the company, or they don't have time to do that task" -PumpCo

"I would imagine that if we're talking about large manufacturing companies, they would have the resources, the time and the people to utilize this standard and also have

use for it, because they need to track and know what has happened. For smaller companies, I would imagine that there is no use case for these kinds of things, because it's not cost effective for them. But for larger organizations, there is always this kind of need for this kind of documentation and having more structured approaches to do things." -SimulationCo1

"If this concept should work, in the end, because you don't want to fill it by hand, and then the concept will sink down and you go back to the old processes." -MachineryCo2

As one of the topics, the interviewees were asked to name technical or architectural weaknesses of AAS. Answers in this theme were unique, as each interviewee named a different issue. Interviewee AutomationCo highlighted problems in semantic referencing of AAS. Interviewee ElectricalCo1 named the highly nestable structure of AAS representations as a problem for consuming applications, and also noted that a security specification for AAS is not present even though it has been under development. Finally, interviewee RobotCo1 highlighted deficient specifications on representing data as historic records in AAS and additionally criticized the capability-based representation of functionalities in AAS for duplicate modelling the modelling convention introduces with programmable assets.

"Enabling semantic interoperability is not an easy task. This is also not directly guaranteed by using AAS, not guaranteed even when you use semantic ID:s, that it's then going to be compatible with some other vendor's information" -AutomationCo

"Sometimes I think the model gets a bit complex, and it's difficult to search information in the AAS because of its structure. I think they're also considering having query languages or something on top of it, but at the moment it's very, very difficult because the submodel can be very nested, and if you are searching a certain property, you cannot just ask 'Is this property in there?', you have to go through everything and look very deep in there to get it out." -ElectricalCo1

"What I think is a bit critical at the moment is that the security part of the specification is not yet released, and it was postponed a couple of times, and I think security is essential for this kind of data exchange ... We are talking about having public data and data which is not for everyone, but only for special customers and special partners, so the security concept is very important and should be there from the beginning, but it's not." -ElectricalCo1

"In some submodels, it's just not as clear how you should display the information change over lifetime." -RobotCo1

"I program the program on a robot and then program in the AAS that this program is a skill ... It's just an abstraction." -RobotCo1

4.1.4 Distinct role of AAS within the ecosystem of standards

Most of the interviewees with experience on AAS were also experienced with other established standard technologies, namely AML and OPC UA. The distinct role of AAS compared to AML and OPC UA was discussed with these interviewees to gain understanding on several emergent questions: what data belongs in AAS representations, where do the boundaries between AAS, AML and OPC UA reside, and in which application areas does AAS have a strong positioning.

When discussing the role of AAS as a modelling language, interviewees AutomationCo and ElectricalCo1 elaborated AML as a more suitable source for detailed information models that represent system-level interactions. Interviewee ElectricalCo1 also noted that the modelling convention "Product-Process-Resource" of AML is established and in operational use at several manufacturing plants, in contrast to BoP and material flow submodels of AAS that are not even published yet. Interviewee RobotCo1 advised to follow the outcomes of the project DIAMOND, as the associations behind AAS and AML are making joint efforts in the project to create relevant specifications for representing complex automation equipment.

> *"For AML, the focus has been very much on the engineering phase, and here these standards can also align together. You can cover engineering aspects and also reuse the libraries that are provided by AML and also have detailed submodels. It's also mentioned in the specification of AAS that it allows detailed submodels to be used, and also AML is one source for having such submodels."* -AutomationCo

> *"For pure component importing, I would say AAS, but if you're really importing the material flow and the connection between the components, I think the AML model would be better"* -ElectricalCo1

> *"In AML there's also, they call it Product-Process-Resource concept ... It's already there in AML, and I know bigger automation and automotive companies who are already using this concept there, also in their factories" -ElectricalCo1*

> *"This project of AML and DIAMOND ... Have a look on which standardized engineering suite data formats are coming up there and how will they be provided ... Their plan is to define more templates which are able to describe things like robots or controllers with interfaces for plug connections and so on"* -RobotCo1

During the discussion on what data should be provided in AAS rather than in the complementary standards, unique but not contradicting definitions were received. According to the interviewees, data that is important for the entire life cycle, is relevant also outside the internal business processes, and that remains static should be maintained in the AAS format.

> *"So the discussion is always what data should be in AAS and what is only AML ... What's important across the life cycle, it should be in AAS"* -ElectricalCo1

> *"The order number, some tracking numbers, what's the stock in different plants, so this is something we don't give outside because this is only information that's relevant on the inside ... What's the standard delivery time of the product in this region, this is also some information we can provide, and also, if there's a white field in some AAS*

specification, we can provide it via this field and then it makes completely sense if other suppliers also maintain the value in the same field." -MachineryCo2

"When there is more detailed data for an instance and later on when I really need dynamic data from the shop floor, I think OPC UA provides much more meaningful mechanisms than the AAS … AML is only a model language … OPC UA is for data in motion because I've got time stamps … and I can provide the static data also in the OPC UA server, but it's just oversized for it." -RobotCo1

Additionally, interviewee RobotCo1 suggested that AAS should not improve device or network interoperability, and they elaborated that several established standards exist already for lower-level interoperability. This hints that the distinct role of AAS would be focused on syntactic and semantic interoperability within the levels of interoperability defined by da Rocha et al. (2022).

"The AAS must not solve my problem of machine interoperability on the shop floor. For this I've got Profinet, EtherCat, Ethernet, IP and OPC UA" -RobotCo1

4.1.5 Future expectations for AAS

As one of the discussion themes, the future expectations for interoperable AAS-based solutions was discussed. As the existing industrial use cases of AAS were mostly related to product data management, the tentative plans for leveraging AAS in the future were considered as one angle to recognize potential future application areas for AAS.

Interviewee RobotCo1 envisioned applications of AI enabled by machine-readable device information. The same interviewee also envisioned centralized management of all information related to product types, as well as information of product instances in a portal provided by a vendor.

"It's just a big chance because I think you could describe and provide each kind of data in a meaningful way, in a modelled way so that applications can consume them. Because I think a future problem will be the lack of workers which do everything, and this is just a chance to provide data which is better consumable for machine applications, program applications or maybe an AI." -RobotCo1

"From a business-to-business view, I have first the contact, the right people, then I make proposals, a cost estimation for several things, then I have an order, then later on I have a delivery date. All this information which flows between vendor and customer could be more centralized. And when the question comes 'Oh, when will the robot be delivered?', I look here inside and see the delivery date, and when will they quit up the provision of this product … So this is a great chance to centralize all data about an asset that is currently in a lot of hands and different sources." -RobotCo1

Interviewees MachineryCo1 and PumpCo expressed their expectations towards AAS solving problems in control of matrix manufacturing systems.

"There's many different product variants, and then you have to decide which way a product has to choose through a system. Think about matrix production, you have more or less no fixed connections. Then you need to orchestrate your complete system and every single order flow on runtime ... And I think this is the next step to provide some standardized approach using the AAS and using the standard communication protocols that are currently used in the industry." -MachineryCo1

"Depending on the tools that you give it, it can perform different actions, but the core structure is same for all work cells. Thinking about what you have in the AAS, it's actually a nice application of that. Whenever you have different products, you would also have different kinds of BoP ... All that [reconfiguration] happens dynamically with different products running in the same system ... Control-wise it's a complete nightmare. But thinking from the AAS perspective, that would actually be very useful, because you only have to develop one cell with all the different capabilities." -PumpCo

Interviewee ElectricalCo1 wished for more fundamental benefits by naming smoother data exchange across life cycle and value chain as their main future expectations for further evolutionary stages of AAS.

"I think at least we have this very smooth data exchange over all the engineering steps or over the whole supply chain. Then there's very little human work needed to put things together and to get the data from one thing to another thing, to another engineering tool and so on." -ElectricalCo1

4.1.6 Interrelation of simulation and REST API

The potential of utilizing reactive AAS interaction patterns with production simulation was discussed with the interviewees that have worked with simulation applications, but the discussion did not directly lead into any potential use cases.

Even though potential use cases for AAS REST API integration were not found in the interview discussions, the interviewees reported related experiences with REST API technology. Interviewee MachineryCo1 defined the REST API technology suitable for applications with low time criticality, such as in initializing simulation models. Additionally, the same interviewee described the benefits of REST API technology in their applications as reduced system complexity due to high level of abstraction in implementing the applications.

"If you have two-sided communication and a very high frequency of messages that need to be transferred, this is the wrong protocol. So HTTP and the REST stack is too slow just to provide big amount of accesses. But if you have single accesses with a centralized data server where you have to acquire for example a simulation model, so when I want to call the simulation model for my AAS, REST would be perfect because it's very easy to implement a REST call. It's slow, but who cares in the creation of a model." -MachineryCo1

"You can just do the REST call, you don't need to care about the detailed network architecture, TCP/IP is doing everything for you. You just need the id from the server, you

can even search for it if you have some kind of identification mechanism or central orchestration or some kind of blackboard architecture ... You can ask at this address how the actual syntax for the REST call is." -MachineryCo1

Interviewee SimulationCo2 stated that there are existing use cases of using REST API with the production simulation software, as integration with ERP systems has required this. Therefore, using similar technology for connecting to reactive AAS:s could potentially enable easy integration to asset information portals based on a reactive AAS.

"[When asked about REST API:s of ERP:s] Sometimes it's a proprietary protocol that we have to use, but it is common. We are aware that some people have already done this, where you have a SAP or ERP integration." -SimulationCo2

However, ideating combinatorial use cases for a reactive AAS and production simulation together was mostly a short-lived discussion theme, as interviewees other than MachineryCo1 and SimulationCo2 had no experiences of simulation applications leveraging this interfacing technology. As interviewee RobotCo2 said, these applications might exist in the future to increase connectedness of engineering work, but specific use cases are difficult to name currently.

"I guess it will be the future, so that we are more and more linked between, inside of a company as well as from outside ... I'm not sure about use cases." -RobotCo2

4.1.7 Potential use cases discussed during interviews

This subsection will describe the potential use cases that were discussed already during the interviews. Potential use cases based on further analysis of the interview data are presented later in section 4.3. All potential use cases found by the interviews are based on file exchange applications, in other words, a passive AAS interaction pattern. The potential use cases found by interviews could be categorized in three distinct application areas: interfacing production simulation with PLM systems, using production simulation as an early-phase engineering tool, and less surprisingly importing virtual representations of production units to simulation.

As was cited earlier, future AAS standardization will focus on product-level information modelling to avoid overlapping standardization with AML. This introduced a challenge for finding potential use cases, as this study began from the assumption that there would exist standardized virtual representations of unit-level assets such as machines on a larger scale.

4.1.7.1 Interfacing production simulation with PLM systems

One application area with potential use cases would be simulation integrated to PLM systems. As was found out earlier, PLM is one of the major application areas of AAS in literature. One PLM system provider already states to support AAS format for product information, and another major vendor of PLM systems has recently joined IDTA. Due to this, potential and purposes of interfacing production simulation with PLM systems was discussed with interviewees.

According to the internal interviewee from sales, interfacing production simulation with PLM systems has existing demand, but there is no knowledge of the specific use cases for this integration. The theme of PLM was further discussed with different interviewees with emphasis on use cases for such integration, and what data stored in PLM systems could be useful as simulation inputs. In Dickopf et al. (2023), the PLM system is stated to support standard AAS submodels Nameplate, Contact Information, Time Series Data, Technical Data and Carbon Footprint. Interviewee ElectricalCo1 suggested that also BoP models of products could be stored in a PLM environment and elaborated the motivation to do so.

> *"If you have the BoP, you know for the product which processes will be done, and maybe you know for each process, how much energy this process is using, and then you could calculate the carbon footprint of your product in production by using the BoP."* -ElectricalCo1

Interviewees SimulationCo1, SimulationCo2 and RobotCo2 reported use cases of interfacing PLM systems with production simulation that customers have inquired on, but that were ultimately not implemented. The discussed potential applications for interfacing PLM with the simulation software together were described as contexts where 3D models, correct revision of information, assembly structure, or technical properties of products are critical inputs for the production simulation model. Some of the reported PLM and production simulation interfacing concepts involve automatic simulation model generation for assembly or welding processes associated with a designed product.

> *"Companies that didn't become our customers because we didn't have any PLM connectivity. So one of their checkboxes is that 'We need to have PLM connection. Do you have it? No? Ok, then we can't buy it' ... But we're talking now about big manufacturing companies. Smaller companies don't have PLM software ... It could for sure open up some opportunities for us if we did have it, and it would make certain discussions easier especially with big companies."* -SimulationCo1

> *"The case could be this version history, and then maybe that they have their CADs locked away in PLM systems."* -SimulationCo1

> *"They [simulation users] need to make some type of design review of that part with a simulation, like moving stuff on a conveyor or having some new part that's assembled by robots. They would want to be able to have a way to import that changed CAD file and then simulate it. If it works great, comment it, put it back in the system that it's ready to the next stage in the life cycle."* -SimulationCo2

"They want to save time and money ... So they have their files which define how something is to be assembled ... Like put this part on, then put this part on ... And that data can then be used to automatically generate what you need in Visual Components, so you would save time in having to redo the whole thing." -SimulationCo2

"So you have thousands of different assemblies and they wanted to weld that, and the question was how to automate programming for the welding as well as how to be sure that everything can be welded because of collisions, reachability, cycle time, and so on ... All the information is inside the PLM system, so they have the CAD data, they have the knowledge about the parameters, and where the weld seams are and so on. Everything was described. But how to automate then the simulation out of it? ... If you have in mind that you have thousands of work pieces, more or less the same, but everything is different and you have to do it manually, it will be a work of years to do that." -RobotCo2

AAS provides a unified format for exchanging technical properties and CAD models of products. On the other hand, the production simulation software supports optional physics modelling and collision checking. These features have previously been used for feasibility checking of a manufacturing system, control code and robot programs. A potential integration with PLM systems would open new applications on product feasibility checking, similarly to the product designing and testing DT applications that Singh et al. (2022) listed among the purposes for DT in manufacturing.

The discussed use cases of integration with PLM systems should be further investigated with specific PLM systems to find out, which standard and custom AAS submodels are provided in PLM systems, and whether automatic generation of product feasibility checking simulation models would be possible from these submodels.

4.1.7.2 Production simulation as early-phase engineering tool

One category of potential use cases was found when discussing the challenges and resource intensity of creating AAS virtual representations for manufacturing resources. Some interviewees directly suggested specific simulation-to-AAS interaction features to applications in engineering system products.

As the first distinct use case in this application area, two interviewees described automatic AAS model generation for system-level assets. Interviewee PumpCo described this use case generally, but interviewee ElectricalCo1 described a specific context for this type of information exchange in writing control code during automation engineering. In practice, this use case would use very fundamental manufacturer and model metadata of the simulation components, and summarise the contents of a simulation model into an aggregate AAS representation with the BoM submodel.

"For a lot of people, if they are familiar with the tool, then that's what they are going to use. So they have the [simulation component library of the software] so they can quickly

build up a workcell or a full production line, and then they can write that from the simulation model to the AASX file." -PumpCo

"I'm always thinking I want to make use of the 3D simulation information in the PLC program ... So I think at least getting information from the simulation model. For example, how many motors I have, how many robots I have? This could simplify the work on the PLC side ... Also, if the component vendor of a conveyor has a function block to control their conveyor, then in the PLC program, I would like to know which conveyor is used in the line and then I can connect to the AAS of the component supplier, get the function block, and make easy engineering of the conveyor in my PLC program." -ElectricalCo1

Interviewee MachineryCo1 envisioned a different simulation-to-AAS use case for exchanging a smaller scope of data, but with more detail. In this use case, an AAS representation is not generated from simulation components or layout, but instead a set of parameters relevant for the production system would be updated to the AAS representation and later used in commissioning of the physical system.

"Think about designing some raw AAS:s for your system, then you set up the layout, import some basic information from these raw administration shells, and now you do the fine configuration of your system cycle times and conveyor times ... Now you export all the settings you can use for the real machines, and the real machines now get all their parameters directly from this digital planning environment, and then you have a very good start for the actual implementation ... You can't replace some manual last configuration of a machine, but when you start with a more or less good working points, that will spare so much time." -MachineryCo1

4.1.7.3 Importing unit virtual representations to simulation

Two interviewees suggested using AAS for automating simulation component generation, namely machines and robot arms. This use case is overlapping with one of the implemented prototypical use cases, but will however be reported as a result. Technical feasibility of a similar use case will be discussed in more detail later in subsections 4.2.1 and 4.2.2.

"Importing maybe a whole machine which the customer has made in whatever other 3D design tool, and then importing into the Visual Components ... You have the BoM of all the components, then maybe 3D data of all those components and then you can import it and get it here." -ElectricalCo1

"So at least if we have this information about also the behaviour of components, so the component modelling would be much easier, or especially if I see a robot model and I have described at least axis limits, and maybe speeds and values and so on, so I can create the robot model completely automatically ... I have only to push a button and then it should be imported. And so in theory, it should work, not only in [simulation software] it could be implemented in each CAD system or other applications, everybody who have access to this data could automate the creation of this information." -RobotCo2

4.2 Results of the technological experimentation

The technological experimentation on using data from AAS virtual representations in Visual Components production simulation software was conducted in parallel with the interview study. Out of the six experiments, five could be deemed successful with a varying level of generalizability, and one was not completed, but requirements for implementing it were recognized. This subsection will briefly illustrate the implementation results, discuss feedback given by interviewees, and describe pitfalls found in the implementation.

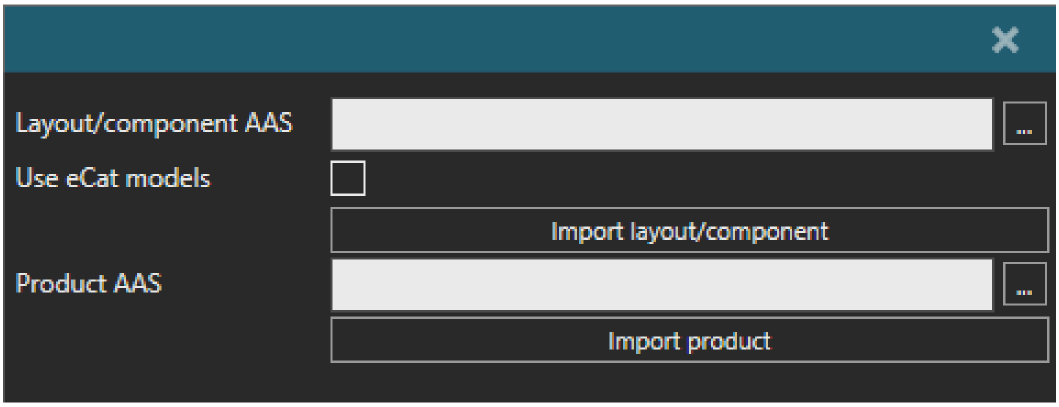


Figure 12: User interface of the proof-of-concept application.

4.2.1 Implemented prototypical use cases

Using AAS files to generate components, i.e. station, control, or field device - level simulation models, was implemented with a small set of component classes: machines, straight conveyors and manual workstations. The implementation reads the class of the represented asset, imports a CAD model packaged in the AAS representation, creates class-specific behaviours for the generated component, and reads numeric properties from technical data AAS submodel. Automatically generated component based on a machine AAS is illustrated in Figure 13. As was mentioned earlier, generating more advanced components such as robot arms or mobile robots was not demonstrated, as domain-specific technical information for these component classes is not represented in existing AAS submodel templates.

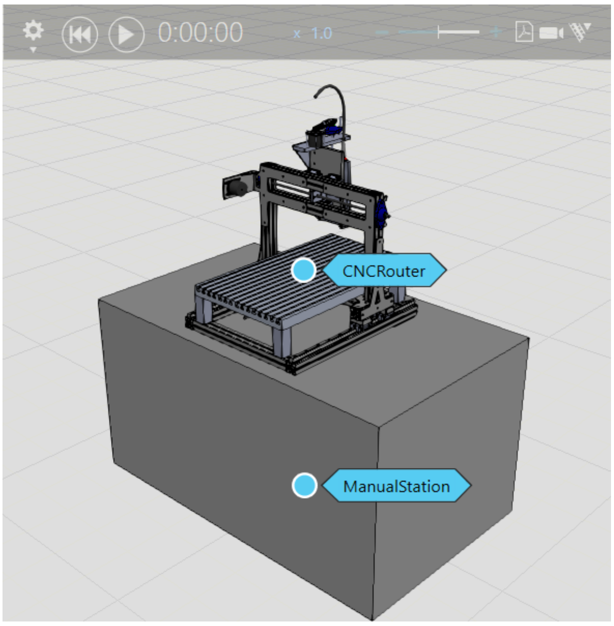


Figure 13: Generic machine and process automatically generated from an AAS with CAD submodel, based on component classification heuristic.

Implementing a more generalizable component auto-generating solution has a fundamental problem. The component generating solution would have to rely on a classification of manufacturing resources, such as a semantic dictionary, but the mapping between component class and simulated functionalities would need to pre-exist. Therefore, the component auto-generating could be implemented for only a limited set of known component classes. Feasibility of capability-based component generating solution should be reviewed, when AAS specifications regarding capability definitions are published.

As the second prototypical use case, simulation layout generation from work center -level AAS was implemented with an option; either existing unit models from simulation component library could be used, or the components would be generated from scratch using script of the first prototypical use case. Figure 14 illustrates a simple robot and conveyor layout generated by reading submodels Nameplate, Asset Location, and parameters with matching names in submodel Technical Data. However, the quality of this use case is limited due to a lack of standardized AAS submodels that would describe the interactions between units of a system, and lack of orientation information in submodel Asset Location.

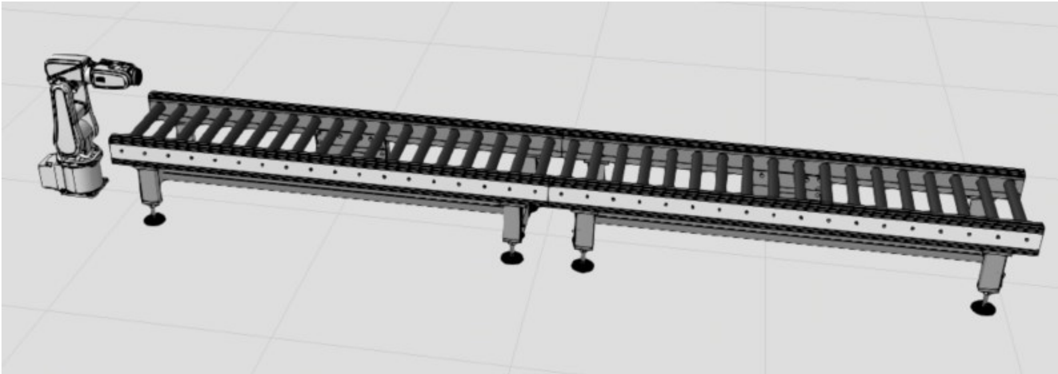


Figure 14: Simple simulation layout generated by matching digital nameplates with names of existing component models.

As the third prototypical use case, the assembly structure of products was generated based on BoM and asset location submodels in an AAS representation. This application could enable fast creation of a large product variety in simulation applications, as the product structures and CAD data could be reused from vendor-neutral virtual representations created in other systems. The source and target representations of a product are illustrated in Figure 15.

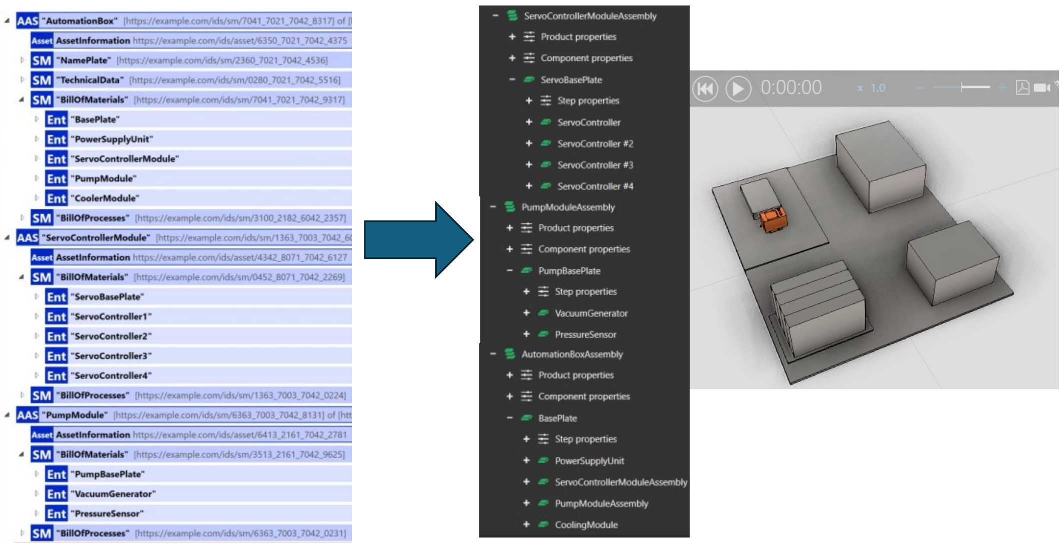


Figure 15: Modular assembly product structure in the simulation environment generated from a product AAS.

In the fourth prototypical use case, implementations of the previous two use cases were combined with additional logic to generate assembly processes and material flows for a complete system simulation model. The script for this use case was developed to enable importing different configurations of the product structure and process quickly for applications with high variety of manufactured products. Generating material flow in the use case relied on the custom AAS submodel illustrated in Figure 8. The generated process simulation model is illustrated in Figure 16, with material flow arrows of two different product configurations visible.

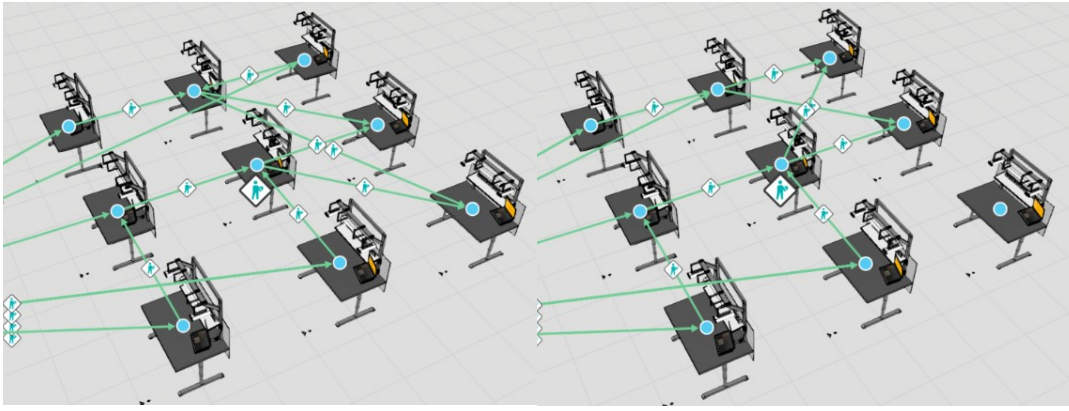


Figure 16: Automatically generated material flows between work cells of a matrix manufacturing system with two different simultaneous flow configurations.

As the fifth prototypical use case, a functionality focused on AAS properties was developed to demonstrate the reading and writing operations without making assumptions of the submodels within an AAS representation. In this prototypical use case, variables in an AAS representation and a selected simulation component can be connected using name-based heuristics mapping. After connecting the variables, the values could be refreshed on demand to either direction.

The use case, illustrated in Figure 17, included moving joints of a simulated robot to positions written in an AAS file. A similar setting could also be used for integrating a simulation model with a reconfigurable manufacturing system that has asset data integration implemented, similarly to what interviewee MachineryCo1 described when discussing potential use cases. Additional development of this functionality could involve creating a user interface for manual connecting of variables or defining property synonyms to connect variables with non-matching names.

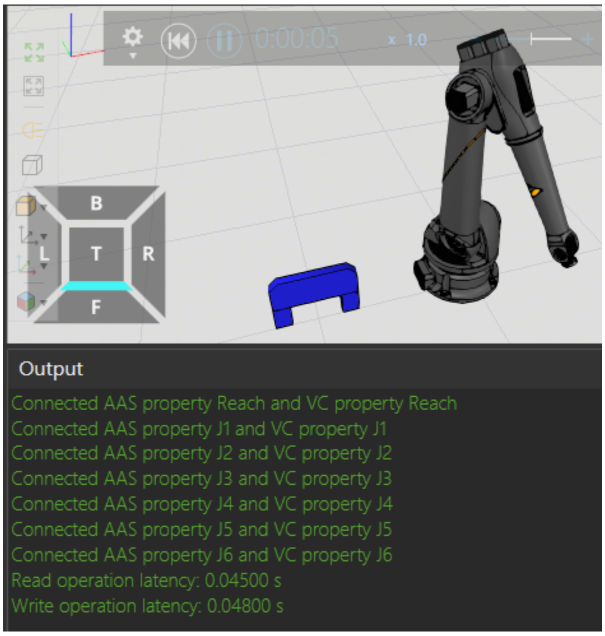


Figure 17: Bidirectional connection between operational variables of existing robot simulation model and the respective AAS representation in action.

The final prototypical use case involving AAS REST API was not implemented successfully. During this experiment, it became evident that reactive AAS implementations do not allow updating their environmental AAS representations with file writing operations, but the updates of simulation-to-AAS variables would have to be implemented using API calls over HTTPS, MQTT or OPC UA communication protocols. In other words, dynamic data could not be served over reactive AAS using only file writing operations between the AAS and simulation environment. Implementation of a sufficient setup was halted due to this finding and time limitations.

4.2.2 Feedback and notes from interviewees

The prototypical use cases were shown to interviewees who have background with the production simulation software to get feedback and to provoke more discussion on the potential use cases. The feedback and issues pointed out by the interviewees are discussed on this subsection.

The prototypical use case of generating simulation components from AAS representations was discussed the most. First, interviewee MachineryCo1 pointed out that modelling arbitrary components in the software is already easy by copying an existing component with similar capabilities and changing its 3D model.

> *"If there's a similar model, I copy it and I just delete the CAD file and use the functional base model to start the modelling"* -MachineryCo1

Interviewee ElectricalCo1 noted that modelling of kinematic behaviour in manufacturing assets cannot yet be automatized with AAS, and the fraction of modelling work with automation potential is therefore small.

> *"This lacks a little bit with the kinematics topic … Getting the 3D data is only not even halfway of what you have to do."* -ElectricalCo1

Interviewee PumpCo commented on the feasibility of this use case by pointing out that companies might not want to provide detailed engineering models over company boundaries, as doing so would disclose the internal mechanisms and control logic of products. To avoid this, they would rather provide FMI simulation models to serve the same purpose without disclosing secrets.

> *"If a company wants to get offers from a system integrators, I imagine that the company doesn't want to show everything of the logics, so if we could hide some of those features and just send the AASX file to the system integrator, that could be a way to go … I think we do that with the FMI. So the beauty is that you're hiding all the complexity inside that, so you cannot move the other way around and extract everything in there."* -PumpCo

One interviewee had coincidentally witnessed similar experiments before. The experiments had resulted in the conclusion that generating simulation models for complex assets from AAS would require too much additional AAS modelling, and therefore attaching simulation model files to the AAS packages was chosen as the approach for generating models with scope of a system. As this note described the company's internal research, the quote is presented without an interviewee code.

> "*When you have some kind of specific movement of an object, a complex [manufacturing resource], it would be very hard to describe it in the AAS in details so that it can be used in the simulation … We just decided that it's easier to attach the simulation model directly to the AAS."*

When discussing other prototypical use cases, interviewees gave fewer comments. When discussing the use case of generating an assembly process simulation model, interviewee ElectricalCo2 pointed out that reusing information from AAS representations in simulation is a concept that should be considered inside the IDTA organization, as simulation is not among the intended purposes of AAS information models.

> *"Having the possibility to include or to import the information from the AAS. So I think that is a really good concept, but I think this concept needs to be handled inside the IDTA … What you've shown there needs to be a standardized submodel for the AAS so that we can use it in the right way, and that the information that you need for the software will be always the same in a standardized way"* -ElectricalCo2

Interviewee PumpCo noticed that the generated simulation model uses constant processing times and noted that they require stochastic inputs in their simulation use cases. AAS specifications do not currently support probability distributions, even though ranges between minimum and maximum values and time series data is demonstrated in submodels.

> *"You had the processing time constant … Can you also do that stochastic? Can you add some kind of distribution? … If you have a strictly deterministic simulation, that will not represent the reality good enough"* -PumpCo

Interviewee MachineryCo1 commented on the use case of generating assembly product structures by describing the corresponding modelling task in a competing simulation environment. The competing simulation environment does not require predefined product structures, consequently not requiring automatic model generation of product structures either. However, they noted that predetermined product structures are useful in simulation of robotic pick-and-place systems, i.e. applications where position of subassemblies is critical.

> *"I find it much easier when you can just do it on runtime. So just say 'attach the following to objects' … In [competing simulation environment] there are two general objects, the products and containers. The difference is that in the container you can put something in, so that's the main difference … Here it's very useful to have this. You have to design it [product structure] once and then for example, when you have to use a robot for pick and place solution, you just have to say to the robot 'Here, and now use the following pattern' "* -MachineryCo1

In summary, the interviewees pointed out various practical issues in the prototypical use cases, mostly in the use case of generating simulation components. The use case has a low automatization potential due to the lack of kinematic modelling in AAS, benefits of automatizing it are small, and companies might not want to hand detailed information of asset functionalities for their customers or partners.

4.2.3 Ambiguities in AAS representations

During the creation of the input AAS dataset, it was learned that the current state of AAS submodel specifications regarding 3D coordinate systems limits the potential use cases of reusing AAS data in simulation models. The submodel used for coordinate representation, "Data Model for Asset Location", (IDTA-02045-1-0, 2024) is designed for intralogistics purposes rather than system 3D visualization use cases, thus specifying the static location of an asset two-dimensionally with X and Y coordinates, ignoring orientation and Z-coordinate of the asset location. The submodel however also presents the Z-coordinate, heading and speed of the asset as history records of the location, but the representation of static location remains ambiguous. This means that the submodel is not directly applicable for use cases which require accurate position information, for example reusing AAS representations of complex products from PLM systems in robotic assembly or welding simulations.

It was also noticed that representing numeric properties in AAS format has practical weaknesses from the perspective of machine-readability, as units can be presented in multiple ways. Properties representing real measures do not have units as mandatory inputs. If the property has a semantic identifier assigned, the value is automatically supplemented with a unit. In addition to that, publicly available AAS samples demonstrate various syntaxes for writing the value and unit as a string.

AAS submodels have not yet specified a way to represent probability distributions, which hinders the implementation of simulative DT applications with random inputs. Range, i.e. minimum and maximum of variable values, is represented in submodel template "Provision of Simulation Models", but it does not tell anything about the density function of values. A unified way of representing histogram data would be useful for AAS, not only for simulation applications, but also for measured values during the operational phase of manufacturing assets.

4.3 Potential use cases

Section 4.1 presented use cases discussed during the interviews, and section 4.2 presented implemented prototypical use cases inspired by earlier related works. In this subsection, further use cases are conceptualized, based on both further synthesis of the interview data and the learnings from technological experimentation. After that, a summary of all use cases contributed by this study will be presented in Table 9, and the results for the research questions are reflected upon.

Discussions with the interviewees on their AAS applications highlighted benefits in application contexts where information is exchanged over company boundaries, and challenges in the internal process of creating AAS representations. Therefore, economically viable applications for AAS should involve exchange of information over company boundaries to realize the benefits of AAS thus overpowering the effort of creating the models. Due to this, building AAS-based DT:s from scratch during the operational phase of asset instances would not make sense in industrial context, as there would be no other party to receive the created information. Therefore, potential applications for AAS and production simulation together would be consuming applications downstream in the asset value chain, or information-creating applications upstream in the value chain.

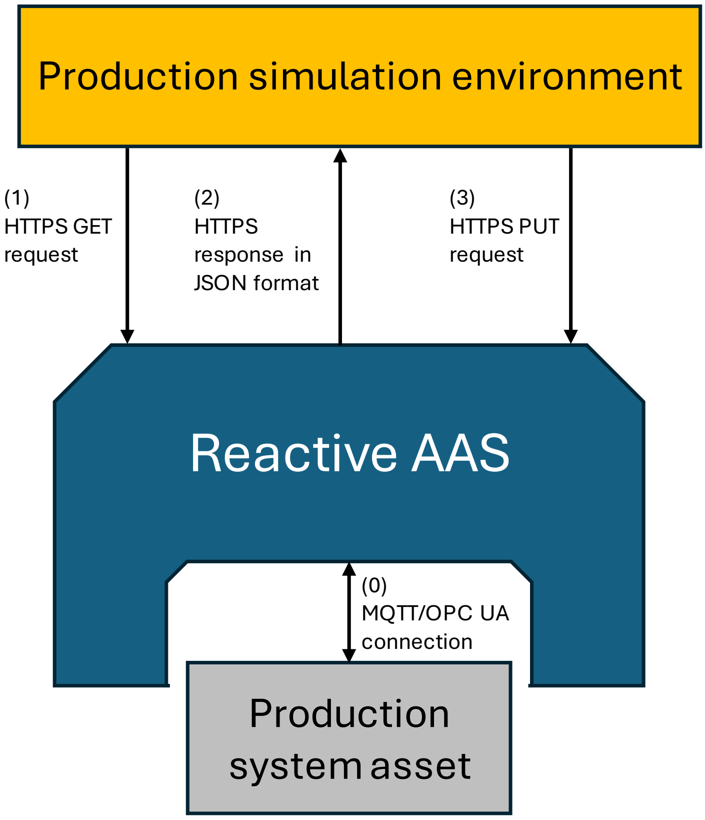
4.3.1 Conceptual applications for reactive AAS and simulation

As was learned during the interview study, the distinct roles of AAS are product-level information modelling and connected world -level communication. Some of the use cases described earlier belong to the area of product-level information modelling. However, direct use cases for connected world -level communication based on REST API were not found despite discussing the theme with interviewees. To conceptualize the potential applications of simulation applications communicating via standardized REST API, definitions for the applications will be proposed next.

First, the theoretical application areas are categorized. An intuitive dimension for categorizing the use cases for file-based information exchange was the direction of interactions between simulation and the AAS representation, i.e. reading or writing AAS virtual representation. With a reactive AAS, this dimension would now describe whether the simulation application is on the client or the server side of a web service, i.e. does it consume the exchanged data or serve it.

Second, descriptions of suitable contexts for the applications are derived based on the interview data. According to interviewee MachineryCo1, REST API is suitable for communication that is not time-critical, and its main benefit is easy data integration to other web applications. Therefore, the data exchanged via REST API should not be required during simulation runs, but before or after them.

To implement a reactive AAS interaction pattern with a production simulation software as one communication participant, a solution with low implementation cost would involve using an existing open-source reactive AAS implementation, such as FA3ST or NOVAAS. Based on the learnings gathered during the technological experimentation, communicating with a separate reactive AAS from the simulation software requires a connection, as file writing operations are not sufficient for updating simulation-to-AAS variables in a server-side simulation application. The data flows between simulation and AAS representation could be satisfied in both client- and server-side simulation applications by fulfilling two requirements: capability to make HTTPS requests from inside the simulation environment, and capability to read values from partial AAS representations serialized in JSON format. A such setup for a simulative DT is illustrated and elaborated in Figure 18.



(0) Asset connection using established communication protocols
(1) Simulation environment requests up-to-date values from reactive AAS
(2) Reactive AAS responds with values in JSON object
(3) Simulation environment updates simulated outputs to reactive AAS

Figure 18: Concept of a reactive AAS -based simulative DT.

Even though AAS has a strength in improving inter-firm data exchange, literature has not yet illustrated AAS applications in domain of supply chain management and SoS-level production coordination. A conceptual application for production simulation and interoperable data exchange in domain of SoS-level production coordination could involve exchanging estimated order delivery dates between companies via the AAS REST API, similarly to what interviewee RobotCo1 described when discussing future expectations for AAS. Two distinct use cases could be defined for this application area, one with simulation on server side and one with simulation on client side.

In the server-side conceptual use case, a vendor would serve simulated delivery dates via the AAS REST API upon requests. The delivery time simulation model would be connected to enterprise information systems to have real-time work order start dates available. The customers would only access the simulated delivery dates, but not access other information that would preferably be kept internal, as the simulation application would synthesize information on current production plan, process lead times, demand mix and resource availability into one date visible for the customer.

In the client-side conceptual use case, a simulation model depicting the real-time state of a factory could use estimated delivery dates from AAS API as inputs for material arrival times, i.e. simulation on client side. The simulation model would be agnostic on whether the vendor-provided delivery dates are based on human estimates, calculation logic of an enterprise information system, or a simulation model. Hypothetically, the supply chain coordination use cases would have the most probable economic viability in contexts with high level of product customizability, high number of suppliers, or high standard deviation of order lead times.

The AAS REST API specification also defines mechanisms for exchanging entire AASX package files. A conceptual client-side application utilizing this feature could enhance the potential early-phase engineering use cases discussed in the interviews. In one conceptual use case, an engineer could connect the simulation software with an AASX file server, download AASX files with CAD models to the simulation environment, design a unit or a system, create an AAS representation integrating the base AAS representations together based on the simulation components, and write arbitrary properties created in the simulation environment to the AAS representation. Benefits of a such application would include reusability of information in simulation model artefacts built during early-phase engineering, as interviewee ElectricalCo1 described in context of PLC engineering.

4.3.2 Summary of the use cases

Finally, all use cases that were derived from the research material are discussed. In order to discuss the resulting use cases conveniently, each use case will be referred with an abbreviated code: each use case is either an interview-based use case (IUC), prototypical use case (PUC), or a conceptual use case (CUC). The summary of the use cases is presented in Table 9 with the use case abbreviation as the first column.

This study resulted in definition of 13 potential use cases. However, the prototypical use cases of generating simulation components and connecting matching variables overlap with two use cases described by interviewees, reducing the number of unique use cases to 11. If the prototypical use cases are excluded, no use cases overlap, and seven unique use cases remain.

Table 9: Summary of the potential use cases.

Use case abbreviation	Category	Use case description	Feasibility of use case
IUC1	Use cases discussed in interviews	Generating product and process simulation models from AAS representations in PLM system	Official and proprietary AAS submodels stored in specific PLM systems must be further studied.
IUC2	Use cases discussed in interviews	Generating AAS representation from simulation model for collaborative engineering	Low complexity, technologically feasible.
IUC3	Use cases discussed in interviews	Writing production system parameters from simulation to AAS for commissioning	Low complexity, technologically feasible. Overlaps with PUC5.
IUC4	Use cases discussed in interviews	Partially generating simulation components from AAS representations created in external engineering systems	No specified representation of asset kinematic behaviour in official AAS submodels. Overlaps with PUC1.
PUC1	Prototypical use cases	Generating simulation components from AAS representations	No specified representation of asset kinematic behaviour in official AAS submodels. Resource classification -based heuristic used. Overlaps with IUC4.
PUC2	Prototypical use cases	Generating simulation layout from AAS	Ambiguity in AAS coordinate representation, no standardized way to represent interactions between units in AAS alone.
PUC3	Prototypical use cases	Generating assembly product model from AAS	Ambiguity in AAS coordinate representation.
PUC4	Prototypical use cases	Generating assembly process simulation model from AAS	Feasibility with the official BoP submodel must be tested after its publishing.
PUC5	Prototypical use cases	Connecting matching variables between AAS file and simulation	Technologically feasible. Overlaps with IUC3.
PUC6	Prototypical use cases	Exchanging values of simulation variables over REST API	Not successful, requires capability to make HTTPS requests.
CUC1	Conceptual use cases based on learnings	Serving simulated lead times over REST API	Requires capability to make HTTPS requests.
CUC2	Conceptual use cases based on learnings	Using product delivery dates from REST API as simulation inputs	Requires capability to make HTTPS requests
CUC3	Conceptual use case based on learnings	Integrating component representations to enriched system AAS	Requires capability to make HTTPS requests to an AASX file server, or access to an AASX file catalogue.

Use cases involving automatic model generation of simulation components are hindered by the lack of specifications around kinematic and interactional behaviour in AAS. However, the feasibility of using AML detail models embedded in AAS representations should be studied further in these use cases.

As the behaviour of assets is not described in AAS representations in sufficient level of detail, the use cases with system-level model generation are also hindered. However, generating a system layout without interactions of stations or a system simulation model with a product-centric view of processes are both feasible options, as was shown in PUC4.

4.3.3 Results for the research questions

The research question RQ1 aimed in recognizing problematic contexts where interplay of AAS and production simulation would create value by finding and defining potential use cases.

The sub-question RQ1.a: “What existing simulation use cases could be improved by increasing simulation environment interoperability?” aimed in defining these potential use cases to existing application contexts of production simulation in same companies that seek benefits from AAS. The results for this sub-question comprise the interviewee quotes on topic of PLM systems and use case IUC1; validation of product design was described to be conducted by simulating a manufacturing process for the product, and steps in configuring the product and process entities in the simulation model were described to have automation potential. Use case IUC4 could also be considered as a result for this sub-question, but it was found to have poor technological potential in current state of the AAS specifications.

The sub-question RQ1.b: “What existing AAS use cases could be improved by incorporating factory simulation in the application?” aimed to define the potential use cases to existing application contexts of AAS by adding production simulation technology in the application. Results for this sub-question include the interviewee quotes on creating information from simulation to AAS during early-phase engineering of a production system, the potential use cases IUC2 and IUC3, and the conceptual use case CUC3 as an enhancing concept for IUC2 and IUC3.

The sub-question RQ1.c: “What potential new use cases exist for AAS-interoperable factory simulation?” could not be answered fully due to the formulation of the question. The question was formulated to comprise the set of potential use cases not covered by RQ1.a and RQ1.b in a collectively exhaustive manner. The interview discussion theme of future expectations for AAS was expected to act as input for this research question, yielding use case CUC2 as a result, and CUC1 as a further concept in the same application area.

Finally, the result for the top-level research question RQ1: “What problems in manufacturing does AAS address, and which of these problems could discrete-event factory simulation help in solving?” could be summarized as follows: AAS addresses problems in data exchange between companies, and in centralization of information from different sources to digitalized representations of technological industrial products. AAS helps to realize the concept of DT by offering a syntactically and semantically interoperable representation format for information, which enables machine-readability of information that has been fused together from different sources across a value network. The potential role of production simulation in these applications is to act as one of the data sources and to enrich the information in AAS representations, either by creating representations of systems in the beginning, by computing feedback of design during development phase of an asset, or by computing values of non-time-critical parameters during the operational phase of the asset.

The research question RQ2, albeit answered partly in the literature section, addressed the research problem from a more practical and technical perspective. The results for sub-question RQ2.a: “What data stored in AAS virtual representations could be used as inputs for factory simulation models?” were demonstrated in the prototypical use cases, leveraging standard representation models for asset type information, installation coordinates, hierarchical structure, 3D CAD models, and technical data containing the technical properties specific for each product.

Result for the top-level research question RQ2: “How are AAS virtual representations of manufacturing resources structured?” could be summarized as follows: An AAS representation consists of a top-level description of the AAS and its respective asset, and purpose-specific submodels that could be added or removed during the life cycle of asset in a modular manner. In addition to that, concept description elements could be defined to create semantic definitions inside the AAS model, and supplemental files such as CAD models, pictures, documents or scripts can be attached in the package. Description of the asset’s functionalities would be modelled using capability definitions or attached simulation models, and communication interfaces of the physical asset are modelled using a standard submodel.

5 Discussion

This section will reflect on the results of this research. First, practical implications regarding the significance of AAS for the production simulation software are discussed in section 5.1. Next, the connection between this study and existing research on AAS is discussed in section 5.2. Finally, conclusions, limitations of this research, and directions for future research are presented in section 5.3.

5.1 Practical implications for production simulation applications

In summary, AAS was found not to serve the purpose of simulation model generation that the original formulation of the research topic assumed. This study was perhaps conducted too early, as many standard submodels relevant for production simulation environments were not yet published, and there had not yet been customer inquiries on AAS-interoperable production simulation applications.

The results of the interview study were rather disappointing given the rich variety of AAS applications illustrated in literature. The industrial applications of AAS described by the interviewees were mostly focused on domains of product data management and PLM, in contrast to system-level DT implementations illustrated in AAS literature. Despite the benefits introduced by AAS for industrial companies, the application areas of AAS and production simulation overlap only narrowly. The use cases ideated in the interactive interviews were limited to three application areas: data exchange with PLM systems, using factory simulation software as an early-phase engineering tool to create AAS representations for collaborative purposes, or partially generating simulation components from AAS:s.

For generating unit- or system-level simulation models, official AAS submodels alone carry too little relevant information to employ model generation approaches that rely on other than resource classification heuristics or existing unit simulation models. However, it is recommended to monitor the future model specifications created in the DIAMOND project, as these specifications could potentially enable model interoperability with additional classes of manufacturing resources. Considering also that AML detail models can be included in AAS representations, application logic of further automatic model generation applications should consider the modelling conventions in AML when generating unit- or system-level simulation models.

A surprising result from the interviews was that more potential use cases were named by interviewees for writing information from factory simulation models to AAS representations, and not vice versa. One possible explanation is that very few of the AAS submodels released so far would contain data that would act as inputs for system-level simulation models. Another possible explanation is that engineering tool providers may not yet have had time to adopt the AAS format and build workflows for streamlined creation of AAS virtual representations, which is why many interviewees reported challenges with the digital tools used for creating AAS:s. A third possible explanation is the temporal context of production simulation, as applications for the simulation software are focused on the early phase of a factory life cycle, implying that the production simulation software is in the providing end of information rather than in the consuming end.

In the future, production simulation environments should offer frameworks for creating custom communication interfaces, as they could allow wider integration with external information systems and fusing data from different sources to simulation models with greater predictive capabilities. In this case, the ability to make HTTPS requests from inside production simulation environment would not only enable leveraging AAS REST API in both client- and server-side simulation applications, but also other interfacing capabilities based on REST API, namely connection to ERP systems.

A challenge which a simulation software vendor might face in modelling of manufacturing resources in the future is that non-confidential information on functionalities of manufacturing resources might be veiled in too much abstraction that accurate simulation models could be created from that information alone. Interviewees reported preferences of attaching existing simulation models, either proprietary or FMI, to AAS representations of their products to serve the exact purpose of simulating the asset while simultaneously avoiding disclosing the inner mechanisms of the products for the public. Another related phenomenon in AAS is the future concept of capabilities, which would be an abstraction of mechanisms and logic with an apparent purpose of serving in smart factory or control applications without disclosure of confidential engineering information. These phenomena could signal a shift towards utilizing data-driven black-box models in design and engineering of industrial systems, if outcomes of engineering with sufficient quality could be achieved using these abstracted models for subsystems.

5.2 Contributions to literature

This study contributed to AAS research in six ways. First, AAS modelling approaches of system-level manufacturing assets was reviewed in the literature section. Second, an up-to-date comprehensive review of implemented AAS application areas in literature was conducted. Third, AAS was studied in context of production simulation, which was a research gap. Fourth, the interview data collected from five industrial companies sheds light on benefits, challenges and opportunities introduced by AAS in industrial contexts. Fifth, a novel concept of a simulation application interacting via a REST API was proposed in section 4.3.1. Finally, the concept of interaction between AAS representations and a production simulation software was formalized in section 4.3.3.

Next, criticism towards the existing research on AAS topics will be presented. First, there is a disconnect between the recent trend in AAS-related research and the industrial use of the technology, namely in applications with full and holistic representations of manufacturing systems. The division of roles between AAS, AML and OPC UA should be considered also in research and not only industrial implementations, so that AAS development efforts could focus on innovations that do not overlap the existing and established capabilities of AML and OPC UA. Second, publications on AAS tend to describe the implemented virtual representations in an ambiguous manner. In future, case study publications on AAS topics should illustrate the used virtual representation information models even on the level of listing the used submodels, and in case of custom submodels provide a detailed illustration to improve research replicability. Currently, numerous publications on the research theme illustrate an AAS application, but do not present the data modelled in the AAS.

5.3 Conclusions

AAS is a technology for syntactic and semantic interoperability over the life cycle of manufacturing resources, designed by and for industrial companies participating in the development. Based on the interview study, the overlapping application areas of production simulation and AAS are during the design phase of products or manufacturing systems. The interaction between AAS and production simulation could be defined as follows: *Asset type representation is enriched with feedback created by behavioural simulation of asset instances.*

AAS has a strong positioning and investment from industrial companies. A competitive strength of AAS is its design philosophy: The landscape of existing industry standards is considered in design of specifications with aim of interoperability with other standards rather than aim of replacing them. However, for consuming applications this also means that AAS alone is insufficient for functional modelling of assets, and the consuming applications must understand the logic of complementary standards such as AML.

5.3.1 Limitations

The validity of the findings in this study is limited by issues in sampling the interviews and assumptions made in the technological experimentation. These limitations are analysed next.

The interview sampling introduces potential sources of bias limiting the validity of the interview data. First, all interviewees from manufacturing sector represented large enterprises. Second, all manufacturing companies represented in the interview study are located in Central Europe. Third, all manufacturing companies in the sample are either components manufacturers or system integrators, meaning that companies operating the assets and using AAS virtual representations during the operational phase of life cycle were not present in the sample. Interviewing representatives of companies from automotive industry would have given an additional perspective, as the automotive industry was found out to be a driving force for certain vendors adopting AAS. Some interviewees described the AAS-related aspirations of customers in automotive industry, but first-hand evidence from the automotive industry could perhaps have provided input for more potential use cases.

Two interviews were scheduled too short, as the amount of input could not be predicted beforehand. This happened in the interviews with ElectricalCo2 and MachineryCo2, as these interviewees described surprisingly many industrial applications for AAS, and the interview could not be continued over the scheduled time due to calendar limitations. In both cases, this was noticed during the interview and discussion on generic themes was shortened by skipping less important questions.

The technological experimentation resulted in a proof-of-concept application which can read and write data in AAS format. The working principle of the application was described in section 3.2, and the results and learnings in section 4.2. The prototypical use cases within the application were simplified by assumptions that limit the validity of adopting the implementations as industry-ready and generalizable solutions. First, the implementation does not support reading contents of an AAS repository, but only processing single AASX files that contain one or many AAS:s. Second, all mapping operations in the prototypical use cases utilized name-based heuristic rules in English instead of semantic dictionary references. Third, the method chosen for the prototypical use case of generating simulation components is applicable only for implemented classes of manufacturing resources. Fourth, the prototypical use case of generating a process model was based on a custom AAS submodel for representing product BoP, as the official submodel for the purpose was not published by the time of conducting this study. The technological feasibility and generalizability of this use case is therefore contingent on the future submodel and its scope of representing processes.

5.3.2 Directions for future research

In context of production simulation software, future research could study the interrelation of PLM and production simulation, perhaps by reviewing application areas of interfacing production simulation with PLM systems. Another potential future research topic could be a proof-of-concept implementation of executing FMI simulation models within the production simulation environment. A such implementation could foster interoperability with additional standardized representation formats by allowing reusability of component FMI simulation models provided by vendors.

Future research in theme of AAS could conceptualize and illustrate SoS-level applications for the technology. In domain of supply chain planning, a future study could implement the AAS REST API setting for exchanging order delivery time information discussed in section 4.3.1.

In context of smart manufacturing and field of automation engineering, future research could conceptualize a control setting for matrix manufacturing systems using product AAS representations as input for autonomous control decisions. However, practical implementations in this research topic could wait that conventions for capability descriptions or proactive AAS functionalities are established in AAS specifications so that more standard building blocks would exist for the purpose.

Future research could also demonstrate implementations of utilizing Large Language Model technologies for achieving semantic interoperability, for example by classifying assets and automatically building semantically identified asset virtual representations.

References

Abdel-Aty, A., Negri, E. & Galparoli, S. (2022) Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin. IFAC Papers Online. [Online] 55 (10), 2533–2538

Anumbe, N., Saidy, C. & Harik, R. (2022) A Primer on the Factories of the Future. Sensors (Basel, Switzerland). [Online] 22 (15), 5834-.

Arm, J., Benesl, T., Marcon, P., Bradac, Z., Schröder, T., Belyaev, A., Werner, T., Braun, V., Kamensky, P., Zezulka, F., Diedrich, C. & Dohnal, P. (2021) Automated Design and Integration of Asset Administration Shells in Components of Industry 4.0. Sensors (Basel, Switzerland). [Online] 21 (6), 2004-.

AutomationML consortium. (2024) AutomationML Application Recommendation: AutomationML in AAS. Magdeburg, Germany: AutomationML e.V. 48 p.

Bangsow, Steffen. (2012) Use cases of discrete event simulation: appliance and research. [Online]. Berlin; Springer.

Bauer, D. A. & Mäkiö, J. (2019) 'Hybrid Cloud - Architecture for Administration Shells with RAMI4.0 Using Actor4j', in 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). [Online]. 2019 IEEE. pp. 79–86.

Bavelos, A., Dimosthenopoulos, D., Togias, T., Kousi, N. & Makris, S. (2022) Integrating AR based operator support features for reconfigurable production systems. Procedia CIRP. [Online] 106276–281.

Ben-Daya, M., Hassini, E. & Bahroun, Z. (2019) Internet of things and supply chain management: a literature review. International Journal of Production Research. [Online] 57 (15–16), 4719–4742.

Bensmaine, A., Dahane, M. & Benyoucef, L. (2014) A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. International journal of production research. [Online] 52 (12), 3583–3594.

Bhandal, R., Meriton, R., Kavanagh, R. & Brown, A. (2022) The application of digital twin technology in operations and supply chain management: a bibliometric review. Supply chain management. [Online] 27 (2), 182–206.

Bruckner, D., Stanica, M., Blair, R., Schriegel, S., Kehrer, S., Seewald, M. & Sauter, T. (2019) An Introduction to OPC UA TSN for Industrial Communication Systems. Proceedings of the IEEE. [Online] 107 (6), 1121–1131.

Cavalieri, S. & Gambadoro, S. (2024) Digital Twin of a Water Supply System Using the Asset Administration Shell. Sensors. [Online] 24 (5), 1360-.

Chilwant, N. & Kulkarni, M. S. (2019) Open Asset Administration Shell for Industrial Systems. Manufacturing letters. [Online] 2015–21.

da Rocha, H., Pereira, J., Abrishambaf, R. & Espirito Santo (2022) An Interoperable Digital Twin with the IEEE 1451 Standards. Sensors (Basel, Switzerland). [Online] 22 (19), 7590-.

Deuter, A. & Imort, S. (2021) Product Lifecycle Management with the Asset Administration Shell. Computers (Basel). [Online] 10 (7), 84-.

di Orio, G., Malo, P. & Barata, J. (2019) 'NOVAAS: A Reference Implementation of Industrie4.0 Asset Administration Shell with best-of-breed practices from IT engineering', in IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society. [Online]. 2019 IEEE. pp. 5505–5512.

Dickopf, T., Forte, S., Weber, S., Sturmer, C. & Muggeo, C. (2023) 'Application of the Asset Administration Shell in the context of Engineering Data Management Systems', in 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2023 IEEE. pp. 1–8.

Diedrich, C., Belyaev, A., Schroder, T., Vialkowitsch, J., Willmann, A., Usländer, T., Koziolek, H., Wende, J., Pethig, F. & Niggemann, O. (2017) 'Semantic interoperability for asset communication within smart factories', in 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2017 IEEE. pp. 1–8.

Drath, R. & Horch, A. (2014) Industrie 4.0: Hit or Hype? [Industry Forum]. IEEE Industrial electronics magazine. Vol. 8. [Online]. New York: IEEE.

Drath, R. (2021) AutomationML: A Practical Guide. [Online]. Berlin; De Gruyter Oldenbourg.

Fimmers, C., Blanke, P., Wieczorek, M., Petrovic, O. & Herfs, W. (2023) Asset Administration Shells in Tool Lifecycle Monitoring. Procedia CIRP. [Online] 1201209–1214.

Fraile, F., Sanchis, R., Poler, R. & Ortiz, A. (2019) Reference Models for Digital Manufacturing Platforms. Applied sciences. [Online] 9 (20), 4433-.

Gil, S., Mikkelsen, P., Gomes, C. & Larsen, P. (2024) Survey on open-source digital twin frameworks–A case study approach. Software, practice & experience. [Online] 54 (6), 929–960.

Gollner, D., Pawlik, T. & Schulte, T. (2021) 'Utilization of the Asset Administration Shell for the Generation of Dynamic Simulation Models', in 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). [Online]. 2021 Piscataway: IEEE. pp. 808–812.

Grangel-González, I., Halilaj, L., Auer, S., Lohmann, S., Lange, C. & Collarana, D. (2016) "An RDF-based approach for implementing industry 4.0 components with Administration Shells," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). [Online] pp. 1–8. doi: 10.1109/ETFA.2016.7733503.

Heppner, S., Miny, T., Kleinert, T., Becker, M., Schmitz, K. & Alt, R. (2023) "Asset Administration Shells as Data Layer for Enabling Automated Simulation-based Engineering," in 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2023 IEEE. pp. 1–7.

Himmelstoß, H., Hall, R., Thieme, P. & Bauernhansl, T. (2023) Designing Smart Services for a Component Manufacturer Based on a Digital Twin. Procedia CIRP. [Online] 119950–956.

Hollocks, B. W. (2006) Forty years of discrete-event simulation-a personal reflection. The Journal of the Operational Research Society. [Online] 57 (12), 1383–1399.

Hu, F., Wang, W. & Zhou, J. (2023) Petri nets -based digital twin drives dual-arm cooperative manipulation. Computers in industry. [Online] 147103880-.

IDTA-01001-3-0 (2023) Specification of the Asset Administration Shell. Part 1: Metamodel. Frankfurt, Germany: Industrial Digital Twin Association. 213 p.

IDTA-01002-3-0 (2023) Specification of the Asset Administration Shell. Part 2: Application Programming Interfaces. Frankfurt, Germany: Industrial Digital Twin Association. 185 p. [Online]

IDTA-01005-3-0 (2023) Specification of the Asset Administration Shell. Part 5: Package File Format (AASX). Frankfurt, Germany: Industrial Digital Twin Association. 27 p.

IDTA-02005-1-0 (2022) Provision of Simulation Models. Frankfurt, Germany: Industrial Digital Twin Association. 46 p.

IDTA-02011-1-0 (2023) Hierarchical Structures Enabling Bills Of Material. Frankfurt, Germany: Industrial Digital Twin Association. 20 p.

IDTA-02013-1-0 (2022) Reliability. Frankfurt, Germany: Industrial Digital Twin Association. 14 p.

IDTA-02017-1-0 (2024) Asset Interfaces Description. Frankfurt, Germany: Industrial Digital Twin Association. 47 p.

IDTA-02026-1-0 (2024) Provision of 3D models. Frankfurt, Germany: Industrial Digital Twin Association. 130 p.

IDTA-02027-1-0 (2024) Asset Interfaces Mapping Configuration. Frankfurt, Germany: Industrial Digital Twin Association. 18 p.

IDTA (2022) The Digital Twin in the Automotive Industry: The Framework for Standardized and Independent Data Exchange. Available at: https://industrialdigitaltwin.org/en/news-dates/the-digital-twin-in-the-automotive-industry-the-framework-for-standardized-and-independent-data-exchange-4873 (Accessed: 29 Aug 2024).

IDTA (2024) AAS Submodel Templates. Available at: https://industrialdigitaltwin.org/en/content-hub/submodels (Accessed: 28 Aug 2024).

IDTA (2024) Service portal by Wittenstein – Touch point for Industry 4.0. Available at: https://industrialdigitaltwin.org/en/use-cases/service-portal-by-wittenstein-touch-point-for-industry-4-0 (Accessed: 26 Jun 2024).

IEC63278-1 (2023) Asset Administration Shell for industrial applications. Part 1: Asset Administration Shell structure. Genève: International Electrotechnical Commission. 149 p.

Ismail, A., Truong, H. & Kastner, W. (2019) Manufacturing process data analysis pipelines: a requirements analysis and survey. Journal of big data. [Online] 6 (1), 1–26.

Isometsä, J. (2018) Component classification and best practices in production simulation modeling.

Jacoby, M. & Usländer, T. (2020) Digital Twin and Internet of Things—Current Standards Landscape. Applied sciences. [Online] 10 (18), 6519-.

Jacoby, M., Jovicic, B., Stojanovic, L. & Stojanovic, N. (2021) An Approach for Realizing Hybrid Digital Twins Using Asset Administration Shells and Apache StreamPipes. Information (Basel). [Online] 12 (6), 217-.

Jacoby, M., Volz, F., Weißenbacher, C. & Müller, J. (2022) 'FA3ST Service - An Open Source Implementation of the Reactive Asset Administration Shell', in 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2022 Piscataway: IEEE. pp. 1–8.

Jeong, D.-Y., Baek, M.-S., Lim, T.-B., Kim, Y.-W., Kim, S.-H., Lee, Y.-T., Jung, W.-S. & Lee, I.-B. (2022) Digital Twin: Technology Evolution Stages and Implementation Layers with Technology Elements. IEEE access. [Online] 1052609–52620.

Khazraei, K. & Deuse, J. (2011) A strategic standpoint on maintenance taxonomy. Journal of facilities management. [Online] 9 (2), 96–113.

Kim, K.-D. & Kumar, P. R. (2012) Cyber-Physical Systems: A Perspective at the Centennial. Proceedings of the IEEE. [Online] 100 (Special Centennial Issue), 1287–1308.

Kim, B., Kim, S., Teijgeler, H., Lee, J., Lee, J.Y., Lim, D., Suh, H.-W. & Mun, Duhwan. (2022) Use of Asset Administration Shell Coupled with ISO 15926 to Facilitate the Exchange of Equipment Condition and Health Status Data of a Process Plant. Processes. [Online] 10 (10), 2155-.

Kritzinger, W., Karner, M., Traar, G., Henjes, J. & Sihn, W. (2018) Digital Twin in manufacturing: A categorical literature review and classification. [Online]

Laemmle, A. & Gust, S. (2019) Automatic layout generation of robotic production cells in a 3D manufacturing simulation environment. Procedia CIRP. [Online] 84316–321.

Lattanzi, L., Raffaeli, R., Peruzzini, M. & Pellicciari, M. (2021) Digital twin for smart manufacturing: a review of concepts towards a practical industrial implementation. International journal of computer integrated manufacturing. [Online] 34 (6), 567–597.

Lee, J., Bagheri, B. & Kao, H.-A. (2015) A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manufacturing letters. [Online] 318–23.

Li, W., Zhou, H., Lu, Z. & Kamarthi, S. (2024) Navigating the Evolution of Digital Twins Research through Keyword Co-Occurence Network Analysis. Sensors (Basel, Switzerland). [Online] 24 (4), 1202-.

Lins, T. & Oliveira, R.A.R. (2020) Cyber-physical production systems retrofitting in context of industry 4.0. Computers & industrial engineering. [Online] 139106193-.

Lu, Y., Xu, X. & Xu, J. (2014) Development of a Hybrid Manufacturing Cloud. Journal of manufacturing systems. [Online] 33 (4), 551–566.

Lu, Y. & Xu, X. (2018) Resource virtualization: A core technology for developing cyber-physical production systems. Journal of manufacturing systems. [Online] 47128–140.

Luxenburger, A., Porta, D., Knoch, S., Mohr, J. & Schwartz, T. (2023) ‘A Service Infrastructure for Industrie 4.0 Testbeds based on Asset Administration Shells’, in 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2023 IEEE. pp. 1–8.

Mabkhot, M., Al-Ahmari, A., Salah, B. & Alkhalefah, H. (2018) Requirements of the Smart Factory System: A Survey and Perspective. Machines (Basel). [Online] 6 (2), 23-.

Martinez, E., Ponce, P., Macias, I. & Molina, A. (2021) Automation Pyramid as Constructor for a Complete Digital Twin, Case Study: A Didactic Manufacturing System. Sensors (Basel, Switzerland). [Online] 21 (14), 4656-.

Miny, T. Thies, M., Lukic, L., Kabisch, S., Oladipupo, K., Diedrich, C. & Kleinert, T. (2023) “Overview and Comparison of Asset Information Model Standards”. IEEE access. [Online] 1199189–99221.

Moghaddam, M., Cadavid, M., Kenley, C. & Deshmukh, A. (2018) Reference architectures for smart manufacturing: A critical review. Journal of manufacturing systems. [Online] 49215–225.

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. & Ueda, K. (2016) Cyber-physical systems in manufacturing. CIRP annals. [Online] 65 (2), 621–641.

Moreno, T., Almeida, A., Toscano, C., Ferreira, F. & Azevedo, A. (2023) Scalable Digital Twins for industry 4.0 digital services: a dataspaces approach. Production & manufacturing research. [Online] 11 (1).

Nguyen, Q.-D., Huang, Y., Keith, F., Leroy, C., Thi, M.-T. & Dhouib, S. (2024) Manufacturing 4.0: Checking the Feasibility of a Work Cell Using Asset Administration Shell and Physics-Based Three-Dimensional Digital Twins. Machines (Basel). [Online] 12 (2), 95-.

Onaji, I., Tiwari, D., Soulantiantork, P., Song, B. & Tiwari, A. (2022) Digital twin in manufacturing: conceptual framework and case studies. International journal of computer integrated manufacturing. [Online] 35 (8), 831–858.

Onggo, B., Corlu, C., Juan, A., Monks, T. & de la Torre, R. (2021) Combining symbiotic simulation systems with enterprise data storage systems for real-time decision-making. Enterprise information systems. [Online] 15 (2), 230–247.

Park, K.T., Son, Y.H. & Noh, S.D. (2021) The architectural framework of a cyber physical logistics system for digital-twin-based supply chain control. International journal of production research. [Online] 59 (19), 5721–5742.

Park, K., Yang, J. & Noh, S. (2021) VREDI: virtual representation for a digital twin application in a work-center-level asset administration shell. Journal of intelligent manufacturing. [Online] 32 (2), 501–544.

Platenius-Mohr, M., Malakuti, S., Grüner, S., Schmitt, J. & Goldschmidt, T. (2020) “File- and API-based interoperability of Digital Twins by Model transformation: An IIoT case study using asset administration shell”. Future generation computer systems. [Online] 11394–105.

Plociennik, C., Pourjafarian, M., Nazeri, A., Windholz, W., Knetsch, S., Rickert, J., Ciroth, A., Precci Lopes, A., Hageborn, T., Vogelgesang, M., Benner, W., Gassmann, A., Bergweiler, S., Ruskowski, M., Schebek, L. & Weidenkaff, A. (2022) Towards a Digital Lifecycle Passport for the Circular Economy. Procedia CIRP. [Online] 105122–127.

Quadrini, W., Cimino, C., Abdel-Aty, T., Fumagalli, L. & Rovere, R. (2023) Asset Administration Shell as an interoperable enabler of Industry 4.0 software architectures: a case study. Procedia computer science. [Online] 2171794–1802.

Rahmanzadeh, S., Pishvaee, M.S. & Govindan, K. (2023) Emergence of open supply chain management: the role of open innovation in the future smart industry using digital twin network. Annals of operations research. [Online] 329 (1–2), 979–1007.

Rajkumar, R., Lee, I., Sha, L. & Stankovic, J. (2010) 'Cyber-physical systems: the next computing revolution', in Design Automation Conference. [Online]. 2010 ACM. pp. 731–736.

Rauh, L., Gärtner, S., Brandt, D., Oberle, M., Stock, D. & Bauernhansl, T. (2022) Towards AI Lifecycle Management in Manufacturing Using the Asset Administration Shell (AAS). Procedia CIRP. [Online] 107576–581.

Redelinghuys, A., Basson, A. & Kruger, K. (2020) A six-layer architecture for the digital twin: a manufacturing case study implementation. Journal of intelligent manufacturing. [Online] 31 (6), 1383–1402.

Renna, P. & Ambrico, M. (2015) Design and reconfiguration models for dynamic cellular manufacturing to handle market changes. International journal of computer integrated manufacturing. [Online] 28 (2), 170–186.

Renna, P. (2024) Performance analysis of matrix-structured manufacturing systems compared to dedicated flow and reconfigurable manufacturing lines. Production engineering (Berlin, Germany). [Online] 18 (3–4), 625–642.

Ribeiro, L. & Bjorkman, M. (2018) Transitioning from Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges. IEEE systems journal. [Online] 12 (4), 3816–3827.

Schweichhart, K. (2016). Reference Architectural Model Industrie 4.0 (RAMI4.0). Available at: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf (Accessed 18 Jul 2024).

Semeraro, C., Lezoche, M., Panetto, H. & Dassisti, M. (2021) Digital twin paradigm: A systematic literature review. Computers in industry. [Online] 130103469-.

Siatras, V., Bakopoulos, Mavrothalassitis, P., Nikolakis, N. & Alexopoulos, K. (2023) On the Use of Asset Administration Shell for Modeling and Deploying Production Scheduling Agents within a Multi-Agent System. Applied sciences. [Online] 13 (17), 9540-.

Singh, M., Srivastava, R., Fuenmayor, E., Kuts, V., Qiao, Y., Murray, N. & Devine, D. (2022) Applications of Digital Twin across Industries: A Review. Applied sciences. [Online] 12 (11), 5727-.

Sorvari, J. (2021) Automatic generation of a 3D factory simulation from a Systems Modeling Language design.

Stock, D., Schel, D. & Bauernhansl, T. (2019) 'Cyber-Physical Production System Self-Description-Based Data Access Layer', in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2019 IEEE. pp. 168–175.

Stock, D., Schneider, M. & Bauernhansl, T. (2021) Towards Asset Administration Shell-Based Resource Virtualization in 5G Architecture-Enabled Cyber-Physical Production Systems. Procedia CIRP. [Online] 104945–950.

Striffler, N. & Voigt, T. (2023) Concepts and trends of virtual commissioning – A comprehensive review. Journal of manufacturing systems. [Online] 71664–680.

Tantik, E. & Anderl, R. (2017) Potentials of the Asset Administration Shell of Industrie 4.0 for Service-Oriented Business Models. Procedia CIRP. [Online] 64363–368.

Tao, F., Zhang, M., Liu, Y. & Nee, A. (2018) Digital twin driven prognostics and health management for complex equipment. CIRP annals. [Online] 67 (1), 169–172.

Tao, F., Zhang, H., Liu, A. & Nee, A. (2019) Digital Twin in Industry: State-of-the-Art. IEEE transactions on industrial informatics. [Online] 15 (4), 2405–2415.

Thelen, A., Zhang, X., Fink, O., Lu, Y., Ghosh, S., Yon, B., Todd, M., Mahadevan, S., Hu, C. & Hu, Z. (2022) A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies. Structural and multidisciplinary optimization. [Online] 65 (12), .

Thiede, S. (2021) Cyber-Physical Production Systems (CPPS): Introduction. Journal of Manufacturing and Materials Processing. [Online] 5 (1), 24-.

Trierweiler, M., Foith-Förster, P. & Bauernhansl, T. (2020) Changeability of Matrix Assembly Systems. Procedia CIRP. [Online] 931127–1132.

van der Valk, H., Hunker, J., Rabe, M. & Otto, B. (2020) “Digital Twins in Simulative Applications: A Taxonomy,” in 2020 Winter Simulation Conference (WSC). [Online]. 2020 IEEE. pp. 2695–2706.

van Erp, T., Goncalves, R. & Rytter, N. (2023) Design of matrix production systems: A skill-based systems engineering approach. Procedia CIRP. [Online] 1201173–1178.

van Erp, T., da Silva, E. & Stenger, M. (2023) Industrial Digital Twin as a facilitator for business model innovations in the manufacturing industry. Procedia CIRP. [Online] 118330–335.

Volz, F., Sutschet, G., Stojanovic, L. & Usländer, T. (2023) On the Role of Digital Twins in Data Spaces. Sensors (Basel, Switzerland). [Online] 23 (17), 7601-.

Wagire, A., Rathore, A. & Jain, R. (2020) “Analysis and synthesis of Industry 4.0 research landscape: using latent semantic analysis approach”. Journal of manufacturing technology management. [Online] 31 (1), 31–51.

Wagner, C., Grothoff, J., Epple, U., Drath, R., Malakuti, S., Grüner, S., Hoffmeister, M. & Zimermann, P. (2017) “The role of the Industry 4.0 asset administration shell and the digital twin during the life cycle of a plant,” in 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). [Online]. 2017 IEEE. pp. 1–8.

Wang, J., Niu, X., Gao, R., Huang, Z. & Xue, Ruijuan. (2023) Digital twin-driven virtual commissioning of machine tool. Robotics and computer-integrated manufacturing. [Online] 81102499-.

Xu, L.D., He, W.& Li, S. (2014) Internet of Things in Industries: A Survey. IEEE transactions on industrial informatics. [Online] 10 (4), 2233–2243.

Ye, X., Jiang, J., Lee, C., Kim, N., Yu, M. & Hong, S. (2020) Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach. Vol. 14. [Online]. New York: IEEE.

Ye, X., Yu, M., Song, W.S. & Hong, S.H. (2021) An Asset Administration Shell Method for Data Exchange Between Manufacturing Software Applications. IEEE access. [Online] 91–1.

Ye, X., Hong, S., Song, W., Kim, Y. & Zhang, X. (2021) An Industry 4.0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems. IEEE access. [Online] 9154448–154459.

Ye, X., Song, W., Hong, S., Kim, Y. & Nam, H. (2022) Toward Data Interoperability of Enterprise and Control Applications via the Industry 4.0 Asset Administration Shell. IEEE access. [Online] 1035795–35803.

Yu, F. & Zheng, C. (2021) Tools, application areas and challenges of factory simulation in Small and Medium-Sized Enterprises – A Review. Procedia CIRP. [Online] 104399–404

Zeid, A., Sundaram, S., Moghaddam, M., Kamarthi, S. & Marion, T. (2019) Interoperability in Smart Manufacturing: Research Challenges. Machines (Basel). [Online] 7 (2), 21-.

Zheng, P. & Sivabalan, A. S. (2020) A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment. Robotics and computer-integrated manufacturing. [Online] 64101958-.

Zhou, L., Jiang, Z., Geng, N., Niu, Y., Cui, F., Liu, K. & Qi, N. (2022) Production and operations management for intelligent manufacturing: a systematic literature review. International journal of production research. [Online] 60 (2), 808–846.

Åhlström, P. & Westbrook, R. (1999) Implications of mass customization for operations management: An exploratory survey. International journal of operations & production management. [Online] 19 (3), 262–275.

Appendix A: Literature included in Figure 6

Modelled assets in publications that were reviewed in Abdel-Aty et al. (2022). Full references of the publications are available in Abdel-Aty et al. (2022).

	Product	Field device	Control device	Station	Work center	Enterprise	Connected world
Tantik and Anderl, 2017		X					
Lang et al., 2019				X			
Cavalieri and Salafia, 2020a				X			
Lüder et al., 2020				X			
Inigo et al., 2020		X		X			
Cavalieri and Salafia, 2020b		X	X				
Pribiš et al., 2021			X				
Schnicke et al., 2021		X					
Lv et al., 2021		X					
Ye et al., 2021b							
Ye et al., 2021a		X		X			
Platenius-Mohr et al., 2019		X					
Ye and Hong, 2019		X	X				
Cavalieri and Salafia, 2020c			X				
Marcon et al., 2019		X					
Assadi et al., 2020)		X					
Terzimehić et al., 2019		X					
Deuter and Imort, 2021	X						
Lopez et al., 2021	X	X					
Vogel-Heuser et al., 2021	X			X			
Arm et al., 2021	X			X			
Birtel et al., 2020	X						
Lu et al., 2021		X					
Pethig et al., 2017				X			
Barig et al., 2019							
Lang et al., 2018				X			
Motsch et al., 2021				X			
Hosseini et al., 2021		X	X	X			
Ye et al., 2020		X		X			
TOTAL	**5**	**15**	**5**	**12**	**0**	**0**	**0**

Modelled assets in literature that was reviewed in this study. Detailed descriptions of the applications are presented in section 2.3.5 and Table 5.

	Product	Field device	Control device	Station	Work center	Enterprise	Connected world
Bavelos et al. 2022	X	X		X			
Cavalieri & Gambadori 2024		X	X	X			
Dickoph et al. 2023	X						
Fimmers et al. 2023		X					
Himmelstosh et al. 2023		X					
Kim et al. 2022		X	X	X	X		
Moreno et al. 2023				X			
Nguyen et al. 2024		X		X			
Plociennik et al. 2023	X						
Park et al. 2021b	X			X	X		
Luxenburger et al. 2023	X	X	X	X	X		
Quadrini et al. 2023	X			X	X		
Siatras et al. 2023							
Volz et al. 2023	X						
Rauh et al. 2022							
Stock et al. 2021		X					
Jacoby et al. 2022	X			X			
TOTAL	**8**	**8**	**3**	**9**	**4**	**0**	**0**

Appendix B: Example interview questions per theme

A. Applications of AAS within the company

- Could you describe the use cases your company has for AAS?
- Do you have plans for future AAS use cases?
- Are you planning to use AAS in a DT application, and what other technologies would the implementation use?
    i. Which communication standards or protocols are you planning to use?
    ii. Which entities would the DT model?

B. Discussing expectations regarding interoperable factory simulation

- Does your company use factory simulation and AAS together in some task?
- If the simulation software could read and write component descriptions in an interoperable format,
    i. how would you use it in layout design phase?
    ii. how would you use it in production system feasibility study phase?
    iii. how would you use it in virtual commissioning phase?
    iv. how would you use it during the factory operations phase?
- (In the end of the interview) Did you imagine any use cases of AAS and Visual Components together during our discussions?

C. Discussing experienced interoperability issues

- What costs are introduced by the interoperability issue?
- What benefits does interoperability bring in your application?
- What data can be exchanged with your information systems in AAS format?
    i. Have you seen industrial applications of exchanging data with ERP, MES or PLM in AAS format?
    ii. What data was exchanged with the ERP/MES/PLM in AAS format?
- Do you know about use cases where production simulation is connected to enterprise information systems?
    i. What simulation use cases would have required an interface with PLM?
    ii. What data in the PLM system would have been required for a simulation model?
    iii. Could you think of a hypothetical use case where simulation is connected with a PLM system?
- Have you heard of use cases where production simulation uses REST API?
    i. Do you have some use case where a simulation model is connected to a REST API?
    ii. Would your simulation models benefit from having up-to-date data from another company's premises?

iii. Could you think of a hypothetical use case where simulation inputs are fetched from an API?
iv. Could you think of a hypothetical use case where simulation outputs are shared over the internet?

D. Discussing expectations and views on I4.0 technologies

- Do you think some Industry 4.0 technology has not lived up to expectations?
- What future expectations do you have for AAS?
- What unsolved problems could be solved by AAS in the future?

E. Discussing strengths and weaknesses of AAS

- What AAS enables that existing technologies haven't yet enabled?
- Some AAS applications model factory-floor functionalities of assets. Is AAS is the right technology for this?
- Where is information of your material flow stored? AAS model, AML model, MES system, some other?
- What challenges have you experienced in using AAS for your use case?
- Does AAS have some technological or architectural weakness that hinders your use cases?
- Would you like to criticize AAS of something?

Appendix C: Outline of interview analysis table

Quote theme \ Interviewee	AutomationCo	ElectricalCo1	ElectricalCo2	MachineryCo1	MachineryCo2	PumpCo	RobotCo1	RobotCo2	SimulationCo1	SimulationCo2
Existing use case of AAS										
Benefits of AAS										
Challenge in using AAS										
Technical weakness of AAS										
Criticism towards AAS										
Distinct role of AAS in information modelling										
Distinct role of AAS in communication										
Future expectation for AAS										
Use of AAS in automotive industry										
Co-use of REST API and simulation										
PLM										
Potential use cases in exchanging engineering data										
Comments on a prototypical use case										